Distributed Routing Software

Event Logging System Messages Guide

Part Number: AA–QL2AB–TE

December 1995

This manual provides information about the Event Logging System process.

Revision/Update Information:	This is a revised manual.
Software Version:	Distributed Routing Software V1.0A

This manual was produced by Shared Engineering Services.

Contents

Introduction

This chapter describes the outputs of the ELS process. A large part of the ELS functionality is based on commands that use the subsystem, event number, and logging level as parameters. These commands are documented in the *System Software Guide*.

Message Presentation

The format of the message explanations in this guide is as follows:

Level:	Describes the logging level of the error message.
Short Syntax:	Shows the message that is displayed on the router console. This is a compressed form of the message.
Long Syntax:	Shows the full, expanded text of the message.
Description:	Explains the meaning of the error message.
Cause:	Describes possible causes of the error that caused this message.
Action:	Specifies possible action to correct the error.

Causes of Events

Events monitored by the Event Logging System (ELS) occur continuously while the router is operating. Any of the following reasons can cause them.

- System activity
- Status changes
- Service requests
- Data transmission and reception
- Data and internal errors

When an event occurs, ELS receives data from the system that identifies the source and nature of the event. Then, ELS generates a message that uses the data received as part of the message.

Interpreting a Message

This section describes how to interpret a message generated by ELS. Table 1–1 shows the principal elements of a message and describes the elements.

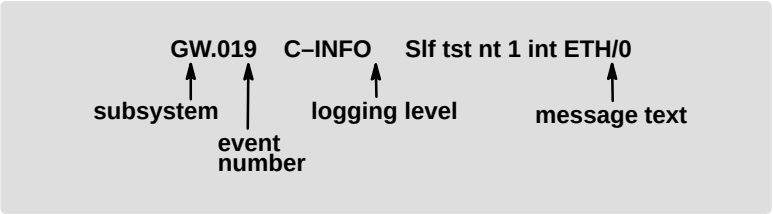

Table 1–1 Message Descriptions

Message Element	Meaning
Subsystem	*Subsystem* is an abbreviation for a router component such as a protocol, packet forwarder, or interface. In the example above, **GW** identifies the subsystem (gateway) through which this event occurred. Examples of subsystems include ARP, IP, TKR, and X25. On a router, the subsystems depend on the hardware and software configured for that router. You can use the ELS **list subsystem** command to list the subsystems that are configured on your router.
Event Number	*Event Number* is a number that is assigned to each message within a subsystem. In the example above, the event number is **19** (within the GW subsystem). The event number always appears with the subsystem abbreviation, for example, **GW.019**. The subsystem and event number together identify an *individual* event. You can use the ELS **list subsystem** command to list the event within a subsystem.
Logging Level	*Logging Level* is a field that classifies each message by the type of event that generated it. Logging levels are as follows: **Logging Level** – **Type** UI – ERROR – Unusual internal errors CI – ERROR – Common internal errors UE – ERROR – Unusual external errors CE – ERROR – Common external errors ERROR – Includes all error levels above U–INFO – Unusual Informational comment C–INFO – Common Informational comment INFO – Includes all comment levels above STANDARD – Includes all error levels and all comment levels (default) P–TRACE – Per packet trace U–TRACE – Unusual operation packet trace message C–TRACE – Common operation packet trace message TRACE – Includes all trace levels above ALL – Includes all logging levels

Table 1–1 Message Descriptions (Cont.)

Message Element	Meaning
Message Text	Message text appears on the console screen in short form. In the sections that follow, variables such as ***source_address***, or ***network***, are replaced with actual data when the message displays on the console. These and other variables are replaced in the message text. The variable ***error_code*** appearing in the message description (usually preceded by "rsn" or "reason"), indicates the type of packet error detected. The next section describes the error and packet completion codes.

Code	Meaning
0	Packet successfully queued for output
1	Random, unidentified error
2	Packet not queued for output due to flow control reasons
3	Packet not queued because network is down
4	Packet not queued to avoid looping or bad broadcast
5	Packet not queued because destination host is down (only on networks where this can be detected)

When you send out an SNMP query, the response you get from the router is usually a 12-digit number, such as **`1.3.6.1.4.1.1.1.3.4.85.31.`** This number refers to various information regarding your query, such as the ELS operating number (**`1.3.6.1.4.1.1.1.3`**), the object (**`.4`**), the ELS subsystem number (**`.85`**), and the event number (**`.31`**). For example, the eleventh digit, **`.85`**, is the numerical equivalent to the subsystem element X.25. The following list describes the +subsystem numerical equivalents.

Subsystem	Numeric Equivalent	Subsystem	Numeric Equivalent
APL	50	MSPF	18
AP2	53	PPP	95
ARP	5	RIP	15
BGP	104	RTMP	52

Subsystem	Numeric Equivalent	Subsystem	Numeric Equivalent
BOSS	106	R2MP	56
BR	74	SDLC	90
BTP	14	SL	83
DLS	107	SNMP	21
DN	25	SPF	17
DNAV	43	SRLY	75
DVM	22	SRT	72
EGP	16	STP	73
ESIS	41	TCP	12
ETH	81	TFTP	19
FDDI	88	TKR	84
FLT	2	UDP	13
FR	92	VN	60
GW	1	V25B	108
ICMP	11	WRS	101
IP	10	XN	30
IPPN	100	X25	85
IPX	35	X251	96
ISIS	42	X252	97
ISO	40	X253	98
LLC	103	ZIP	51
LNM	102	ZIP2	54
MCF	105		

Error and Packet Completion Codes

The console displays the following Network information:

`nt 1 int Eth/0` or `network 1, interface Eth/0,` where:

- `1` is the network number (each network on the router is numbered sequentially from zero).
- `0` is the unit number (the interfaces of each hardware type are numbered sequentially from zero).

Address Type	Display Description
Ethernet and 802.5 hardware addresses	long hexadecimal number, such as 020701003e2c
IP (Internet Protocol) addresses	four decimal bytes separated by periods, such as 18.123.0.16
DECnet addresses	pair of decimal numbers separated by a period, such as 2.17

ELS Timestamp Prefix

The Event Logging System has a configurable Timestamp feature that indicates the time occurrence when reporting the network event in the ELS Log.

Configure this feature from either the ELS process or the CONFIG process (ELS config).

APL

AppleTalk Phase 1

This chapter describes AppleTalk® Phase 1 messages. For information about message content and how to use the message, refer to the Introduction.

APL.001

Level: P–TRACE

Short Syntax: APL.001 *source_net/source_node –> destination_net/destination_node* nt *network* ign

Long Syntax: APL.001 *source_net/source_node –> destination_net/destination_node* net *network* ignored

Description: An AppleTalk packet was recognized but ignored because AppleTalk forwarding is not in this load.

APL.002

Level: P–TRACE

Short Syntax: APL.002 */source_node –> /destination_node* nt *network* ign

Long Syntax: APL.002 */source_node –> /destination_node* net *network* ignored

Description: An AppleTalk packet with a short DDP header was recognized but ignored because the AppleTalk forwarding is not in this load.

APL.003

Level: P–TRACE

Short Syntax: APL.003 q ovf *src_net/src_node –> dest_net/dest_node* nt *network*

Long Syntax: APL.003 queue overflow *src_net/src_node –> dest_net/dest_node* net *network*

Description: The specified packet caused the forwarder input queue to overflow and was discarded.

APL.004

Level: P–TRACE

Short Syntax: APL.004 q ovf */src_node –> /dest_node* nt *network*

Long Syntax: APL.004 queue overflow */src_node –> /dest_node* net *network*

Description: The specified packet caused the forwarder input queue to overflow and was discarded.

APL.005

Level: UE–ERROR

Short Syntax: APL.005 pkt trnc *length* pkt ln *received_length src_net/src_node –> dst_net/dst_node*

Long Syntax: APL.005 packet truncated *length* packet length *received_length src_net/src_node –> dst_net/dst_node*

Description: The physical length of the packet as received was not long enough to contain a packet of the length claimed by the DDP header. Both lengths include only the DDP header and data, and do not include the LAP header of data–link header.

APL.006

Level: UE–ERROR

Short Syntax: APL.006 pkt trnc *length* pkt ln *received_length /src_node –> /dst_node* net *network*

Long Syntax: APL.006 packet truncated *length* packet length *received_length /src_node –> /dst_node* net *network*

Description: The physical length of the packet as received was not long enough to contain a packet of the length claimed by the DDP header. Both lengths include only the DDP header and data, and do not include the LAP header of data–link header.

APL.007

Level: UE–ERROR

Short Syntax: APL.007 bd hdr cksm frm *src_net/src_node*, rcv *rcvd_csum*, comp *comp_csum*

Long Syntax: APL.007 bad header checksum from *src_net/src_node*, received *rcvd_csum*, computed *comp_csum*

Description: The computed checksum of the specified packet did not match the checksum value in the DDP header.

APL.008

Level: U–INFO

Short Syntax: APL.008 no rte *src_net/src_node –> dest_net/dest_node*

Long Syntax: APL.008 no route *src_net/src_node –> dest_net/dest_node*

Description: No routing table entry was found for the destination net while trying to route the specified packet.

APL.009

Level: UE–ERROR

Short Syntax: APL.009 hp cnt ovf *src_net/src_node –> dest_net/dest_node*

Long Syntax: APL.009 hop count overflow *src_net/src_node –> dest_net/dest_node*

Description: The specified packet was discarded while attempting forwarding due to overflow of the packet hop count.

Cause: Packets whose hop counts overflow are typically victims of a routing loop. This is usually a temporary condition.

Action: If the problem is excessive or persistent then check for improper network configuration

APL.010

Level: UI–ERROR

Short Syntax: APL.010 no iorb for copy

Long Syntax: APL.010 no i/o request block to copy packet

Description: The system was making a copy of a directed broadcast packet for internal processing of the packet, and was unable to allocate a system buffer to copy the packet. The packet will still be forwarded, but no local copy will be received.

Cause: There is a buffer shortage in the router. This may be a temporary condition.

APL.011

Level: UE–ERROR

Short Syntax: APL.011 bd nd addr *src_net/src_node –> dest_net/dest_node*

Long Syntax: APL.011 bad node address *src_net/src_node –> dest_net/dest_node*

Description: An illegal destination node address was detected while attempting to forward the specified packet.

APL.012

Level: P–TRACE

Short Syntax: APL.012 *src_net/src_node –> dest_net/dest_node*

Long Syntax: APL.012 *src_net/src_node –> dest_net/dest_node*

Description: The specified AppleTalk packet was forwarded.

APL.013

Level: UI–ERROR

Short Syntax: APL.013 pkt too lg *pkt_len* > *max_len* nt *network src_net/src_node –> dest_net/dest_node*

Long Syntax: APL.013 packet too large *pkt_len* > *max_len* net *network src_net/src_node –> dest_net/dest_node*

Description: A packet exceeded the maximum length of a packet that was on the outgoing network and was discarded.

APL.014

Level: UI–ERROR

Short Syntax: APL.014 pkt *src_net/src_node –> dest_net/dest_node* dsc, rsn *code*

Long Syntax: APL.014 packet *src_net/src_node –> dest_net/dest_node* discarded, reason *code*

Description: An outgoing packet was not successfully transmitted for the reason indicated by the error code.

APL.015

Level: U–INFO

Short Syntax: APL.015 unrcgnzd LAP typ *lap_type*

Long Syntax: APL.015 unrecognized LAP type *lap_type*

Description: A packet was received with an unknown LAP type.

APL.016

Level: UE–ERROR

Short Syntax: APL.016 rsv bits *bits* not 0 pkt */src_node –> /dest_node*

Long Syntax: APL.016 Short DDP header reserved bits *bits* not 0 packet */src_node –> /dest_node*

Description: A short header DDP packet was received with the reserved bits above the length not zero.

APL.017

Level: UE–ERROR

Short Syntax: APL.017 bad dst skt *socket*

Long Syntax: APL.017 bad destination socket *socket*

Description: A locally destined packet had a destination socket on which there was no listener.

APL.018

Level: UE–ERROR

Short Syntax: APL.018 unk prt tp *type*

Long Syntax: APL.018 unkown protocol type *type*

Description: A locally destined packet had an unrecognized value in the protocol type field.

APL.019

Level: UE–ERROR

Short Syntax: APL.019 no uniq nd addr avial nt *network*

Long Syntax: APL.019 no unique node address avialable net *network*

Description: The handler was unable to find a unique node address available on this network.

Cause: There already exist the maximum of 254 nodes on the network; all the node numbers are taken.

APL.020

Level: C–INFO

Short Syntax: APL.020 nd addr assgnd *node_number* nt *network*

Long Syntax: APL.020 node address assigned *node_number* net *network*

Description: The indicated node address has been assigned to the specified interface.

APL.021

Level: C–INFO

Short Syntax: APL.021 intfc up *net_num/node_num* nt *network*

Long Syntax: APL.021 interface up *net_num/node_num* net *network*

Description: The specified interface has secured both a net and node address, and is now up.

APL.022

Level: C–INFO

Short Syntax: APL.022 intfc up *net_num/node_num* zn *zone_name* nt *network*

Long Syntax: APL.022 interface up *net_num/node_num* zone *zone_name* net *network*

Description: The specified interface has secured both a net and node address, and is now up.

APL.023

Level: UE–ERROR

Short Syntax: APL.023 NBP bd cnt *tuple_count* in BrRq frm */src_node* nt *network*

Long Syntax: APL.023 NBP bad count *tuple_count* in BrRq from */src_node* net *network*

Description: The NBP Broadcast Request packet from the specified host contained a tuple count not equal to 1.

APL.024

Level: P–TRACE

Short Syntax: APL.024 NBP BrRq rcvd frm */src_node* nt *network*

Long Syntax: APL.024 NBP BrRq received from */src_node* net *network*

Description: An NBP Broadcast Request packet was received from the specified host.

APL.025

Level: U–INFO

Short Syntax: APL.025 no knwn zn nm for nt *net_num* in NBP BrRq frm */src_node*

Long Syntax: APL.025 no known zone name for net *net_num* in NBP BrRq from */src_node*

Description: An associated zone name for the requested net in a BrRq packet was not found.

APL.026

Level: U–INFO

Short Syntax: APL.026 zn *zone_name* not fnd in ZIT, NBP BrRq frm */src_node*

Long Syntax: APL.026 zone *zone_name* not found in ZIT, NBP BrRq from */src_node*

Description: The requested zone in BrRq from the specified host was not found in the Zone Information Table.

APL.027

Level: UI–ERROR

Short Syntax: APL.027 no mem for NBP LkUp

Long Syntax: APL.027 no memory for NBP LookUp

Description: An iorb was not available for an NBP LookUp packet.

APL.028

Level: UI–ERROR

Short Syntax: APL.028 NBP LkUp disc nt *network* rsn *error_code*

Long Syntax: APL.028 NBP LookUp discarded net *network* reason *error_code*

Description: An NBP LookUp was not sent for the indicated reason.

APL.029

Level: P–TRACE

Short Syntax: APL.029 NBP LkUp snt for net *net_number*

Long Syntax: APL.029 NBP LookUp sent for net *net_number*

Description: An NBP LookUp was sent as a directed broadcast on the indicated net.

APL.030

Level: UI–ERROR

Short Syntax: APL.030 no mem for NBP stat block, BrRq frm */src_node* ign

Long Syntax: APL.030 no memory for NBP status block, BrRq from */src_node* ign

Description: No memory was available for status block to process NBP BrRq from the indicated host.

APL.031

Level: UI–ERROR

Short Syntax: APL.031 no mem for AARP Probe

Long Syntax: APL.031 no memory for AARP Probe

Description: An iorb was not available for an AARP Probe packet.

APL.032

Level: UI–ERROR

Short Syntax: APL.032 AARP Probe disc nt *network* rsn *error_code*

Long Syntax: APL.032 AARP Probe discarded net *network* reason *error_code*

Description: An Apple ARP Probe was not sent for the indicated reason.

APL.033

Level: P–TRACE
Short Syntax: APL.033 AARP Probe snt nt *network*
Long Syntax: APL.033 AARP Probe sent net *network*
Description: An Apple ARP Probe was sent on the indicated net.

APL.034

Level: C–INFO
Short Syntax: APL.034 AARP Rsps match tentative addr, new addr selected nt *network*
Long Syntax: APL.034 AARP Response match tentative addr, new addr selected nt *network*
Description: An Apple ARP Response was received in response to our probe claiming the tentative address. A new node address was selected for continued probing.

APL.035

Level: UE–ERROR
Short Syntax: APL.035 Unrec AARP pkt typ *arp_type* rcvd nt *network*
Long Syntax: APL.035 Unrecognized AARP packet type *arp_type* received net *network*
Description: An Apple ARP packet with an unrecognized type was received.

APL.036

Level: P–TRACE
Short Syntax: APL.036 AARP Probe rcvd nt *network*
Long Syntax: APL.036 AARP Probe received net *network*
Description: An Apple ARP Probe packet was received.

APL.037

Level: UI–ERROR
Short Syntax: APL.037 AARP Response disc nt *network* rsn *error_code*
Long Syntax: APL.037 AARP Response discarded net *network* reason *error_code*
Description: An Apple ARP Response was not sent for the indicated reason.

APL.038

Level: P–TRACE
Short Syntax: APL.038 AARP Response snt nt *network*
Long Syntax: APL.038 AARP Response sent net *network*
Description: An Apple ARP Response to a probe was sent on the indicated net.

APL.039

Level: UE–ERROR

Short Syntax: APL.039 Echo pkt short (*length*) frm *src_net/src_node* nt *network*

Long Syntax: APL.039 Echo packet too short (*length* bytes) from *src_net/src_node* net *network*

Description: An Echo packet, that was too short to contain the echo packet header, was received.

APL.040

Level: U–TRACE

Short Syntax: APL.040 Echo pkt, func *function_code*, frm *src_net/src_node* nt *network*

Long Syntax: APL.040 Echo packet, echo function *function_code*, received from *src_net/src_node* net *network*

Description: An Echo Protocol packet, which was not a request, was received from the specified node. It will not be answered. If function was 2, it was an Echo Reply.

APL.041

Level: P–TRACE

Short Syntax: APL.041 Echo Req frm *src_net/src_node* nt *network*, rplyng

Long Syntax: APL.041 Echo Request from *src_net/src_node* net *network*, replying

Description: An Echo Request packet was received from the specified host. A reply will be sent.

APL.042

Level: UE–ERROR

Short Syntax: APL.042 Echo pkt short (*length*) frm */src_node* nt *network*

Long Syntax: APL.042 Echo packet too short (*length* bytes) from */src_node* net *network*

Description: An Echo packet was received that was too short to contain the echo packet header.

APL.043

Level: U–TRACE

Short Syntax: APL.043 Echo pkt, func *function_code*, frm */src_node* nt *network*

Long Syntax: APL.043 Echo packet, echo function *function_code*, from */src_node* net *network*

Description: An Echo Protocol packet, which was not a request, was received from the specified node. It will not be answered. If function was 2, it was an Echo Reply.

APL.044

Level: P–TRACE

Short Syntax: APL.044 Echo Req frm */src_node* nt *network*, rplyng

Long Syntax: APL.044 Echo Request from */src_node* net *network*, replying

Description: An Echo Request packet was received from the specified host. A reply will be sent.

APL.045

Level: UI–ERROR

Short Syntax: APL.045 Echo Rply disc nt *network* rsn *error_code*

Long Syntax: APL.045 Echo Reply discarded net *network* reason *error_code*

Description: An Echo Reply was not sent for the indicated reason.

APL.046

Level: UE–ERROR

Short Syntax: APL.046 Bad LAP dst *LAP_dest_node* frm *LAP_source_node* net *network*

Long Syntax: APL.046 Bad LAP destination *LAP_dest_node* from *LAP_source_node* net *network*

Description: A packet was received whose LAP (or ELAP) destination was not the broadcast address, and was not equal to the AppleTalk address of this network.

APL.047

Level: UE–ERROR

Short Syntax: APL.047 pkt too short (*length*) dst *LAP_dest_node* frm *LAP_source_node* net *network*

Long Syntax: APL.047 Long DDP packet too short for header (*length* bytes) dst *LAP_dest_node* from *LAP_source_node* net *network*

Description: A long format DDP packet has been received that is shorter than the length of a long DDP header (13 bytes).

APL.048

Level: UE–ERROR

Short Syntax: APL.048 pkt too long (*length*) *src_net/src_node –> dst_net/dst_node*

Long Syntax: APL.048 Long DDP packet too long (*length* bytes) *src_net/src_node –> dst_net/dst_node*

Description: A long format DDP packet has been received with more than the limit of 586 bytes of data after the DDP header.

APL.049

Level: UE–ERROR

Short Syntax: APL.049 DDP rsvd bits *src_net/src_node –> dst_net/dst_node*

Long Syntax: APL.049 Long DDP packet reserved bit(s) set *src_net/src_node –> dst_net/dst_node*

Description: A long format DDP packet has been received with one (or more) of the two reserved bits above the hop count set.

APL.050

Level: UE–ERROR

Short Syntax: APL.050 pkt too short (*length*) dst *LAP_dest_node* frm *LAP_source_node* net *network*

Long Syntax: APL.050 Short DDP packet too short for header (*length* bytes) dst *LAP_dest_node* from *LAP_source_node* net *network*

Description: A short format DDP packet has been received that is shorter than the length of a long DDP header (5 bytes).

APL.051

Level: UE–ERROR

***Short Syntax*:** APL.051 pkt too long (*length*) dst *LAP_dest_node* frm *LAP_source_node* net *network*

***Long Syntax*:** APL.051 Short DDP packet too long (*length* bytes) dst *LAP_dest_node* from *LAP_source_node* net *network*

Description: A short format DDP packet has been received with more than the limit of 586 bytes of data after the DDP header.

APL.052

Level: UI–ERROR

***Short Syntax*:** APL.052 no mem for copy buf

***Long Syntax*:** APL.052 No memory for buffer to copy packet

Description: The system was making a copy of a directed broadcast packet for internal processing of the packet, and was unable to allocate memory to copy the packet. The packet will still be forwarded, but no local copy will be received.

Cause: There is a memory shortage in the router. This may be a temporary condition.

APL.053

Level: UE–ERROR

***Short Syntax*:** APL.053 NBP shrt (*length*) frm */src_node* nt *network*

***Long Syntax*:** APL.053 NBP short (*length* bytes) from */src_node* nt *network*

Description: A NBP packet was received that was too short to contain the NBP header. The packet will be discarded.

APL.054

Level: U–INFO

***Short Syntax*:** APL.054 NBP bd func *function* frm */src_node* nt *network*

***Long Syntax*:** APL.054 NBP bad function *function* from */src_node* nt *network*

Description: A NBP packet was received with a bad function code. Only BrReq packets are processed, LkUp and LkUp–Reply packets are ignored silently. The packet will be discarded.

APL.055

Level: UE–ERROR

Short Syntax: APL.055 NBP trnc (*length*) frm */src_node* nt *network*

Long Syntax: APL.055 NBP truncated (*length* bytes) from */src_node* nt *network*

Description: A NBP packet, that is too short to contain the NBP data was received. The packet will be discarded.

APL.056

Level: P–TRACE

Short Syntax: APL.056 *source_net/source_node –> destination_net/destination_node* nt *network* ign

Long Syntax: APL.056 *source_net/source_node –> destination_net/destination_node* net *network* ignored

Description: An AppleTalk packet was recognized but ignored because AppleTalk forwarding was not enabled on the interface.

APL.057

Level: P–TRACE

Short Syntax: APL.057 */source_node –> /destination_node* nt *network* ign

Long Syntax: APL.057 */source_node –> /destination_node* net *network* ignored

Description: An AppleTalk packet with a short DDP header was recognized but ignored because AppleTalk forwarding was not enabled on the interface.

APL.058

Level: UE–ERROR

Short Syntax: APL.058 NBP ilg zn len length frm */src_node* nt *network*

Long Syntax: APL.058 NBP ilg zn len length from */src_node* nt *network*

Description: A NBP packet was received that has a zone name more than 32 characters long. The packet will be discarded.

APL.059

Level: UI–ERROR

Short Syntax: APL.059 Ilg zone *zone_name* seed w/o net seed nt *network*

Long Syntax: APL.059 Illegal zone *zone_name* seed without network seed net *network*

Description: The user configured a zone name for a network for which no network number was configured. The zone name will be ignored.

AP2

AppleTalk Phase 2

This chapter describes AppleTalk® Phase 2 messages. For information about message content and how to use the message, refer to the Introduction.

AP2.001

Level: P–TRACE

Short Syntax: AP2.001 *source_net/source_node –> destination_net/destination_node* nt *network* ign

Long Syntax: AP2.001 *source_net/source_node –> destination_net/destination_node* net *network* ignored

Description: An AppleTalk packet was recognized but ignored because AppleTalk forwarding is not in this load.

AP2.002

Deleted: Message deleted.

AP2.003

Level: P–TRACE

Short Syntax: AP2.003 q ovf *src_net/src_node –> dest_net/dest_node* nt *network*

Long Syntax: AP2.003 queue overflow *src_net/src_node –> dest_net/dest_node* net *network*

Description: The specified packet caused the forwarder input queue to overflow and was discarded.

AP2.004

Deleted: Message deleted.

AP2.005

Level: UE–ERROR

Short Syntax: AP2.005 pkt trnc *length* pkt ln *received_length src_net/src_node –> dst_net/dst_node*

Long Syntax: AP2.005 packet truncated *length* packet length *received_length src_net/src_node –> dst_net/dst_node*

Description: The physical length of the packet as received was not long enough to contain a packet of the length claimed by the DDP header. Both lengths include only the DDP header and data, and do not include the LAP header of data–link header.

AP2.006

Deleted: Message deleted.

AP2.007

Level: UE–ERROR

Short Syntax: AP2.007 bd hdr cksm frm *src_net/src_node*, rcv *rcvd_csum*, comp *comp_csum*

Long Syntax: AP2.007 bad header checksum from *src_net/src_node*, received *rcvd_csum*, computed *comp_csum*

Description: The computed checksum of the specified packet did not match the checksum value in the DDP header.

AP2.008

Level: U–INFO

Short Syntax: AP2.008 no rte *src_net/src_node –> dest_net/dest_node*

Long Syntax: AP2.008 no route *src_net/src_node –> dest_net/dest_node*

Description: No routing table entry was found for the destination net while trying to route the specified packet.

AP2.009

Level: UE–ERROR

Short Syntax: AP2.009 hp cnt ovf *src_net/src_node –> dest_net/dest_node*

Long Syntax: AP2.009 hop count overflow *src_net/src_node –> dest_net/dest_node*

Description: The specified packet was discarded while attempting forwarding due to overflow of the packet hop count.

Cause: Packets whose hop counts overflow are typically victims of a routing loop. This is usually a temporary condition.

Action: If the problem is excessive or persistent then check for improper network configuration.

AP2.010

Level: UI–ERROR

Short Syntax: AP2.010 no iorb for copy

Long Syntax: AP2.010 no i/o request block to copy packet

Description: The system was making a copy of a directed broadcast packet for internal processing of the packet, and was unable to allocate a system buffer to copy the packet. The packet will still be forwarded, but no local copy will be received.

Cause: There is a buffer shortage in the router. This may be a temporary condition.

AP2.011

Level: UI–ERROR

Short Syntax: AP2.011 No RTMP entry for FwdReq pkt to nt *dest_net*, rcvd nt *network*

Long Syntax: AP2.011 No RTMP entry for FwdReq pkt to net *dest_net*, received net *network*

Description: An Apple NBP Forward request packet was received and either RTMP has no entry for the network or the net is no longer directly connected.

AP2.012

Level: P–TRACE

Short Syntax: AP2.012 *src_net/src_node –> dest_net/dest_node*

Long Syntax: AP2.012 *src_net/src_node –> dest_net/dest_node*

Description: The specified AppleTalk packet was forwarded.

AP2.013

Level: UI–ERROR

Short Syntax: AP2.013 pkt too lg *pkt_len* > *max_len* nt *network src_net/src_node –> dest_net/dest_node*

Long Syntax: AP2.013 packet too large *pkt_len* > *max_len* net *network src_net/src_node –> dest_net/dest_node*

Description: A packet exceeded the maximum length of a packet on the outgoing network and was discarded.

AP2.014

Level: UI–ERROR

Short Syntax: AP2.014 pkt *src_net/src_node –> dest_net/dest_node* dsc, rsn *code*

Long Syntax: AP2.014 packet *src_net/src_node –> dest_net/dest_node* discarded, reason *code*

Description: An outgoing packet was not successfully transmitted for the reason indicated by the error code.

AP2.015

Deleted: Message deleted.

AP2.016

Deleted: Message deleted.

AP2.017

Level: UE–ERROR

Short Syntax: AP2.017 bad dst skt *socket*

Long Syntax: AP2.017 bad destination socket *socket*

Description: A locally destined packet had a destination socket on which there was no listener.

AP2.018

Level: UE–ERROR

Short Syntax: AP2.018 unk prt tp *type*

Long Syntax: AP2.018 unkown protocol type *type*

Description: A locally destined packet had an unrecognized value in the protocol type field.

AP2.019

Level: UE–ERROR

*Short Syntax***:** AP2.019 no uniq nd addr avial nt *network*

*Long Syntax***:** AP2.019 no unique node address available net *network*

Description: The handler was unable to find a unique node address available on this network.

Cause: There already exist the maximum number of nodes on the *network*; all node numbers are taken. The net range should be extended.

AP2.020

Level: C–INFO

*Short Syntax***:** AP2.020 nt/nd addr assgnd *net_number/node_number* nt *network*

*Long Syntax***:** AP2.020 net/node address assigned *net_number/node_number* net *network*

Description: The indicated net / node address has been assigned to the specified interface.

AP2.021

Level: C–INFO

*Short Syntax***:** AP2.021 intfc up *net_num/node_num* nt *network*

*Long Syntax***:** AP2.021 interface up *net_num/node_num* net *network*

Description: The specified interface has secured both a net and node address, and is now up and looking for a zone name.

AP2.022

Level: C–INFO

*Short Syntax***:** AP2.022 intfc up *net_num/node_num* zn *zone_name* nt *network*

*Long Syntax***:** AP2.022 interface up *net_num/node_num* zone *zone_name* net *network*

Description: The specified interface has secured a net, node and zone name, and is now up.

AP2.023

Deleted: Message deleted.

AP2.024

Deleted: Message deleted.

AP2.025

Deleted: Message deleted.

AP2.026

Deleted: Message deleted.

AP2.027

Level: UI–ERROR
Short Syntax: AP2.027 no mem for NBP LkUp
Long Syntax: AP2.027 no memory for NBP LookUp
Description: An iorb was not available for an NBP LookUp packet.

AP2.028

Level: UI–ERROR
Short Syntax: AP2.028 NBP LkUp disc nt *network* rsn *error_code*
Long Syntax: AP2.028 NBP LookUp discarded net *network* reason *error_code*
Description: An NBP LookUp was not sent for the indicated reason.

AP2.029

Level: P–TRACE
Short Syntax: AP2.029 NBP LkUp/FwdReq snt to net *net_number*
Long Syntax: AP2.029 NBP LookUp or FwdReq sent to net *net_number*
Description: An NBP LookUp or FwdReq was sent to the indicated net.

AP2.030

Deleted: Message deleted.

AP2.031

Level: UI–ERROR
Short Syntax: AP2.031 no mem for AARP Probe
Long Syntax: AP2.031 no memory for AARP Probe
Description: A buffer was not available for an AARP Probe packet.

AP2.032

Level: UI–ERROR
Short Syntax: AP2.032 AARP Probe disc nt *network* rsn *error_code*
Long Syntax: AP2.032 AARP Probe discarded net *network* reason *error_code*
Description: An Apple ARP Probe was not sent for the indicated reason.

AP2.033

Level: P–TRACE
Short Syntax: AP2.033 AARP Probe snt nt *network*
Long Syntax: AP2.033 AARP Probe sent net *network*
Description: An Apple ARP Probe was sent on the indicated net.

AP2.034

Level: C–INFO
Short Syntax: AP2.034 AARP Rsps match tentative addr, new addr selected nt *network*
Long Syntax: AP2.034 AARP Response match tentative addr, new addr selected nt *network*
Description: An Apple ARP Response was received in response to our probe claiming the tentative address. A new node address was selected for continued probing.

AP2.035

Level: UE–ERROR
Short Syntax: AP2.035 Unrec AARP pkt typ *arp_type* rcvd nt *network*
Long Syntax: AP2.035 Unrecognized AARP packet type *arp_type* received net *network*
Description: An Apple ARP packet with an unrecognized type was received.

AP2.036

Level: P–TRACE
Short Syntax: AP2.036 AARP Probe rcvd *src_net/src_node* nt *network*
Long Syntax: AP2.036 AARP Probe received *src_net/src_node* net *network*
Description: An Apple ARP Probe packet was received.

AP2.037

Level: UI–ERROR
Short Syntax: AP2.037 AARP Response disc nt *network* rsn *error_code*
Long Syntax: AP2.037 AARP Response discarded net *network* reason *error_code*
Description: An Apple ARP Response was not sent for the indicated reason.

AP2.038

Level: P–TRACE

Short Syntax: AP2.038 AARP Response snt nt *network*

Long Syntax: AP2.038 AARP Response sent net *network*

Description: An Apple ARP Response to a probe was sent on the indicated net.

AP2.039

Level: UE–ERROR

Short Syntax: AP2.039 Echo pkt short (*length*) frm *src_net/src_node* nt *network*

Long Syntax: AP2.039 Echo packet too short (*length* bytes) from *src_net/src_node* net *network*

Description: An Echo packet was received that was too short to contain the echo packet header.

AP2.040

Level: U–TRACE

Short Syntax: AP2.040 Echo pkt, func *function_code*, frm *src_net/src_node* nt *network*

Long Syntax: AP2.040 Echo packet, echo function *function_code*, received from *src_net/src_node* net *network*

Description: An Echo Protocol packet, which was not a request, was received from the specified node. It will not be answered. If function was 2, it was an Echo Reply.

AP2.041

Level: P–TRACE

Short Syntax: AP2.041 Echo Req frm *src_net/src_node* nt *network*, rplyng

Long Syntax: AP2.041 Echo Request from *src_net/src_node* net *network*, replying

Description: An Echo Request packet was received from the specified host. A reply will be sent.

AP2.042

Deleted: Message deleted.

AP2.043

Deleted: Message deleted.

AP2.044

Deleted: Message deleted.

AP2.045

Level: UI–ERROR

Short Syntax: AP2.045 Echo Rply disc nt *network* rsn *error_code*

Long Syntax: AP2.045 Echo Reply discarded net *network* reason *error_code*

Description: An Echo Reply was not sent for the indicated reason.

AP2.046

Deleted: Message deleted.

AP2.047

Level: UE–ERROR

Short Syntax: AP2.047 pkt too short (*length*) net *network*

Long Syntax: AP2.047 Long DDP packet too short for header (*length* bytes) net *network*

Description: A long format DDP packet has been received that is shorter than the length of a long DDP header (13 bytes).

AP2.048

Level: UE–ERROR

Short Syntax: AP2.048 pkt too long (*length*) *src_net/src_node –> dst_net/dst_node*

Long Syntax: AP2.048 Long DDP packet too long (*length* bytes) *src_net/src_node –> dst_net/dst_node*

Description: A long format DDP packet has been received with more than the limit of 586 bytes of data after the DDP header.

AP2.049

Level: UE–ERROR

Short Syntax: AP2.049 DDP rsvd bits *src_net/src_node –> dst_net/dst_node*

Long Syntax: AP2.049 Long DDP packet reserved bit(s) set *src_net/src_node –> dst_net/dst_node*

Description: A long format DDP packet has been received with one (or more) of the two reserved bits above the hop count set.

AP2.050

Deleted: Message deleted.

AP2.051

Deleted: Message deleted.

AP2.052

Deleted: Message deleted.

AP2.053

Deleted: Message deleted.

AP2.054

Deleted: Message deleted.

AP2.055

Deleted: Message deleted.

AP2.056

Level: P–TRACE

*Short Syntax***:** AP2.056 *source_net/source_node –> destination_net/destination_node* nt *network* ign

*Long Syntax***:** AP2.056 *source_net/source_node –> destination_net/destination_node* net *network* ignored

Description: An AppleTalk packet was recognized but ignored because AppleTalk forwarding was not enabled on the interface.

AP2.057

Deleted: Message deleted.

AP2.058

Deleted: Message deleted.

AP2.059

Level: UI–ERROR

Short Syntax: AP2.059 Ilg zone *zone_name* seed w/o net seed nt *network*

Long Syntax: AP2.059 Illegal zone *zone_name* seed without network seed net *network*

Description: The user configured a zone name for a network in which no network number was configured. The zone name will be ignored.

AP2.060

Level: UE–ERROR

Short Syntax: AP2.060 NBP bd cnt *tuple_count* in BrRq frm *src_net/src_node* nt *network*

Long Syntax: AP2.060 NBP bad count *tuple_count* in BrRq from *src_net/src_node* net *network*

Description: The NBP Broadcast Request packet from the specified host contained a tuple count not equal to 1.

AP2.061

Level: P–TRACE

Short Syntax: AP2.061 NBP BrRq rcvd frm *src_net/src_node* nt *network*

Long Syntax: AP2.061 NBP BrRq received from *src_net/src_node* net *network*

Description: An NBP Broadcast Request packet was received from the specified host.

AP2.062

Level: U–INFO

Short Syntax: AP2.062 no knwn zn nm for nt *net_num* in NBP BrRq frm *src_net/src_node*

Long Syntax: AP2.062 no known zone name for net *net_num* in NBP BrRq from *src_net/src_node*

Description: An associated zone name for the requested net in a BrRq packet was not found.

AP2.063

Level: U–INFO

Short Syntax: AP2.063 zn *zone_name* not fnd in ZIT, NBP BrRq frm *src_net/src_node*

Long Syntax: AP2.063 zone *zone_name* not found in ZIT, NBP BrRq from *src_net/src_node*

Description: The requested zone in BrRq from the specified host was not found in the Zone Information Table.

AP2.064

Level: UI–ERROR

Short Syntax: AP2.064 no mem for NBP stat block, BrRq frm *src_net/src_node* ign

Long Syntax: AP2.064 no memory for NBP status block, BrRq from *src_net/src_node* ign

Description: No memory was available for status block to process NBP BrRq from the indicated host.

AP2.065

Level: UE–ERROR

Short Syntax: AP2.065 NBP shrt (*length*) frm *src_net/src_node* nt *network*

Long Syntax: AP2.065 NBP short (*length* bytes) from *src_net/src_node* nt *network*

Description: An NBP packet was received that is too short to contain the NBP header. The packet will be discarded.

AP2.066

Level: UE–ERROR

Short Syntax: AP2.066 NBP bd func *function* frm *src_net/src_node* nt *network*

Long Syntax: AP2.066 NBP bad function *function* from *src_net/src_node* nt *network*

Description: An NBP packet was received with a bad function code. Only BrReq and FwdReq packets are processed, LkUp and LkUp–Reply packets are ignored silently. The packet will be discarded.

AP2.067

Level: UE–ERROR

Short Syntax: AP2.067 NBP trnc (*length*) frm *src_net/src_node* nt *network*

Long Syntax: AP2.067 NBP truncated (*length* bytes) from *src_net/src_node* nt *network*

Description: An NBP packet was received that is too short to contain the NBP data. The packet will be discarded.

AP2.068

Level: UE–ERROR

Short Syntax: AP2.068 NBP ilg zn len length frm *src_net/src_node* nt *network*

Long Syntax: AP2.068 NBP ilg zn len length from *src_net/src_node* nt *network*

Description: An NBP packet was received that has a zone name more than 32 characters long. The packet will be discarded.

ARP

Address Resolution Protocol

This chapter describes Address Resolution Protocol (ARP) messages. For information about message content and how to use the message, refer to the Introduction.

ARP.001

Level: U–INFO

Short Syntax: ARP.001 Q ovf nt *network*

Long Syntax: ARP.001 Queue overflow net *network*

Description: An ARP packet was discarded, rather than being queued, because the queue of unprocessed ARP packets was too long. This means that ARP packets are arriving faster than they can be processed. Note that this event does not get counted in ELS, it is instead counted in the ARP console. The counters (kept per input network) can be read using the ARP>STATISTICS command, in the ‘‘input packet overflows’’ section.

Cause: This is often a symptom of a so–called ‘‘ARP storm’’. Some packets (usually an IP broadcast) arrive at hosts (usually a popular workstation) which do not recognize the destination address; they then attempt (in contravention of the Host specification) to forward the packet, but to do so they need the ARP mapping. Since they all receive the broadcast at the same time, they all attempt to forward the packet at the same time, and all do an ARP request at the same time.

Action: Prevail on the appropriate host manufacturer to bring their software into compliance with the specification. In the short term, it may be possible to disable the source of the packets, or cause it to use an address that the misbehaving hosts do recognize as a broadcast.

ARP.002

Level: P–TRACE

Short Syntax: ARP.002 Pkt in *operation_type hardware_address_space protocol_type* nt *network ID*

Long Syntax: ARP.002 Packet received *operation_type hardware_address_space protocol_type* net *network ID*

Description: An ARP packet of the type indicated has just arrived for processing.

ARP.003

Level: U–INFO

Short Syntax: ARP.003 Unkwn hdw *hardware_address_space* nt *network ID*

Long Syntax: ARP.003 Unknown hardware space *hardware_address_space* net *network ID*

Description: An incoming ARP packet was received on a network which is not using ARP for address translation in any protocol.

Cause: The gateway is misconfigured.

Action: Correct the configuration.

Cause: A protocol is in use on that network which requires the use of ARP, but the router does not support that protocol.

Action: None.

ARP.004

Level: UE–ERROR

Short Syntax: ARP.004 Bd hdw *hardware_address_space hardware_address_length* nt *network ID*

Long Syntax: ARP.004 Bad hardware address space *hardware_address_space hardware_address_length* nt *network ID*

Description: An incoming ARP packet was received with a hardware address space code or hardware address length which does not match the one which should be used on that network.

Cause: This is probably caused by an error (possible a byte swap problem) in some other equipment on the network.

Action: Use a network management tool to detect the source host and contact the manufacturer of the equipment and report the problem.

ARP.005

Level: P–TRACE

Short Syntax: ARP.005 Unkwn prt *protocol_type* nt *network ID*

Long Syntax: ARP.005 Unknown protocol type *protocol_type* net *network ID*

Description: An incoming ARP packet was received for a protocol for which the router is not using ARP for address translation.

Cause: The gateway is misconfigured.

Action: Correct the configuration.

Cause: A protocol is in use on that network which requires the use of ARP, but the router does not support that protocol.

Action: None.

ARP.006

Level: UE–ERROR

Short Syntax: ARP.006 Bd prt *protocol_type protocol_address_length* nt *network ID*

Long Syntax: ARP.006 Bad protocol address length *protocol_type protocol_address_length* net *network ID*

Description: An incoming ARP packet was received with a protocol address length which does not match the one which should be used on that network.

Cause: This is probably caused by an error (possible a byte swap problem) in some other equipment on the network.

Action: Use a network management tool to detect the source host and contact the manufacturer of the equipment and report the problem.

ARP.007

Level: U–TRACE

Short Syntax: ARP.007 Mk ent *hardware_address_space protocol_type* nt *network ID*

Long Syntax: ARP.007 Make translation entry *hardware_address_space protocol_type* net *network ID*

Description: An incoming ARP packet addressed to this host contained a mapping which was not in the translation cache. A new cache entry was filled in with the information in the packet.

ARP.008

Level: UE–ERROR

Short Syntax: ARP.008 Bd opc *operation_type hardware_address_space protocol_type* nt *network ID*

Long Syntax: ARP.008 Bad operation code *operation_type hardware_address_space protocol_type* net *network ID*

Description: An incoming ARP packet was received with an illegal operation code.

Cause: This is probably caused by an error (possibly a byte swap problem) in some other equipment on the network.

Action: Use a network management tool to detect the source host and contact the manufacturer of the equipment and report the problem.

ARP.009

Level: U–TRACE

Short Syntax: ARP.009 Rply *hardware_address_space protocol_type* nt *network ID*

Long Syntax: ARP.009 Reply sent *hardware_address_space protocol_type* net *network ID*

Description: An ARP reply is being sent as the result of a request for a translation from another host.

ARP.010

Level: UI–ERROR

Short Syntax: ARP.010 Err on rply nt *network ID*

Long Syntax: ARP.010 Transmission error on sending reply net *network ID*

Description: An outgoing reply packet was dropped as the result of some problem in the router.

Cause: There are many potential causes of this problem; an overloaded output queue, a down network, etc.

Action: Consult logging output from the relevant *network* subsystem for more information.

ARP.011

Level: U–TRACE

Short Syntax: ARP.011 Del ent *hardware_address_space protocol_type* nt *network ID*

Long Syntax: ARP.011 Deleting translation entry *hardware_address_space protocol_type* net *network ID*

Description: A translation cache entry timed out (which was not used or refreshed recently) has been deleted. Consult the ARP manual for more details on controlling this process.

ARP.012

Level: UI–ERROR

Short Syntax: ARP.012 No iorb fr rqst nt *network ID*

Long Syntax: ARP.012 No buffer for outgoing request packet net *network ID*

Description: An outgoing reply packet was dropped as the result of a lack of buffers in the router.

Cause: There are many potential causes of this problem; temporary overloads, etc.

Action: Consult logging output from the rest of the router for more information. If the problem persists, contact Customer Service.

ARP.013

Deleted: Message deleted at Release 13.0.

ARP.014

Level: U–TRACE

Short Syntax: ARP.014 Rqst *hardware_address_space protocol_type* nt *network ID*

Long Syntax: ARP.014 Translation request sent *hardware_address_space protocol_type* net *network ID*

Description: An ARP translation request is being sent as the result of the transmission of a packet from the router for which the translation of another host's address is needed.

ARP.015

Deleted: Message deleted.

ARP.016

Level:	P–TRACE
Short Syntax:	ARP.016 unkn dst prot ad nt *network ID*
Long Syntax:	ARP.016 Unknown destination protocol address net *network ID*
Description:	This message is generated when an ARP request specifies an unknown protocol address (i.e. request not for this router).
Cause:	ARP request for a host on this network that is not this router.
Action:	None needed. This is normal for the ARP protocol, all requests are sent as broadcasts.

ARP.017

Level:	UI–ERROR
Short Syntax:	ARP.017 Rqst send failed rsn *reason_code* nt *network ID*
Long Syntax:	ARP.017 Transmission of request failed for reason *reason_code* net *network ID*
Description:	An outgoing ARP request packet was dropped as the result of some problem in the router. The reason_code gives the cause.
Cause:	Miscellaneous handler error. (Reason code 1.)
Action:	Check for error messages from handler for *network_name.*
Cause:	Output queue overflow, or other flow control. (Reason code 2.)
Action:	Alleviate congestion.
Cause:	Network down. (Reason code 3.)
Action:	See why handler thinks network is down.
Cause:	Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)
Action:	Check configuration.
Cause:	Host down. (Reason code 5.)
Action:	See why handler thinks host is down.

BGP

Border Gateway Protocol

This chapter describes Border Gateway Protocol (BGP) messages. BGP is a routing protocol used by the IP protocol family. For information about message content and how to use the message, refer to the Introduction.

BGP.001

Level: UI–ERROR

Short Syntax: BGP.001 bgp sec code in opn wrg nbr *neighbor*

Long Syntax: BGP.001 bgp security code in open message is incorrect from neighbor *neighbor*

Description: The bgp RFC specifies only a single acceptable security code of 0. This message is printed if another code is received.

Cause: Neighbor sent a security code in the open message that is non null.

Action: Use a router that adheres more closely to the bgp specification.

BGP.002

Level: UI–ERROR

Short Syntax: BGP.002 bgp msg hdr len wrg nbr *neighbor*

Long Syntax: BGP.002 bgp message header length is incorrect from neighbor *neighbor*

Description: The speaker received a message in which the header length was incorrect.

Cause: Neighbor sent an open message which is of incorrect length.

Action: Use a router that adheres to the bgp specification.

BGP.003

Level: U–INFO

Short Syntax: BGP.003 no supprt for this bgp vers nbr *neighbor*

Long Syntax: BGP.003 No support for this bgp version request from neighbor *neighbor*

Description: The current version supported by BGP is version 4. No other version support exists. This message is printed when a neighbor requests a lower version of BGP.

Cause: Neighbor is requesting a version of BGP, which is unsupported.

Action: Neighbor router must be configured for the proper version.

BGP.004

Level: UI–ERROR

Short Syntax: BGP.004 mkr fld is wrg frm nbr *neighbor*

Long Syntax: BGP.004 marker field is incorrect from neighbor *neighbor*

Description: The only supported marker field is 16 octets of 0xff. This message is printed when anything but this is received.

Cause: Neighbor is using an incorrect marker field.

Action: Use a router that adheres to the BGP specification.

BGP.005

Level: UI–ERROR

Short Syntax: BGP.005 bad AS number frm nbr *neighbor*

Long Syntax: BGP.005 Bad AS number from neighbor *neighbor*

Description: This message is printed when the neighbor and this speaker have the same AS number. This of course is an illegal configuration.

Cause: Neighbor is using an AS that matches this one.

Action: Make sure that the neighbor and this router have properly configured AS.

BGP.006

Level: UI–ERROR

Short Syntax: BGP.006 bad BGP id frm nbr *neighbor*

Long Syntax: BGP.006 Bad BGP identifier from neighbor *neighbor*

Description: This message is printed when the neighbor and this speaker have the same BGP identifier. Since this is used to resolve tcp connection collisions, this is an illegal configuration.

Cause: Neighbor is using a BGP identifier that is the same as this one.

Action: Make sure that the neighbor and this router have properly configured BGP identifiers.

BGP.007

Level: U–INFO

Short Syntax: BGP.007 conn error to nbr *neighbor* clsg with noty

Long Syntax: BGP.007 connection to neighbor *neighbor* closing with notification

Description: Some error in the connection sm resulted in this message.

Cause: An error in the conection sm resulted in connection termination.

Action: Note other connection errors that occur with this event.

BGP.008

Level: U–INFO

Short Syntax: BGP.008 conn error to nbr *neighbor* clsg with no noty

Long Syntax: BGP.008 connection to neighbor *neighbor* closing with no notification

Description: Some error in the connection sm resulted in this message, usually because this speaker received a notificaton message and there is no reason to send another one back to the neighbor who sent this.

Cause: An error in the connection sm resulted in connection termination.

Action: Note other connection errors that occur along with this one.

BGP.009

Level: UI–ERROR

Short Syntax: BGP.009 foreign cls frm nbr *neighbor* sprt *sourceport* dprt *destinationport*

Long Syntax: BGP.009 foreign close from neighbor *neighbor* source port *sourceport* destination port *destinationport*

Description: The speaker just received a foreign close.

Cause: Neighbor is issuing a close.

Action: Neighbor should issue a close after a notification or during BGP id negotion. If this is the case, no action is necessary. If a connection closes for reasons other than these, the neighbor is in error.

BGP.010

Level: U–INFO

Short Syntax: BGP.010 reinit bgp conn to nbr *neighbor*

Long Syntax: BGP.010 reinitialize the bgp connection to neighbor *neighbor*

Description: If a previous connection to this neighbor resulted in termination, the speaker reinitiates the connection. This message is printed when this occurs.

Cause: Speaker is reinitializing a connection to this neighbor after an intial failure.

Action: None, unless this happens many times with no connection to the neighbor.

BGP.011

Level: U–INFO

Short Syntax: BGP.011 conn to nbr *neighbor* clsg with no noty

Long Syntax: BGP.011 connection to neighbor *neighbor* closing with no notification

Description: Probably in response to a notify received from the other end, the router is closing the BGP connection to the neighbor without sending a notify.

BGP.012

Level: UI–ERROR

Short Syntax: BGP.012 no conn listen can be done

Long Syntax: BGP.012 no connection listen can be done

Description: Something is preventing the speaker from issuing a listen.

Cause: Probably an internal error in the tcp subsystem. Also, the router could be low on memory.

Action: Check for low memory. If memory is low, check the BGP config statistics for memory utilization. A large number of neighbor connections can conceiveably use up memory.

BGP.013

Level: UI–ERROR

Short Syntax: BGP.013 open fail to bgp nbr *neighbor*

Long Syntax: BGP.013 open failure to bgp neighbor *neighbor*

Description: The bgp speaker initiates a tcp_listen request in order to receive connection requests from neighbors. This message is printed when the invocation to this function fails.

Cause: The open to the tcp subsystem failed.

Action: Serious problem. Check amount of heap memory available to router.

BGP.014

Level: U–INFO

Short Syntax: BGP.014 conn tmr frd for nbr *neighbor*

Long Syntax: BGP.014 connection timer fired for neighbor *neighbor*

Description: A connection timer is used to continue attempts to make active connections from this speaker to this neighbor. The firing of this timer causes the speaker to quit the previous tcp_open and issue another tcp_open.

Cause: The connection timer fired because no neighbor connection was completed in the specified time.

Action: None. Connection process will continue until connection to neighbor completes.

BGP.015

Level: U–INFO

Short Syntax: BGP.015 conn to nbr *neighbor* open on sprt *sourceport* dprt *destinationport*

Long Syntax: BGP.015 connection to neighbor *neighbor* open on soure port *sourceport* destination port *destinationport*

Description: An open message has been received on this connection for this neighbor.

Cause: The connection to the neighbor has completed successfully.

Action: None. This is an informational message.

BGP.016

Level: U–INFO

Short Syntax: BGP.016 Open sent to nbr *neighbor*

Long Syntax: BGP.016 Open message sent to neighbor *neighbor*

Description: When a connection is opened, the speaker sends an open message to the neighbor. This message is printed when this happens.

Cause: This is part of the connection process.

Action: None. This is an informational message.

BGP.017

Level: UI–ERROR

Short Syntax: BGP.017 msg len recd frm nbr *neighbor* sprt *sourceport* dprt *destinationport*

Long Syntax: BGP.017 message length received from neighbor *neighbor* source port *sourceport* destination port *destinationport*

Description: The message length is checked when received. This message is printed if the length of this message is smaller than the header should be.

Cause: This is probably caused by some device driver error or defect in the software either with the speaker of the neighbor.

Action: Determine if this happens with other neighbors. If yes, suspect some problem with this router. If no, probably a problem with the neighbor. This is a serious error that might require information from many sources.

BGP.018

Level: UI–ERROR

Short Syntax: BGP.018 some message to use

Long Syntax: BGP.018 some message to use

Description: None.

BGP.019

Level: UI–ERROR

Short Syntax: BGP.019 msg type fld frm nbr *neighbor* sprt *sourceport* dprt *destinationport*

Long Syntax: BGP.019 message type field from neighbor *neighbor* source port *sourceport* destination port *destinationport*

Description: BGP messages can only be of four types. This message is printed if the type field is something other than these four types.

Cause: This is probably the result of a garbled message, since message types are among the most basic pieces of bgp information.

Action: Determine if this happens with other neighbors. If yes, suspect some problem with this router. If no, this is probably a problem with this speaker. This is a serious error that requires information from many sources.

BGP.020

Level: U–INFO

Short Syntax: BGP.020 bgp init

Long Syntax: BGP.020 bgp initialization

Description: This message is printed when bgp has been enabled.

BGP.021

Level: U–INFO

Short Syntax: BGP.021 No nbr rcrd fnd for wt rule nbr *neighbor*

Long Syntax: BGP.021 No neighbor record found for this weight rule neighbor *neighbor*

Description: During initialization, no neighbor record was found for this weight rule. This can result from the removal of a neighbor record without the removal of a weight rule. Nothing is necessarily amiss.

Cause: No neighbor record for the configured weight rule.

Action: None.

BGP.022

Level: U–INFO

Short Syntax: BGP.022 No nbr rcrd fnd for ex rule nbr *neighbor*

Long Syntax: BGP.022 No neighbor record found for this external rule neighbor *neighbor*

Description: Refer to description for trap 21. This is the same message, except that there is no neighbor record for this external rule.

Cause: No neighbor record for the configured external rule.

Action: None.

BGP.023

Level: U–INFO

Short Syntax: BGP.023 Nbr *neighbor* is dbld

Long Syntax: BGP.023 Neighbor *neighbor* is disabled

Description: The neighbor record has been found, but the neighbor is disabled.

Cause: The user has disabled the neighbor.

Action: None.

BGP.024

Level: UI–ERROR

***Short Syntax*:** BGP.024 Attr len to long frm nbr *neighbor* len length

***Long Syntax*:** BGP.024 Attribute length too long from neighbor *neighbor* length *length*

Description: The length of the path attributes exceeds the length in the header.

Cause: The speaker has either garbled the message or the neighbor has.

BGP.025

Level: UI–ERROR

***Short Syntax*:** BGP.025 mand attr wo tran bit set frm nbr *neighbor* attr *attribute_type*

***Long Syntax*:** BGP.025 mandatory attribute without transitive bit set from neighbor *neighbor* attribute type *attribute_type*

Description: The neighbor has sent a mandatory attribute with the nontransitive bit set. This is a violation of the specification.

Cause: This is so basic to the protocol that the user would have to suspect some data corruption in the neighbor or the speaker.

BGP.026

Level: UI–ERROR

***Short Syntax*:** BGP.026 Mand attr with partial bit set frm nbr *neighbor* attr *attribute_type*

***Long Syntax*:** BGP.026 Mandatory attribute with partial bit set from neighbor *neighbor* attribute type *attribute_type*

Description: The neighbor has sent a mandatory attribute with the partial bit set. This is a violation of the specification.

Cause: This is so basic to the protocol that the user would have to suspect some data corruption in the neighbor or the speaker.

BGP.027

Level: UI–ERROR

Short Syntax: BGP.027 Opt non–tran attr with partial bit set from nbr *neighbor* attr *attributestring*

Long Syntax: BGP.027 Optional non–transitive attribute with partial bit set from neighbor *neighbor* attribute *attributestring*

Description: The neighbor has sent an optional attribute with the partial bit set. This is a violation of the specification.

Cause: This is a basic protocol violation and the user should suspect data corruption in the neighbor or the speaker.

BGP.028

Level: UI–ERROR

Short Syntax: BGP.028 origin path attr with bad len frm nbr *neighbor* len length

Long Syntax: BGP.028 origin path attribute has bad length from neighbor *neighbor* length *length*

Description: The origin attribute must be one byte long. This attribute has a different length.

Cause: This is a basic protocol violation and the user should suspect data corruption in the neighbor or the speaker.

BGP.029

Level: UI–ERROR

Short Syntax: BGP.029 origin path attr with bad type frm nbr *neighbor* origin *origin_type*

Long Syntax: BGP.029 origin path attribute with bad type from neighbor *neighbor* origin *origin_type*

Description: The origin attribute contains an unidentified origin type.

Cause: This is a basic protocol violation.

BGP.030

Level: UI–ERROR

Short Syntax: BGP.030 dupl AS in path attr frm nbr *neighbor* pathlen *AS_path_length*

Long Syntax: BGP.030 duplicate AS in path attribute from neighbor *neighbor* path length *AS_path_length*

Description: The neighbor has sent an ASpath attribute with a duplicate.

Cause: The AS path attribute contains a loop as evidenced by a duplicate AS. A speaker should never advertise a path with a duplicate AS.

Action: The problem is with the neighbor, since the probability of data corruption causing a duplicate is low. The neighbor is operating with a defective implementation and should be corrected, since this is a core function of BGP.

BGP.031

Level: UI–ERROR

Short Syntax: BGP.031 next hop attr with bad len frm nbr *neighbor* len length

Long Syntax: BGP.031 next hop attribute with bad length from neighbor *neighbor* length *length*

Description: The next hop should be the length of an ip address. This attribute has an incorrect length.

Cause: The neighbor has sent next hop attribute with an incorrect length. This could be the result of data corruption.

Action: If the length field is completely garbled, suspect data corruption with the speaker or the neighbor. If the length field is off by a byte, suspect a protocol violation by the *neighbor.*

BGP.032

Level: UI–ERROR

Short Syntax: BGP.032 next hop attr is bad frm nbr *neighbor* next hop *next_hop_attribute*

Long Syntax: BGP.032 next hop attribute is bad from neighbor *neighbor* next hop *next_hop_attribute*

Description: The next hop attribute is the proper length but has been determined to be incorrect.

Cause: The neighbor has sent a next hop address which is ours or a subnet address.

Action: If the address is our address, the neighbor is in definite violation of the protocol. If the address is a subnet, the neighbor is probably in violation.

BGP.033

Level: UI–ERROR

Short Syntax: BGP.033 mult exit disc attr has bad len frm nbr *neighbor* len length

Long Syntax: BGP.033 mult exit disc attribute has bad length from neighbor *neighbor* length *length*

Description: The mult exit disc attribute length is incorrect.

Cause: The neighbor has sent a mult exit disc attribute with the incorrect length.

Action: If the length is wildly off, suspect data corruption in the speaker or the neighbor. If the length is only off by a byte, the neighbor is probably in violation of the protocol.

BGP.034

Level: UI_ERROR

Short Syntax: BGP.034 local pref attr has bad len frm nbr *neighbor* len length

Long Syntax: BGP.034 local preference attribute has bad length from neighbor *neighbor* length *length*

Description: The local preference attribute length is incorrect.

Cause: The neighbor has sent a local preference with an incorrect length.

Action: If the length is wildly off, suspect data corruption in the speaker or the neighbor. If the length is only off by a byte, the neighbor is probably in violation of the protocol.

BGP.035

Level: UI–ERROR

Short Syntax: BGP.035 atom aggr attr has bad len frm nbr *neighbor* len length

Long Syntax: BGP.035 atomic aggregate attribute has bad length from neighbor *neighbor* length *length*

Description: The atomic aggregate attribute should be of length 0, but has a length different than 0.

Cause: The neighbor has sent an incorrectly formatted atomic aggregate attribute.

Action: If the length is wildly off, suspect data corruption in the speaker or the neighbor. If the length is only off by a byte, the neighbor is probably in violation of the protocol. “head3un”>BGP.036

BGP.036

Level: UI–ERROR

Short Syntax: BGP.036 aggr attr has bad len frm nbr *neighbor* len length

Long Syntax: BGP.036 aggregator attribute has bad length from neighbor *neighbor* length *length*

Description: The aggregator attribute has an incorrect length.

Cause: The neighbor has sent an aggregator attribute with the incorrect length.

Action: If the length is wildly off, suspect data corruption in the speaker or the neighbor. If the length is only off by byte, the neighbor is probably in violation of the protocol.

BGP.037

Level: UI_ERROR

Short Syntax: BGP.037 aggr attr is bad frm nbr *neighbor* attr *attributestring*

Long Syntax: BGP.037 aggregator attribute is bad from neighbor *neighbor* attribute *attributestring*

Description: The aggregator attribute has the AS of this speaker.

Cause: The neighbor has sent an aggregator attribute with the AS of this speaker.

Action: The neighbor is in violation of the protocol. The neighbor must correct this problem if this attribute is to be used.

BGP.038

Level: UI_ERROR

Short Syntax: BGP.038 unrg opt path attr frm nbr *neighbor* attr *attributestring*

Long Syntax: BGP.038 unrecognized optional path attribute from neighbor *neighbor* attribute *attributestring*

Description: This optional path attribute is unrecognized.

Cause: The neighbor has sent an optional attribute that is unrecognized.

Action: The neighbor is in violation of the protocol. The neighbor has to use optional attributes for this speaker that are recognizable. Some speakers only implement a subset of optional attributes, which is an acceptable interpretation of the specification. This speaker is fully implemented to handle optional attributes.

BGP.039

Level: UI–ERROR

Short Syntax: BGP.039 urrg well knwn attr frm nbr *neighbor* attr *attribute_type*

Long Syntax: BGP.039 unrecognized well known attribute from neighbor *neighbor* attribute *attribute_type*

Description: The well known attribute is unrecognized.

Cause: The neighbor has sent an well know attribute that is unrecognized.

Action: Since this would be a basic protocol violation, the user should suspect data corruption with the speaker or the neighbor.

BGP.040

Level: UI–ERROR

Short Syntax: BGP.040 dupl attr frm nbr *neighbor* attr *attribute_type*

Long Syntax: BGP.040 multiple attributes from neighbor *neighbor* attribute *attribute_type*

Description: Duplicate path attributes were found in the update.

Cause: The neighbor has sent an update with a duplicate path attribute.

Action: The neighbor should be checked, since this is a protocol violation.

BGP.041

Level: UI–ERROR

Short Syntax: BGP.041 missing well knwn attr frm nbr *neighbor* attr *attribute_type*

Long Syntax: BGP.041 misssing well known attribute from neighbor *neighbor* attribute *attribute_type*

Description: There is a missing well known attribute.

Cause: The neighbor has failed to send the necessary well known attributes.

Action: The neighbor should be checked, since this is a protocol violation.

BGP.042

Level: UI–ERROR

Short Syntax: BGP.042 No nlri in updt frm nbr *neighbor*

Long Syntax: BGP.042 No network layer routing information in update from neighbor *neighbor*

Description: The update message had no network layer routing information.

Cause: The neighbor sent an update message with path attributes but no routing information.

Action: The neighbor should be checked for a protocol violation.

BGP.043

Level:	U–INFO
Short Syntax:	BGP.043 nlri *nlri* rejected by ex policy frm nbr *neighbor*
Long Syntax:	BGP.043 Network layer routing information *nlri* rejected by external policy from neighbor *neighbor*
Description:	The Network layer routing information described by the path attribute has been rejected after applying policy.
Cause:	Policy configuration commands have resulted in this nlri described by the path attribute to be rejected.
Action:	None, unless this nlri should have been included.

BGP.044

Level:	U–INFO
Short Syntax:	BGP.044 new or updted rib entry *nlri* frm nbr *neighbor*
Long Syntax:	BGP.044 new or updated rib entry *nlri* from neighbor *neighbor*
Description:	A nlri has passed filters and is being put into the rib.
Cause:	The neighbor has sent an update with nlri and path that is acceptable by external policy rule definitions.
Action:	None, unless this nlri should have been excluded.

BGP.045

Level: U–INFO

***Short Syntax*:** BGP.045 unable to insert non–contig rt

***Long Syntax*:** BGP.045 unable to insert non–contiguous route

Description: The nlri from the ip forwarding table is non–contiguous.

Cause: Nothing is incorrect here. BGP is unable to handle this.

BGP.046

Level: U–INFO

***Short Syntax*:** BGP.046 Notify rcvd from *neighbor*, err *error_code*:*sub_code*

***Long Syntax*:** BGP.046 Notify received from *neighbor*, error code *error_code*, sub code *sub_code*

Description: A Notify message has been received from the neighbor. This terminates the BGP connection, and usually indicates some kind of error. The error code and sub code can be found in the BGP specification, giving the exact reason for the notification.

BGP.047

Level: U–INFO

***Short Syntax*:** BGP.047 dest *destination* frm ip fw table accepted

***Long Syntax*:** BGP.047 destination *destination* from ip forwarding table included

Description: The speaker has just included this destination as directed by internal policy.

Cause: The internal policy can specifically include destinations.

Action: None, unles the internal policy should have excluded this destination.

BGP.048

Level: UI–ERROR

***Short Syntax*:** BGP.048 bgp spkr unable to get mem

***Long Syntax*:** BGP.048 bgp speaker unable to get memory

Description: BGP was unable to allocate the necessary memory. BGP is unable to run because of this.

Cause: There is a shortage in heap memory, possibly because too many memory intensive forwarders/protocols are running.

Action: Disable unnecesary forwarders/protocols or get more memory.

BGP.049

Level: U–INFO

Short Syntax: BGP.049 closing conn to nbr *neighbor* sprt *sourceport* dprt *destinationport* conn collision

Long Syntax: BGP.049 closing connection to neighbor *neighbor* source port *sourceport* destination port *destinationport* because of connection collision

Description: BGP is removing a duplicate connection to this neighbor because of a connection collision.

Cause: Multiple tcp connections can form during the original neighbor connection establishment.

Action: None. Collisions can occur and the BGP RFC describes procedures to decide which connection wins.

BGP.050

Level: U–INFO

Short Syntax: BGP.050 Update(s) sent to nbr *neighbor*, len *message_length*

Long Syntax: BGP.050 Update sent to neighbor *neighbor*, length *message_length*

Description: One or more BGP update messages are being queued to the given neighbor. This occurs only on topology changes. The length of the entire collection of updates is displayed.

BGP.051

Level: U–INFO

Short Syntax: BGP.051 bgp state change to *state* nbr *neighbor* sprt *sourceport* dprt *destinationport*

Long Syntax: BGP.051 bgp state change to *state* neighbor *neighbor* source port *sourceport* destination port *destinationport*

Description: The state of the connection to this neighbor has just changed.

BGP.052

Level: U–INFO

*Short Syntax***:** BGP.052 Update rcvd from nbr *neighbor*, len *message_length*

*Long Syntax***:** BGP.052 Update received from neighbor *neighbor*, length *message_length*

Description: BGP update message of a given length has been received from the given neighbor. This indicates some kind of topology change.

BGP.053

Level: UI–ERROR

*Short Syntax***:** BGP.053 Del BGP route to *network*

*Long Syntax***:** BGP.053 Deleted BGP route to network *network*

Description: The BGP route to the given network is no longer valid, and has been deleted from the IP routing table.

BGP.054

Level: UI–ERROR

*Short Syntax***:** BGP.054 no more path desc idents available

*Long Syntax***:** BGP.054 no more path descriptor indentifiers available

Description: The number of path descriptor identifiers has been used up.

Cause: The number of path descriptor identifiers was used up because of the reception of a larger number of paths than expected.

Action: Allocate a larger number of path descriptor identifiers. The external policy filters can also be used to reduce the identifier demand.

BGP.055

Level: UI–ERROR

*Short Syntax***:** BGP.055 Ext Nbr *neighbor* not on common net

*Long Syntax***:** BGP.055 External Neighbor *neighbor* is not on common network

Description: External neighbors must share a common *network* with the router, else the neighbor will be ignored. The neighbor's address on the common *network* must be configured in the "BGP Config> add neighbor" command.

BGP.056

Level: U–INFO

Short Syntax: BGP.056 Open rcvd from nbr *neighbor*

Long Syntax: BGP.056 Open received from nbr *neighbor*

Description: BGP open message has been received from the given neighbor. This indicates that the neighbor wishes to initiate a conversation.

BGP.057

Level: P–TRACE

Short Syntax: BGP.057 Keepalive rcvd from nbr *neighbor*

Long Syntax: BGP.057 Keepalive received from neighbor *neighbor*

Description: BGP keepalive message has been received from the given neighbor. These are sent and received periodically in order to ensure that the BGP connection is still in tact.

BGP.058

Level: U–INFO

Short Syntax: BGP.058 Notify sent to *neighbor*

Long Syntax: BGP.058 Notify sent to *neighbor*

Description: A Notify message has been sent to the neighbor. This terminates the BGP connection, and means that we have encountered an unrecoverable error, probably the reception of bad data from the neighbor. A previously displayed ELS message indicates the exact nature of the error.

BGP.059

Level: P–TRACE

Short Syntax: BGP.059 Keepalive sent to nbr *neighbor*

Long Syntax: BGP.059 Keepalive sent to neighbor *neighbor*

Description: BGP keepalive message has been sent to the given neighbor. These are sent and received periodically in order to ensure that the BGP connection is still in tact.

BGP.060

Level: U–INFO

Short Syntax: BGP.060 Couldn't add net *network* mask *mask*

Long Syntax: BGP.060 Couldn't add network *network* mask *mask* to routing table

Description: Router unable to add a network that was received in a BGP update message to its routing table. This is either because the routing table overflowed, or because the *network* number was badly formed.

BGP.061

Level: U–INFO

Short Syntax: BGP.061 No mem for update to nbr *neighbor*

Long Syntax: BGP.061 No memory for update to neighbor *neighbor*

Description: Unable to get memory to send BGP update to peer. Router will continue to retry. If message persists, router may have run out of available memory.

BGP.062

Level: UI–ERROR

Short Syntax: BGP.062 rej Nbr *neighbor*, not in nbr tbl

Long Syntax: BGP.062 External neighbor *neighbor* is not in the neighbor table

Description: External neighbor is trying to establish a BGP connection with this speaker which does not have the neighbor in the configuration.

Action: Check the neighbor configuration in both speakers.

BGP.063

Level: U–INFO

Short Syntax: BGP.063 pasv conn exists for nbr *neighbor* new pasv conn closed

Long Syntax: BGP.063 Passive connection already exists for neighbor *neighbor.* New Passive connection is closed

Description: A passive TCP connection already exists for this neighbor, and for some reason, the neighbor tries for another passive connection (ex. Neighbor is restarted). This connection will close now.

BGP.064

Level: U–INFO

Short Syntax: BGP.064 actv conn exists for nbr *neighbor* new pasv conn closed

Long Syntax: BGP.064 active connection already exists for neighbor *neighbor.* New Passive connection is closed

Description: An active TCP connection already exists for this neighbor, and for some reason, the neighbor tries for another passive connection (ex. Neighbor is restarted). This connection will close now.

BGP.065

Level: U–INFO

Short Syntax: BGP.065 pasv conn exists for nbr *neighbor* new actv conn closed

Long Syntax: BGP.065 passive connection already exists for neighbor *neighbor.* New Active connection is closed

Description: A Passive TCP connection already exists for this neighbor, and for some reason, the neighbor tries for another active connection (ex. Neighbor is restarted). This connection will close now.

BGP.066

Level: UI–ERROR

Short Syntax: BGP.066 tcp send failed nbr *neighbor*

Long Syntax: BGP.066 tcp send failed for neighbor *neighbor*

Description: tcp_send() could not send data to the specified neighbor.

BGP.067

Level: UI–ERROR

Short Syntax: BGP.067 hold tmr exp nbr *neighbor* clsng conn

Long Syntax: BGP.067 hold timer expired for neighbor *neighbor.* Connection closing

Description: No Keep Alive message has been received from this neighbor. Thus the Keep Alive Timer expires and the connection will be closed.

BGP.068

Level: UI–ERROR

Short Syntax: BGP.068 closing conn to nbr *neighbor* sprt *sourceport* dprt *destinationport*

Long Syntax: BGP.068 abruptly closing connection to neighbor *neighbor* neighbor source port *sourceport* destination port *destinationport*

Description: The connection to this neighbor has been abruptly closed by underlying transport (TCP).

BGP.069

Level: U–INFO

Short Syntax: BGP.069 bgp state change nbr *neighbor* ev event oldst *oldstate* newst *newstate*

Long Syntax: BGP.069 bgp state change neighbor *neighbor* event event old state *oldstate* new state *newstate*

Description: The state of the connection to this neighbor has just changed.

BGP.070

Level: UI–ERROR

Short Syntax: BGP.070 Unexp event nbr *neighbor* ev *event* st *state*

Long Syntax: BGP.070 Unexpected event neighbor *neighbor* event *event* state *state*

Description: An event not handled by the BGP implementation has occurred. This indicates a software error, and should be reported to customer support.

BGP.071

Level: UE–ERROR

Short Syntax: BGP.071 Bad aggregate net *aggregate_net* mask *aggregate_mask*

Long Syntax: BGP.071 Bad aggregate net *aggregate_net* mask *aggregate_mask*

Description: An aggregate has been configured that the router cannot use. This is probably due to misconfiguration. The aggregate is ignored.

BGP.072

Level: P–TRACE

Short Syntax: BGP.072 Add NLRI *destination_net* len *destination_mask_len* updt for nbr *neighbor*

Long Syntax: BGP.072 Add NLRI *destination_net* len *destination_mask_len* update for neighbor *neighbor*

Description: A new Network Layer Reachability Information has been added to the list of NLRIs associated with a particular attribute list in the new Update message being constructed for this neighbor.

BGP.073

Level: P–TRACE

Short Syntax: BGP.073 Wdra NLRI *destination_net* len *destination_mask_len* updt for nbr *neighbor*

Long Syntax: BGP.073 Withdraw NLRI *destination_net* len *destination_mask_len* update for neighbor *neighbor*

Description: The *Network* Layer Reachability Information has been added to the list of unfeasible routes in the new Update message being constructed for this neighbor.

BOSS

BOSS SNMP Agent

This chapter describes BOSS SNMP Agent messages. The BOSS SNMP Agent provides remote management of the hub cards in the Series 90 BOSS, and also provides remote network management of the attached token ring. For information about message content and how to use the message, refer to the Introduction.

BOSS.001

Level: U_INFO

Short Syntax: BOSS.001 *msgtype msg*

Long Syntax: BOSS.001 *msgtype msg*

Description: These are messages posted by the BOSS general event logging system: BEACONING_START, BEACONING_STOP,STATION_STATE_CHANGE, RING_STATE_CHANGE, RING_CONFIG_CHANGE,NEW_ACTIVE_MONITOR, ACTIVE_MONITOR_ERROR, RING_POLL_FAIL,ISOLATING_ERRORS, NONISOLATING_ERRORS, LOBE_WIRE_FAULT,AUTO_REMOVAL_ERROR, SIGNAL_LOSS, AUTO_ERROR_ISOLATION,NEW_MODULE, MODULE_CONFIG_CHANGE, CAGE_POWER_FAIL,CAGE_POWER_OK, TEMPERATURE_BAD, and AUTOMAP_FAIL_RIRO.

BOSS.002

Level: U_INFO

Short Syntax: BOSS.002 Internal MAC transmit error

Long Syntax: BOSS.002 Internal MAC transmit error

Description: The ring manager's Token–Ring interface is unavailable.

Cause: The ring is beaconing, or the interface is disabled, or the ring manager ran out of internal buffers.

BOSS.003

Level: U_INFO

Short Syntax: BOSS.003 Automapping aborted

Long Syntax: BOSS.003 Automapping aborted

Description: The automapping algorithm has identified a software deadlock condition. It is aborting automapping because it cannot run to a successful completion.

Cause: Misconfigured agent drop, unsupported module inserted, or unstable hardware.

BOSS.004

Level: U_INFO

Short Syntax: BOSS.004 Automapping started

Long Syntax: BOSS.004 Automapping started

Description: The automapping algorithm has successfully completed active node checking, agent anchor checking, and is about to start the automapping algorithm.

BOSS.005

Level: U_INFO

Short Syntax: BOSS.005 Automapping completed

Long Syntax: BOSS.005 Automapping completed

Description: The automapping algorithm has successfully mapped all active logical nodes to physical ports.

BOSS.006

Level: U_INFO

Short Syntax: BOSS.006 Autoreconfiguration aborted

Long Syntax: BOSS.006 Autoreconfiguration aborted

Description: Autoreconfiguration cannot be completed because a hard error was detected.

Cause: Misconfigured agent drop, unsupported module inserted, or unstable hardware.

Cause: Fatal ‘‘BOSS_MEMORYERR’’

Short Syntax: BOSS: Unexpected out of memory

Description: This message indicates abnormal operation of the memory al–location system.

Description: Fatal ‘‘BOSS_INTERNALERR’’

Short Syntax: BOSS: Internal error

Description: This message indicates abnormal operation of the BOSS SNMP Agent.

BR

Bridge Routing

This chapter describes Bridge Routing messages. These are generated by the software that makes the decision whether to bridge, route, or drop received packets based on their protocol type and MAC addresses. For information about message content and how to use the message, refer to the Introduction.

BR.001

Level: C–INFO

Short Syntax: BR.001 *source_mac–>dest_mac* drp, port block/list, nt *network*

Long Syntax: BR.001 Frame from *source_mac* to *dest_mac* dropped, received on blocked or listening port, network *network*

Description: A MAC frame has been received by the hardware, but is being dropped because the port it was received on is in the”blocking” or “listening” state. Frames are only processed when the port is in the “learning” or “forwarding” state.

Cause: Normal on port bring up.

Action: Wait for port to transition to “learning” and “forwarding”states.

BR.002

Level: P–TRACE

Short Syntax: BR.002 *source_mac–>dest_mac* drp, dst same LAN, nt *network*

Long Syntax: BR.002 Frame from *source_mac* to *dest_mac* dropped, destination on same LAN, network *network*

Description: A MAC frame has been received whose destination address is known to be on the same side of the bridge as the packet came from. It is dropped by the filtering logic since it does not need to be bridged. Note that this event is not counted by ELS for performance reasons. A counter is kept in ASRT, it is the ‘‘Dropped, dest addr filtering’’ entry in the ASRT>LIST SOURCE–ROUTE COUNTERS and ASRT>LIST TRANSPARENT COUNTERS commands.

Cause: Normal local traffic on network.

BR.003

Level: UE–ERROR

Short Syntax: BR.003 *source_mac–>dest_mac*, brdg encap for rout prot IPX(802.3), drp, nt *network*

Long Syntax: BR.003 Frame from *source_mac* to *dest_mac*, WAN bridge encapsulation for routed protocol IPX (802.3), dropped, network *network*

Description: A frame has been received over a WAN interface in the Ether–net bridge encapsulation, but the IPX protocol (in 802.3 encapsulation) is routed by this node.

Cause: Configuration error at local or remote node with respect to bridging or routing particular protocols over the WAN link.

Action: All hosts on a WAN link must agree on whether to bridge or route a given protocol. Reconfigure as appropriate.

BR.004

Level: P–TRACE

Short Syntax: BR.004 *source_mac–>dest_mac* prt IPX (802.3) filt, drp, nt *network*

Long Syntax: BR.004 Frame from *source_mac* to *dest_mac*, protocol IPX(802.3) filtered, dropped, network *network*

Description: A frame has been received for the IPX protocol (in 802.3 encapsulation), but the IPX protocol is being administratively filtered by the bridge. The frame will be dropped.

Cause: Frame received for filtered protocol.

BR.005

Level: P–TRACE

Short Syntax: BR.005 *source_mac–>dest_mac* SNAP *protocol_identifier* filt, drp, nt *network*

Long Syntax: BR.005 Frame from *source_mac* to *dest_mac*, IEEE 802 SNAP Protocol Identifier *protocol_identifier* filtered, dropped, network *network*

Description: A frame has been received for the specified IEEE 802 Subnetwork Access Protocol (SNAP) Protocol Identifier (PID), but this PID is being administratively filtered by the bridge. The frame will be dropped.

Cause: Frame received for filtered protocol.

BR.006

Level: U–TRACE

Short Syntax: BR.006 Unreg dst *source_mac–>dest_mac* SNAP *protocol_identifier*, drp, nt *network*

Long Syntax: BR.006 Frame from *source_mac* to unregistered destination MAC address *dest_mac*, IEEE 802 SNAP Protocol Identifier *protocol_identifier*, dropped, network *network*

Description: A frame has been received for the IEEE 802 Subnetwork Access Protocol (SNAP) Protocol Identifier (PID) which corresponds with an enabled protocol, but the destination MAC address is not registered in the bridge. The frame will be dropped.

Cause: If *dest_mac* is a unicast address, a station on the LAN is sending frames for this protocol to the wrong next hop MAC address.

Action: Correct action of remote station.

Cause: If *dest_mac* is a multicast address, a station on the LAN maybe sending frames to the wrong multicast address, or perhaps just to one that this router does not have enabled. Depend–ing on the protocol, this may or may not be an error.

Action: Correct action of remote station, if necessary.

BR.007

Level: P–TRACE

Short Syntax: BR.007 *source_mac–>dest_mac* SNAP *protocol_identifier*, end–node, nt *network*

Long Syntax: BR.007 Frame from *source_mac* to *dest_mac*, IEEE 802 SNAP Protocol Identifier *protocol_identifier* for endnode protocol,network *network*

Description: A multicast frame has been received for the IEEE 802 Subnetwork Access Protocol (SNAP) Protocol Identifier (PID) which corresponds with an endnode protocol. The frame will be both bridged and locally processed by the endnode protocol.

BR.008

Level: UE–ERROR

Short Syntax: BR.008 *source_mac–>dest_mac*, brdg encap for rout SNAP *protocol_identifier*, drp, nt *network*

Long Syntax: BR.008 Frame from *source_mac* to *dest_mac*, WAN bridge encapsulation for routed IEEE 802 SNAP Protocol Identifier *protocol_identifier*, dropped, network *network*

Description: An IEEE 802.2 frame has been received over a WAN interface in a bridge encapsulation, but its IEEE 802 Subnetwork Access Protocol (SNAP) Protocol Identifier (PID) is one that is being routed by this node. The frame will be dropped.

Cause: Configuration error at local or remote node with respect to bridging or routing particular protocols over the WAN link.

Action: All hosts on a WAN link must agree on whether to bridge or route a given protocol. Reconfigure as appropriate.

BR.009

Level: UE–ERROR

Short Syntax: BR.009 BDPU *source_mac–>dest_mac*, wrng dst, drp, nt *network*

Long Syntax: BR.009 IEEE 802.1D BPDU *source_mac* to *dest_mac*, wrong destination, dropped, network *network*

Description: An IEEE 802.1D Bridge Protocol Data Unit (BPDU) was received at the wrong destination address. It is supposed to be ad–dressed to a particular multicast address. The BPDU will be dropped.

Cause: Programming error at remote node.

Action: Correct software in remote node.

Cause: Node speaking IBM Token–Ring proprietary source–routing spanning tree protocol, which uses a non–standard destination address for BPDUs.

Action: Ignore message, or reconfigure source–routing node.

BR.010

Level: P–TRACE

Short Syntax: BR.010 *source_mac–>dest_mac* DSAP *destination_service_access_point* filt, drp, nt *network*

Long Syntax: BR.010 Frame from *source_mac* to *dest_mac*, IEEE 802.2 DSAP *destination_service_access_point* filtered, dropped, network *network*

Description: A frame has been received for the specified IEEE 802.2 Destination Service Access Point (DSAP), but this DSAP is being administratively filtered by the bridge. The frame will be dropped.

Cause: Frame received for filtered protocol.

BR.011

Level: U–TRACE

Short Syntax: BR.011 Unreg dst *source_mac–>dest_mac* DSAP *destination_service_access_point,* drp, nt *network*

Long Syntax: BR.011 Frame from *source_mac* to unregistered destination MAC address *dest_mac*, IEEE 802.2 DSAP *destination_service_access_point,* dropped, network *network*

Description: A frame has been received for the IEEE 802.2 Destination Service Access Point (DSAP) which corresponds with an enabled protocol, but the destination MAC address is not registered in the bridge. The frame will be dropped.

Cause: If *dest_mac* is a unicast address, a station on the LAN is sending frames for this protocol to the wrong next hop MAC address.

Action: Correct action of remote station.

Cause: If *dest_mac* is a multicast address, a station on the LAN maybe sending frames to the wrong multicast address, or perhaps just to one that this router does not have enabled. Depend–ing on the protocol, this may or may not be an error.

Action: Correct action of remote station, if necessary.

BR.012

Level: P–TRACE

Short Syntax: BR.012 *source_mac–>dest_mac* DSAP *destination_service_access_point,* endnode, nt *network*

Long Syntax: BR.012 Frame from *source_mac* to *dest_mac*, IEEE 802.2 DSAP *destination_service_access_point* for endnode protocol, network *network*

Description: A multicast frame has been received for the IEEE 802.2 Destination Service Access Point (DSAP) which corresponds with an endnode protocol. The frame will be both bridged and locally processed by the endnode protocol.

BR.013

Level: UE–ERROR

Short Syntax: BR.013 *source_mac–>dest_mac,* brdg encap for rout DSAP *destination_service_access_point,* drp, nt *network*

Long Syntax: BR.013 Frame from *source_mac* to *dest_mac*, WAN bridge encapsulation for routed IEEE 802.2 DSAP *destination_service_access_point,* dropped, network *network*

Description: An IEEE 802.2 frame has been received over a WAN interface in a bridge encapsulation, but its IEEE 802.2 Destination Service Access Point (DSAP) is one that is being routed by this node. The frame will be dropped.

Cause: Configuration error at local or remote node with respect to bridging or routing particular protocols over the WAN link.

Action: All hosts on a WAN link must agree on whether to bridge or route a given protocol. Reconfigure as appropriate.

BR.014

Level: P–TRACE

Short Syntax: BR.014 *source_mac–>dest_mac* Etype *Ethernet_type* filt, drp,nt *network*

Long Syntax: BR.014 Frame from *source_mac* to *dest_mac*, Ethernet type *Ethernet_type* filtered, dropped, network *network*

Description: A frame has been received for the specified Ethernet type,but this type is being administratively filtered by the bridge. The frame will be dropped.

Cause: Frame received for filtered protocol.

BR.015

Level: U–TRACE

Short Syntax: BR.015 Unreg dst *source_mac–>dest_mac* Etype *Ethernet_type*,drp, nt *network*

Long Syntax: BR.015 Frame from *source_mac* to unregistered destination MAC address *dest_mac*, Ethernet type *Ethernet_type*, dropped, network *network*

Description: A frame has been received for the Ethernet type which corresponds with an enabled protocol, but the destination MAC address is not registered in the bridge. The frame will be dropped.

Cause: If *dest_mac* is a unicast address, a station on the LAN is sending frames for this protocol to the wrong next hop MAC address.

Action: Correct action of remote station.

Cause: If *dest_mac* is a multicast address, a station on the LAN maybe sending frames to the wrong multicast address, or perhaps just to one that this router does not have enabled. Depend–ing on the protocol, this may or may not be an error.

Action: Correct action of remote station, if necessary.

BR.016

Level: P–TRACE

Short Syntax: BR.016 *source_mac–>dest_mac* Etype *Ethernet_type*, endnode, nt *network*

Long Syntax: BR.016 Frame from *source_mac* to *dest_mac*, Ethernet type *Ethernet_type* for endnode protocol, network *network*

Description: A multicast frame has been received for the Ethernet type which corresponds with an endnode protocol. The frame will be both bridged and locally processed by the endnode protocol.

BR.017

Level: UE–ERROR

Short Syntax: BR.017 *source_mac–>dest_mac*, brdg encap for rout Etype *Ethernet_type*, drp, nt *network*

Long Syntax: BR.017 Frame from *source_mac* to *dest_mac*, WAN bridge encapsulation for routed Ethernet type *Ethernet_type*, dropped,network *network*

Description: An Ethernet frame has been received over a WAN interface in the Ethernet bridge encapsulation, but its Ethernet type is one that is being routed by this node. The frame will be dropped.

Cause: Configuration error at local or remote node with respect to bridging or routing particular protocols over the WAN link.

Action: All hosts on a WAN link must agree on whether to bridge or route a given protocol. Reconfigure as appropriate.

BR.018

Level: P–TRACE

Short Syntax: BR.018 SR *source_mac–>dest_mac* DSAP *destination_service_access_point* filt, drp, nt *network*

Long Syntax: BR.018 Source–routed frame from *source_mac* to *dest_mac*, IEEE802.2 DSAP *destination_service_access_point* filtered,dropped, network *network*

Description: A source–routed frame has been received for the specified IEEE 802.2 Destination Service Access Point (DSAP), but this DSAP is being administratively filtered by the bridge. The frame will be dropped.

Cause: Frame received for filtered protocol.

BR.019

Level: U–TRACE

Short Syntax: BR.019 SR unreg dst *source_mac–>dest_mac* DSAP *destination_service_access_point*, drp, nt *network*

Long Syntax: BR.019 Source–routed frame from *source_mac* to unregistered destination MAC address *dest_mac*, IEEE 802.2 DSAP *destination_service_access_point*, dropped, network *network*

Description: A source–routed frame has been received for the IEEE 802.2Destination Service Access Point (DSAP) which corresponds with an enabled protocol, but the destination MAC address is not registered in the bridge. The frame will be dropped.

Cause: If *dest_mac* is a unicast address, a station on the LAN is sending frames for this protocol to the wrong next hop MAC address.

Action: Correct action of remote station.

Cause: If *dest_mac* is a multicast address, a station on the LAN maybe sending frames to the wrong multicast address, or perhaps just to one that this router does not have enabled. Depend–ing on the protocol, this may or may not be an error.

Action: Correct action of remote station, if necessary.

BR.020

Level: P–TRACE

Short Syntax: BR.020 SR *source_mac–>dest_mac* DSAP *destination_service_access_point*, endnode, nt *network*

Long Syntax: BR.020 Source–routed frame from *source_mac* to *dest_mac*, IEEE802.2 DSAP *destination_service_access_point* for endnode protocol, network *network*

Description: A multicast source–routed frame has been received for the IEEE 802.2 Destination Service Access Point (DSAP) which corresponds with an endnode protocol. The frame will be both bridged and locally processed by the endnode protocol.

BR.021

Level: P–TRACE

Short Syntax: BR.021 SR *source_mac–>dest_mac* SNAP *protocol_identifier* filt, drp, nt *network*

Long Syntax: BR.021 Source-routed frame from *source_mac* to *dest_mac*, IEEE802 SNAP Protocol Identifier *protocol_identifier* filtered,dropped, network *network*

Description: A source–routed frame has been received for the specified IEEE 802 Subnetwork Access Protocol (SNAP) Protocol Identifier (PID), but this PID is being administratively filtered by the bridge. The frame will be dropped.

Cause: Frame received for filtered protocol.

BR.022

Level: U–TRACE

Short Syntax: BR.022 SR unreg dst *source_mac–>dest_mac* SNAP *protocol_identifier*, drp, nt *network*

Long Syntax: BR.022 Source–routed frame from *source_mac* to unregistered destination MAC address *dest_mac*, IEEE 802 SNAP Protocol Identifier *protocol_identifier*, dropped, network *network*

Description: A source–routed frame has been received for the IEEE 802Subnetwork Access Protocol (SNAP) Protocol Identifier (PID)which corresponds with an enabled protocol, but the destination MAC address is not registered in the bridge. The frame will be dropped.

Cause: If *dest_mac* is a unicast address, a station on the LAN is sending frames for this protocol to the wrong next hop MAC address.

Action: Correct action of remote station.

Cause: If *dest_mac* is a multicast address, a station on the LAN maybe sending frames to the wrong multicast address, or perhaps just to one that this router does not have enabled. Depend–ing on the protocol, this may or may not be an error.

Action: Correct action of remote station, if necessary.

BR.023

Level: P–TRACE

Short Syntax: BR.023 SR *source_mac–>dest_mac* SNAP *protocol_identifier*,endnode, nt *network*

Long Syntax: BR.023 Source–routed frame from *source_mac* to *dest_mac*, IEEE802 SNAP Protocol Identifier *protocol_identifier* for endnode protocol, network *network*

Description: A source–routed multicast frame has been received for the IEEE 802 Subnetwork Access Protocol (SNAP) Protocol Identifier (PID) which corresponds with an endnode protocol. The frame will be both bridged and locally processed by the end–node protocol.

BR.024

Level: P–TRACE

Short Syntax: BR.024 *source_mac–>dest_mac* drp, dst add flt, nt *network*

Long Syntax: BR.024 Frame from *source_mac* to *dest_mac* dropped, destination address filtered, network *network*

Description: A MAC frame has been received by the hardware, but is being dropped because the destination MAC address is being administratively filtered by the bridge. The frame will be dropped.

Cause: Receipt of frame whose destination MAC address matches the destination filter.

BR.025

Level: P–TRACE

Short Syntax: BR.025 *source_mac–>dest_mac* drp, src add flt, nt *network*

Long Syntax: BR.025 Frame from *source_mac* to *dest_mac* dropped, source address filtered, network *network*

Description: A MAC frame has been received by the hardware, but is being dropped because the source MAC address is being administratively filtered by the bridge. The frame will be dropped.

Cause: Receipt of frame whose source MAC address matches the destination filter.

BR.026

Level: P–TRACE

*Short Syntax***:** BR.026 SR *source_mac–>dest_mac* drp, dst add flt, nt *network*

*Long Syntax***:** BR.026 Frame from *source_mac* to *dest_mac* dropped, destination address filtered, network *network*

Description: A source–routed MAC frame has been received by the hardware,but is being dropped because the destination MAC address is being administratively filtered by the bridge. The frame will be dropped.

Cause: Receipt of frame whose destination MAC address matches the destination filter.

BR.027

Level: P–TRACE

*Short Syntax***:** BR.027 SR *source_mac–>dest_mac* drp, src add flt, nt *network*

*Long Syntax***:** BR.027 Frame from *source_mac* to *dest_mac* dropped, source address filtered, network *network*

Description: A source–routed MAC frame has been received by the hardware,but is being dropped because the source MAC address is being administratively filtered by the bridge. The frame will be dropped.

Cause: Receipt of frame whose source MAC address matches the destination filter.

BR.028

Level: UI–ERROR

*Short Syntax***:** BR.028 No buf for endnode bridge, *source_mac–>dest_mac*, nt *network*, not bridged

*Long Syntax***:** BR.028 No buffer to copy packet for endnode bridge and process, from *source_mac* to *dest_mac*, network *network*, not bridged

Description: A Multicast frame has been received for an endnode protocol that is both bridged and locally processed. There was no buffer to make two copies of the frame for both types of processing, so it will not be bridged, only locally processed.

Cause: Severe packet buffer shortage.

Action: Check memory statistics in GWCON to verify packet buffer level. If possible, make routing or bridging tables smaller. If tables cannot be made smaller, increase memory size.

Cause: Traffic peak using all available buffers.

Action: This is the problem if this message occurs very infrequently.

BR.029

Level: C–TRACE

Short Syntax: BR.029 NB inp pkt fltd *source_mac–>dest_mac*, prt *port*, nt *network*

Long Syntax: BR.029 NETBIOS Input Packet Filtered – *source_mac–>dest_mac*,port *port*, network *network*

Description: A NETBIOS packet has matched the criteria specified in a NETBIOS Filter configuration record. The packet is dropped.

BTP

BOOTP

This chapter describes messages for the bootstrap protocol (BOOTP). BOOTP is part of the IP protocol. For information about message content and how to use the message, refer to the Introduction.

BTP.001

Level: U–TRACE

Short Syntax: BTP.001 rcvd rqst frm (*client_IP_address*, nt *network ID*)

Long Syntax: BTP.001 received request from (*client_IP_address*, net *network ID*)

Description: A BOOTP request has been received on a particular interface. The client IP address is included in the message, but may be unknown at this time, in which case it will show up as0.0.0.0.

BTP.002

Level: UE–ERROR

Short Syntax: BTP.002 bd rqst frm (*client_IP_address*, nt *network ID*): *reason*

Long Syntax: BTP.002 bad request from (*client_IP_address*, net *network ID*): *reason*

Description: A BOOTP request has been received on a particular interface. The client IP address is included in the message, but may be unknown at this time, in which case it will show up as 0.0.0.0. The request is bad for the stated reason, and is therefore discarded.

BTP.003

Level: U–TRACE

Short Syntax: BTP.003 fwd rqst to *server_IP_address*

Long Syntax: BTP.003 Forwarding request to *server_IP_address*

Description: A BOOTP request is being forwarded to a particular server.

BTP.004

Level: U–TRACE

Short Syntax: BTP.004 fwd rply *server_IP_address –> client_IP_address*

Long Syntax: BTP.004 Forwarding reply from *server_IP_address* to *client_IP_address*

Description: A BOOTP reply is being forwarded from a particular server back to the client, using the router as a relay agent.

BTP.005

Level: UE–ERROR

Short Syntax: BTP.005 bad reply *server_IP_address –> client_IP_address*: *reason*

Long Syntax: BTP.005 bad reply from *server_IP_address* to *client_IP_address*: *reason*

Description: A BOOTP reply was received. We are unable to relay the reply to the client due to the stated error. The BOOTP reply has been discarded.

Description: Panic ‘‘btpudperr’’

Short Syntax: bootp udp port not avail

Description: Another application registered previously with bootp’s UDP port.

Action: Contact customer service.

DLS

Data-link Switching

This chapter describes data-link Switching (DLSw) Protocol messages. For information about message content and how to use the message, refer to the Introduction.

DLS.001

Level: UI–ERROR

Short Syntax: DLS.001 nghbr TCP cnfgrtn could not be found for dst *ip_address*

Long Syntax: DLS.001 neighbor TCP configuration entry could not be found for this destination ip address *ip_address*

Description: As a result of DLS requesting an OPEN to a particular destination specified by the ip address, TCPIM tries to find the corresponding tcp configuration entry in the RAM list. If not found, it prints this message.

DLS.002

Level: C–INFO

Short Syntax: DLS.002 opening new trnsprt cnn to nghbr at *ip_address*

Long Syntax: DLS.002 opening a new transport connection to the neighbor at *ip_address*

Description: As a result of DLS requesting an OPEN to a particular destination specified by the ip address, TCPIM opens a connection to the destination.

DLS.003

Level: UI–ERROR

Short Syntax: DLS.003 illgl entry thrgh rd prt from *ip_address*

Long Syntax: DLS.003 illegal entry through read port from ip address *ip_address*

Description: Our read tcp connection has been opened through an illegal neighbor (that is, someone we do not know about). As a result, we reject the connection, and thereby close it.

DLS.004

Level: C–INFO

Short Syntax: DLS.004 tnsprt cnn alrdy in pnd/estblsh state *ip_address*

Long Syntax: DLS.004 transport connection already in a pending/established state for ip address *ip_address*

Description: DLS has requested we open a connection to the destination ip address. However, a connection already exists (or is in the pending state).

DLS.005

Level: C–INFO

Short Syntax: DLS.005 opening new trnsprt cnn to nghbr at *ip_address*

Long Syntax: DLS.005 opening a new transport connection to the neighbor at *ip_address*

Description: As a result of DLS requesting an OPEN to a particular destination specified by the ip address, TCPIM opens a connection to the destination.

DLS.006

Level: UI–ERROR

Short Syntax: DLS.006 cannot close cnn – no estblshd nghbr at *ip_address*

Long Syntax: DLS.006 cannot close the transport connection – no established neighbor at ip address *ip_address*

Description: DLS is requesting a transport connection to be closed – however, it cannot be closed because there is no established connection to that neighbor.

DLS.007

Level: C–INFO

Short Syntax: DLS.007 read prt opened, now opening write prt to nghbr at *ip_address*

Long Syntax: DLS.007 read port opened, now opening write port to neighbor at *ip_address*

Description: The other side has opened our read port, and as a result, we open our write port to the other side.

DLS.008

Level: UE–ERROR

Short Syntax: DLS.008 DLSw disabled no SRB seg defined config

Long Syntax: DLS.008 DLSw forwarder disabled no SRB segment defined

Description: The Data Link Switching forwarder has been disabled because of improper configuration. This was no SRB segment number defined, though there was LLC–2 saps defined.

DLS.009

Deleted: Message deleted.

DLS.010

Deleted: Message deleted.

DLS.011

Deleted: Message deleted.

DLS.012

Deleted: Message deleted.

DLS.013

Level: UE–ERROR
*Short Syntax***:** DLS.013 can't register with UDP on DLS group port
*Long Syntax***:** DLS.013 can't register with UDP on DLS group port
Description: Registration with UDP on DLS group port failed.

DLS.014

Level: UE–ERROR
*Short Syntax***:** DLS.014 no mem to join group
*Long Syntax***:** DLS.014 no memory to join group
Description: There was not enough free memory allocated to the data structures necessary to join a group.

DLS.015

Level: UE–ERROR
*Short Syntax***:** DLS.015 no iorb to send group packet
*Long Syntax***:** DLS.015 no iorb to send group packet
Description: There was no iorb buffer available to send a group join or join response.

DLS.016

Level: P–TRACE
*Short Syntax***:** DLS.016 Sent group pkt type *type* group *group* role *role* dest *destination*
*Long Syntax***:** DLS.016 Sent group packet type *type* group *group* role *role* dest *destination*
Description: A DLSw group packet was sent.

DLS.017

Deleted: Message deleted.

DLS.018

Level: P–TRACE

Short Syntax: DLS.018 Rcvd group pkt type *type* group *group* role *role* src *source*

Long Syntax: DLS.018 Received group packet type *type* group *group* role *role* source *source*

Description: A DLSw group packet was received.

DLS.019

Level: UE–ERROR

Short Syntax: DLS.019 Rcvd bad group pkt vers *version* type *type* priority *priority* domain *domain*

Long Syntax: DLS.019 Received bad group packet version *version* type *type* priority *priority* domain *domain*

Description: A DLSw group packet was received with either a bad version#, type, priority, or domain id.

DLS.020

Level: C–INFO

Short Syntax: DLS.020 Rcvd group pkt for deleted group *group*

Long Syntax: DLS.020 Received group packet for deleted group *group*

Description: A group packet was received for a group that was dynamically deleted.

DLS.021

Level: C–INFO

Short Syntax: DLS.021 Rcvd group pkt mismatched roles group *group* role *role*

Long Syntax: DLS.021 Received group packet but mismatched roles group *group* role *role*

Description: A group packet was received but the roles were mismatched. The only valid role matches are Client/Server and Peer/Peer.

DLS.022

Level: C–INFO

Short Syntax: DLS.022 Connecting to *address* grp *group*

Long Syntax: DLS.022 Connecting to IP Address *address* group *group*

Description: A group match has been found and we are initiating a connect.

DLS.023

Level: C–INFO

Short Syntax: DLS.023 Can't connect to *address* from grp *group*, *IPaddr* deleted

Long Syntax: DLS.023 Can't connect to *address* from group *group*, IP address dynamically deleted

Description: A group match has been found but we are not initiating a connection because a static IP connection to the partner has been deleted.

DLS.024

Deleted: Message deleted.

DLS.025

Level: UE–ERROR

Short Syntax: DLS.025 No mem to queue group packet to tasker

Long Syntax: DLS.025 No memory to queue group packet to tasker

Description: There was not enough memory to get a queue header to add a task to send a group packet.

DLS.026

Level: UE–ERROR

Short Syntax: DLS.026 group packet not sent, tasker queue full

Long Syntax: DLS.026 group packet not sent, tasker queue full

Description: A group packet could not be sent because the tasker queue was full.

DLS.027

Level: UE–ERROR

Short Syntax: DLS.027 max number of sdlc link sta exceeded sta *station* on int *interface* not opened

Long Syntax: DLS.027 maximum number of sdlc link stations exceeded sta *station* on int *interface* not opened

Description: The maximum number of sdlc link stations has been exceeded since all available source SAPs have been allocated. The link station was not opened.

DLS.028

Level: UE–ERROR

Short Syntax: DLS.028 no mem to init SDLC link nt *network ID*

Long Syntax: DLS.028 no memory to initialize SDLC link net *network ID*

Description: There was not enough memory available to initialize an SDLC link.

DLS.029

Level: UI–ERROR

***Short Syntax*:** DLS.029 unexp rtn code from sdlc open station = *rtn_code* nt *network ID*

***Long Syntax*:** DLS.029 unexpected return code from sdlc open station =*rtn_code* net *network ID*

Description: The sdlc open station function returned an unexpected return code.

DLS.030

Level: UI–ERROR

***Short Syntax*:** DLS.030 sdlc lnk ctl blk not fnd during del lnk nt *network ID*

***Long Syntax*:** DLS.030 sdlc link control block not found during delete link net *network ID*

Description: The sdlc link control block was not found for the SDLC link being deleted.

DLS.031

Level: C–INFO

***Short Syntax*:** DLS.031 sdlc station closed nt *network ID*

***Long Syntax*:** DLS.031 sdlc station closed net *network ID*

Description: The sdlc station for the network interface has been successfully closed.

DLS.032

Level: UI–ERROR

***Short Syntax*:** DLS.032 unexp rtn code from sdlc cls station = *rtn_code* nt *network ID*

***Long Syntax*:** DLS.032 unexpected return code from sdlc close station =*rtn_code* net *network ID*

Description: The sdlc close station function returned an unexpected return code.

DLS.033

Level: UI–ERROR

***Short Syntax*:** DLS.033 sdlc lnk ctl blk not fnd during init lnk sta nt *network ID*

***Long Syntax*:** DLS.033 sdlc link control block not found during init link station net *network ID*

Description: The sdlc link control block was not found for the SDLC link station being initialized.

DLS.034

Level: UE–ERROR

***Short Syntax*:** DLS.034 no mem to init SDLC link sta nt *network ID*

***Long Syntax*:** DLS.034 no memory to initialize SDLC link station net *network ID*

Description: There was not enough memory available to initialize an SDLC link station.

DLS.035

Level: C–INFO

***Short Syntax*:** DLS.035 sdlc link sta open addr *link_address* nt *network ID*

***Long Syntax*:** DLS.035 sdlc link station opened address *link_address* net *network ID*

Description: The sdlc link station for the link address has been successfully opened on the network interface.

DLS.036

Level: UI–ERROR

***Short Syntax*:** DLS.036 dupl sdlc link sta addr *link_address* nt *network ID*

***Long Syntax*:** DLS.036 duplicate sdlc link station *address link_address* net *network ID*

Description: The specified sdlc link station could not be opened bceause it is a duplicate of one already opened.

DLS.037

Level: UI–ERROR

***Short Syntax*:** DLS.037 unexp rtn code from sdlc open lnk sta = *rtn_code* nt *network ID*

***Long Syntax*:** DLS.037 unexpected return code from sdlc open link station =*rtn_code* net *network ID*

Description: The sdlc open link station function returned an unexpected return code.

DLS.038

Level: C–INFO

***Short Syntax*:** DLS.038 sdlc link station closed addr *link_address* nt *network ID*

***Long Syntax*:** DLS.038 sdlc station closed address *link_address* net *network ID*

Description: The sdlc link station for the address and network interface specified has been successfully closed.

DLS.039

Level: C–INFO

Short Syntax: DLS.039 processing sdlc net up for addr *link_address* nt *network ID*

Long Syntax: DLS.039 processing sdlc net up for address *link_address* net *network ID*

Description: A net up indication has been received for an sdlc link station.

DLS.040

Level: C–INFO

Short Syntax: DLS.040 processing sdlc net down for addr *link_address* nt *network ID*

Long Syntax: DLS.040 processing sdlc net down for address *link_address* net *network ID*

Description: A net down indication has been received for an sdlc link station.

DLS.041

Level: C–INFO

Short Syntax: DLS.041 rcvd sdlc net up while not in down st for addr *link_address* nt *network ID*

Long Syntax: DLS.041 received sdlc net up while not in down state for address *link_address* net *network ID*

Description: A net up indication has been received for an sdlc link station on an interface that is not down.

DLS.042

Level: C–INFO

Short Syntax: DLS.042 sdlc trans to resolve pend st for addr *link_address* nt *network ID*

Long Syntax: DLS.042 sdlc transition to resolve pending state for address *link_address* net *network ID*

Description: The sdlc link station specified is transitioning to resolve pending state.

DLS.043

Level: UI–ERROR

Short Syntax: DLS.043 unexp sdlc test rsp for addr *link_address* nt *network ID*

Long Syntax: DLS.043 unexpected sdlc test response for address *link_address* net *network ID*

Description: An unexpected test response was received from the sdlc link station specified.

DLS.044

Level: UE–ERROR

Short Syntax: DLS.044 unexp sdlc non xid0 from pu 2 dev for addr *link_address* nt *network ID*

Long Syntax: DLS.044 unexpected sdlc non xid0 from pu 2 device for address *link_address* net *network ID*

Description: An unexpected XID type was received for a PU type 2 device from the sdlc link station specified.

DLS.045

Level: UE–ERROR

Short Syntax: DLS.045 invalid sdlc xid0 len from addr *link_address* nt *network ID*

Long Syntax: DLS.045 invalid sdlc xid0 length from address *link_address* net *network ID*

Description: An invalid length XID–0 was received from the sdlc link station specified.

DLS.046

Level: UE–ERROR

Short Syntax: DLS.046 conn ind rcvd from sec sdlc station from addr *link_address* nt *network ID*

Long Syntax: DLS.046 connection indication received from secondary sdlc station from address *link_address* net *network ID*

Description: An indication that a secondary SDLC link station sent a SNRM was received from the sdlc link station specified.

DLS.047

Level: C–INFO

Short Syntax: DLS.047 sdlc trans to contacted st for addr *link_address* nt *network ID*

Long Syntax: DLS.047 sdlc transition to contacted state for address *link_address* net *network ID*

Description: The sdlc link station specified is transitioning to contacted state.

DLS.048

Level: UE–ERROR

Short Syntax: DLS.048 unexp sdlc conn cfm for addr *link_address* nt *network ID*

Long Syntax: DLS.048 unexpected sdlc connect confirm for address *link_address* net *network ID*

Description: An unexpected connect confirm was received from the sdlc link station specified.

DLS.049

Level: C–INFO

Short Syntax: DLS.049 sdlc disc ind rcvd addr *link_address* nt *network ID*

Long Syntax: DLS.049 sdlc disconnect indication received address *link_address* net *network ID*

Description: A disconnect indication was received from the SDLC link station specified.

DLS.050

Level: C–INFO

Short Syntax: DLS.050 sdlc disc compl addr *link_address* nt *network ID*

Long Syntax: DLS.050 sdlc disconnect complete address *link_address* net *network ID*

Description: The disconnect sequence for the SDLC link station specified is complete.

DLS.051

Level: UE–ERROR

Short Syntax: DLS.051 unexp sdlc disc ind in st *state* for addr *link_address* nt *network ID*

Long Syntax: DLS.051 unexpected sdlc disconnect indication in state *state* for address *link_address* net *network ID*

Description: An unexpected disconnect indication was received from the sdlc link station specified.

DLS.052

Level: C–INFO

Short Syntax: DLS.052 sdlc disc conf addr *link_address* nt *network ID*

Long Syntax: DLS.052 sdlc disconnect cofirm address *link_address* net *network ID*

Description: A disconnect confirm was received for the SDLC link station specified.

DLS.053

Level: UE–ERROR

Short Syntax: DLS.053 unexp sdlc disc cfm in state *state* for addr *link_address* nt *network ID*

Long Syntax: DLS.053 unexpected sdlc disconnect confirm in state *state* for address *link_address* net *network ID*

Description: An unexpected disconnect confirm was received from the sdlc link station specified.

DLS.054

Level: UE–ERROR

Short Syntax: DLS.054 unexp sdlc resolve_r in state *state* for addr *link_address* nt *network ID*

Long Syntax: DLS.054 unexpected sdlc resolve_r in state *state* for address *link_address* net *network ID*

Description: An unexpected resolve_r event was received for the sdlc link station specified.

DLS.055

Level: C–INFO

Short Syntax: DLS.055 sdlc trans to connected st for addr *link_address* nt *network ID*

Long Syntax: DLS.055 sdlc transition to connected state for address *link_address* net *network ID*

Description: The sdlc link station specified is transitioning to connected state.

DLS.056

Level: UI–ERROR

Short Syntax: DLS.056 unexp rtn code from sdlc conn req = *rtn_code* for addr *link_address* nt *network ID*

Long Syntax: DLS.056 unexpected return code from sdlc connect request =*rtn_code* for address *link_address* net *network ID*

Description: The sdlc connect request function returned an unexpected return code.

DLS.057

Level: UE–ERROR

Short Syntax: DLS.057 unexp sdlc xid from dls in state *state* for addr *link_address* nt *network ID*

Long Syntax: DLS.057 unexpected sdlc xid from dls in state *state* for address *link_address* net *network ID*

Description: An unexpected xid event was received for the sdlc link station specified.

DLS.058

Level: UE–ERROR

Short Syntax: DLS.058 unexp sdlc xid3 from dls for pu 2 dev for addr *link_address* nt *network ID*

Long Syntax: DLS.058 unexpected sdlc xid3 from dls for pu 2 device for address *link_address* net *network ID*

Description: An unexpected XID–3 was received from DLS for a PU type 2device for the sdlc link station specified.

DLS.059

Level: UE–ERROR

Short Syntax: DLS.059 unexp sdlc dlc_contact from dls in state *state* for addr *link_address* nt *network ID*

Long Syntax: DLS.059 unexpected sdlc dlc_contact from dls in state *state* for address *link_address* net *network ID*

Description: An unexpected DLC_CONTACT event was received from DLS for the sdlc link station specified.

DLS.060

Level: UE–ERROR

Short Syntax: DLS.060 unexp sdlc dlc_info from dls in state *state* for addr *link_address* nt *network ID*

Long Syntax: DLS.060 unexpected sdlc dlc_info from dls in state *state* for address *link_address* net *network ID*

Description: An unexpected DLC_INFO event was received from DLS for the sdlc link station specified.

DLS.061

Level: UE–ERROR

Short Syntax: DLS.061 unexp sdlc dlc_dgrm from dls in state *state* for addr *link_address* nt *network ID*

Long Syntax: DLS.061 unexpected sdlc dlc_dgrm from dls in state *state* for address *link_address* net *network ID*

Description: An unexpected DLC_DGRM event was received from DLS for the sdlc link station specified.

DLS.062

Level: UE–ERROR

Short Syntax: DLS.062 unexp I–frame from sdlc in state *state* for addr *link_address* nt *network ID*

Long Syntax: DLS.062 unexpected I–frame from sdlc in state *state* for address *link_address* net *network ID*

Description: An unexpected I–frame was received from SDLC for the sdlc link station specified.

DLS.063

Level: UE–ERROR

Short Syntax: DLS.063 unexp UI–frame from sdlc in state *state* for addr *link_address* nt *network ID*

Long Syntax: DLS.063 unexpected UI–frame from sdlc in state *state* for address *link_address* net *network ID*

Description: An unexpected UI–frame was received from SDLC for the sdlc link station specified.

DLS.064

Level: C–INFO

Short Syntax: DLS.064 rcvd halt_dl from dls for sdlc addr *link_address* nt *network ID*

Long Syntax: DLS.064 received halt_dl for sdlc address *link_address* net *network ID*

Description: A HALT_DL event was sent from DLS for the specified sdlc link station

DLS.065

Level: C–INFO

Short Syntax: DLS.065 sdlc trans to disc pend st for addr *link_address* nt *network ID*

Long Syntax: DLS.065 sdlc transition to disconnect pending state for address *link_address* net *network ID*

Description: The sdlc link station specified is transitioning to disconnect pending state.

DLS.066

Level: UI–ERROR

Short Syntax: DLS.066 unexp rtn code from sdlc disc req = *rtn_code* addr *link_address* nt *network ID*

Long Syntax: DLS.066 unexpected return code from sdlc disconnect request= *rtn_code* addr *link_address* net *network ID*

Description: The sdlc disconnect request function returned an unexpected return code.

DLS.067

Level: C–INFO

Short Syntax: DLS.067 sdlc trans to disc st for addr *link_address* nt *network ID*

Long Syntax: DLS.067 sdlc transition to disconnect state for address *link_address* net *network ID*

Description: The sdlc link station specified is transitioning to disconnect state.

DLS.068

Level: UE–ERROR

Short Syntax: DLS.068 unexp sdlc dlc_halt_dl from dls in state *state* for addr *link_address* nt *network ID*

Long Syntax: DLS.068 unexpected sdlc dlc_halt_dl from dls in state *state* for address *link_address* net *network ID*

Description: An unexpected DLC_HALT_DL event was received from DLS for the sdlc link station specified.

DLS.069

Level: C–INFO

Short Syntax: DLS.069 cleanup timer expired for addr *link_address* nt *network ID*

Long Syntax: DLS.069 cleanup timer expired for address *link_address* net *network ID*

Description: The cleanup timer expired for the specified sdlc link station.

DLS.070

Level: UE–ERROR

Short Syntax: DLS.070 unexp sdlc cleanup timer exp in state *state* for addr *link_address* nt *network ID*

Long Syntax: DLS.070 unexpected sdlc cleanup timer expiration in state *state* for address *link_address* net *network ID*

Description: The sdlc cleanup timer expired, but the sdlc link station is in an unexpected state.

DLS.071

Level: C–INFO

Short Syntax: DLS.071 sdlc buf retry timer expired for addr *link_address* nt *network ID*

Long Syntax: DLS.071 sdlc buffer retry timer expired for address *link_address* net *network ID*

Description: The buffer retry timer expired for the specified sdlc link station.

DLS.072

Level: UE–ERROR

Short Syntax: DLS.072 unexp sdlc buf retry timer exp in state *state* for addr *link_address* nt *network ID*

Long Syntax: DLS.072 unexpected sdlc buffer retry timer expiration in state *state* for address *link_address* net *network ID*

Description: The sdlc buffer retry timer expired, but the sdlc link station is in an unexpected state.

DLS.073

Level: UE–ERROR

Short Syntax: DLS.073 unknown sdlc fsm input = event for addr *link_address* nt *network ID*

Long Syntax: DLS.073 unknown sdlc fsm input = event for address *link_address* net *network ID*

Description: The sdlc interface finite state machine was passed an unknown event.

DLS.074

Deleted: Message deleted.

DLS.075

Level: UE–ERROR

Short Syntax: DLS.075 unexp sdlc non xid3 from pu 2.1 dev for addr *link_address* nt *network ID*

Long Syntax: DLS.075 unexpected sdlc non xid3 from pu 2.1 device for address *link_address* net *network ID*

Description: An unexpected XID type was received for a PU type 2.1 device from the sdlc link station specified.

DLS.076

Deleted: Message deleted.

DLS.077

Level: UE–ERROR

Short Syntax: DLS.077 no buf for sdlc test for addr *link_address* nt *network ID*

Long Syntax: DLS.077 no buffer for sdlc test for address *link_address* net *network ID*

Description: No buffer could be obtained for sending a test frame to an SDLC link station. The operation will be retried later.

DLS.078

Level: UI–ERROR

Short Syntax: DLS.078 unexp rtn code from sdlc test req = *rtn_code* addr *link_address* nt *network ID*

Long Syntax: DLS.078 unexpected return code from sdlc test request =*rtn_code* addr *link_address* net *network ID*

Description: The sdlc test request function returned an unexpected return code.

DLS.079

Level: UE–ERROR

Short Syntax: DLS.079 no buf for sdlc xid0 for addr *link_address* nt *network ID*

Long Syntax: DLS.079 no buffer for sdlc xid0 for address *link_address* net *network ID*

Description: No buffer could be obtained for sending an xid0 frame to an SDLC link station. The operation will be retried later.

DLS.080

Level: UE–ERROR

Short Syntax: DLS.080 no buf for sdlc null xid for addr *link_address* nt *network ID*

Long Syntax: DLS.080 no buffer for sdlc null xid for address *link_address* net *network ID*

Description: No buffer could be obtained for sending a null xid frame to an SDLC link station. The operation will be retried later.

DLS.081

Level: UI–ERROR

Short Syntax: DLS.081 unexp rtn code from sdlc I frm req = *rtn_code* addr *link_address* nt *network ID*

Long Syntax: DLS.081 unexpected return code from sdlc I frame request =*rtn_code* address *link_address* net *network ID*

Description: The sdlc I frame request function returned an unexpected return code.

DLS.082

Level: UI–ERROR

Short Syntax: DLS.082 unexp rtn code from sdlc UI frm req = *rtn_code* addr *link_address* nt *network ID*

Long Syntax: DLS.082 unexpected return code from sdlc UI frame request =*rtn_code* addr *link_address* net *network ID*

Description: The sdlc UI frame request function returned an unexpected return code.

DLS.083

Level: UI–ERROR

Short Syntax: DLS.083 unexp rtn code from sdlc force rnr req = *rtn_code* addr *link_address* nt *network ID*

Long Syntax: DLS.083 unexpected return code from sdlc force rnr request =*rtn_code* addr *link_address* net *network ID*

Description: The sdlc force rnr request function returned an unexpected return code.

DLS.084

Deleted: Message deleted.

DLS.085

Deleted: Message deleted.

DLS.086

Level: UE–ERROR

Short Syntax: DLS.086 sdlc disc rcvd rsn *reason* for addr *link_address* nt *network ID*

Long Syntax: DLS.086 sdlc disconnect received reason *reason* for address *link_address* net *network ID*

Description: The specified SDLC connection was disconnected due to an error detected by the SDLC protocol.

DLS.087

Level: C–INFO

Short Syntax: DLS.087 sdlc trans to null_xid_pend st for addr *link_address* nt *network ID*

Long Syntax: DLS.087 sdlc transition to null_xid_pend state for address *link_address* net *network ID*

Description: The sdlc link station specified is transitioning to NULL_XID_PENDING state, meaning that it is awaiting a response to a NULL XID that was sent.

DLS.088

Level: C–INFO

Short Syntax: DLS.088 sdlc trans to xid_0_pend st for addr *link_address* nt *network ID*

Long Syntax: DLS.088 sdlc transition to xid_0_pend state for address *link_address* net *network ID*

Description: The sdlc link station specified is transitioning to XID_0_PENDING state, meaning that it is awaiting a response to an XID–0 that was sent.

DLS.089

Level: UE–ERROR

Short Syntax: DLS.089 DLS, TCP conn brk to *address*, DLS sess closed *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.089 DLS forwarder experience a session loss due to TCP connection to address break, origin MAC *source_mac_address*–>Target MAC *dest_mac_address*, origin SAP *source_sap*–>Target SAP *dest_sap*

Description: TCP connection to the DLS neighbor went down. All the DLS sessions active on that TCP connection are brought down individually.

DLS.090

Deleted: Message deleted.

DLS.091

Deleted: Message deleted.

DLS.092

Deleted: Message deleted.

DLS.093

Deleted: Message deleted.

DLS.094

Deleted: Message deleted.

DLS.095

Level: UI–ERROR

Short Syntax: DLS.095 DLS session inappropriately absent for the event

Long Syntax: DLS.095 DLS session is absent inappropriately for the event received

Description: While processing an event, either an SSP message received over TCP or an event presented from underlying DLC, the DLS state machine discovered that a valid session is absent which it should not be.

DLS.096

Level: U–INFO

Short Syntax: DLS.096 DLS, DL_STARTED event in cir–est or cir–restart state, ignore

Long Syntax: DLS.096 DLS DLC_DL_STARTED event from underlying DLCST in circuit established or circuit restart state

Description: DLS state machine received DLC_DL_STARTED event indication from the underlying LINK (LLC or SDLC) when the circuit has already been established. This could be the dribbling TEST responses coming in through bridge multipath. Ignore them.

DLS.097

Level: UI–ERROR

Short Syntax: DLS.097 DLS, No memory available to create DLS session.

Long Syntax: DLS.097 DLS, No memory available to create DLS session.

Description: There is no memory available to allocate the resources that are required to create a DLS session.

DLS.098

Level: C–INFO

Short Syntax: DLS.098 DLS, DLC_RESOLVE_C firewalled for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.098 DLS, DLC_RESOLVE_C is firewalled for the DLS sessionwith data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: The TEST command frame received from the underlying DATALINK is being enqueued in the firewall queue. This is true for the case when there is already a CANUREACH SSP message dispatched and is awaiting for the ICANREACH response.

DLS.099

Deleted: Message deleted.

DLS.100

Level: U–INFO

Short Syntax: DLS.100 DLS, Unidirected CANUREACH to *ip_address* failed for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.100 DLS, Unidirected CANUREACH to *ip_address* failed for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing TEST(c) for a given destination, DLS sends out a CANUREACH to specific DLS partner(s) if it knows that destination station is connected to one of its LAN. If this operation fails, for some difficulty within TCP connection,this event is produced. This does not necessarily result in termination of the DLS session as there can be success sending to other DLS partner.

DLS.101

Level: C–INFO

Short Syntax: DLS.101 DLS, Unidirected CANUREACH to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.101 DLS, Unidirected CANUREACH to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing TEST(c) for a given destination, DLS sent out a CANUREACH to specific DLS partner(s) if it knows that destination station is connected to one of its LAN.

DLS.102

Level: C–INFO

Short Syntax: DLS.102 DLS, Broadcast CANUREACH sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.102 DLS, Broadcast CANUREACH sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing TEST(c) for a given destination, DLS sent out broadcast CANUREACH to all the DLS partner(s) it has TCP connection established with.

DLS.103

Level: UI–ERROR

Short Syntax: DLS.103 DLS, No memory for to create address entry in cache table

Long Syntax: DLS.103 DLS, No memory for to create address entry in cache table

Description: There is no memory left to allocate the resources that are required to create an IP entry in the address table database.

DLS.104

Level: C–INFO

Short Syntax: DLS.104 DLS, SAPs resolved for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.104 DLS, SAPs resolved for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS connection can get established without using the specific SAPs, for example, SAP 0 as DSAP or SSAP. However, when the specific SAPs are used, the same connection's SAPs are updated.

DLS.105

Level: UI–ERROR

Short Syntax: DLS.105 DLS, No memory for to create IP entry in cache table

Long Syntax: DLS.105 DLS, No memory for to create IP entry in cache table

Description: There is no memory left to allocate the resources that are required to create an IP entry in the address table database.

DLS.106

Level: UI–ERROR

Short Syntax: DLS.106 DLS, rcvd CANUREACH not proc by any DLCsfor *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.106 DLS, received CANUREACH could not be processed by any DLC for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A CANUREACH SSP message we received over the TCP from a DLS Peer. However, none of the underlying data link layer, SDLC or LLC, could translate this to a TEST(c) frame.

DLS.107

Level: C–INFO

Short Syntax: DLS.107 DLS, Learning new MAC–IP assoc from *IPaddr* for *MACaddr*

Long Syntax: DLS.107 DLS, Learning new MAC–IP association from *IPaddr* for *MACaddr*

Description: DLS is learning New MAC and IP association from an SSP message received from the peer DLS. This typically occurs during CANUREACH or ICANREACH message receipt time.

DLS.108

Level: UE–ERROR

Short Syntax: DLS.108 DLS, unsolicited ICANREACH, from *IPaddr* for *MACaddr*, Ignored

Long Syntax: DLS.108 DLS, unsolicited ICANREACH message, from *IPaddr* for *MACaddr*, Ignored

Description: DLS has received an unanticipated ICANREACH message. ICAN–REACH SSP message is usually expected only as a response to a CANUREACH SSP message sent earlier. The DLS state machine will not process this message. However, it still will go ahead and learn the IP address and MAC address association.

DLS.109

Level: U–INFO

Short Syntax: DLS.109 DLS, secondary ICANREACH, from *IPaddr* for *MACaddr*, Ignored

Long Syntax: DLS.109 DLS, secondary ICANREACH message, from *IPaddr* for *MACaddr*, Ignored

Description: DLS has received a secondary ICANREACH message from other DLS peer(s) which also claims reachability to the MAC address through him(them). This is normal. The DLS state machine takes note of these DLS peers. But no further communication takes place.

DLS.110

Level: C–TRACE

Short Syntax: DLS.110 DLS, Flushing Firewall Queue to *MACaddr*

Long Syntax: DLS.110 DLS, Flushing Firewall Queue to *MACaddr*

Description: DLS has received an ICANREACH message from one or more DLS peers. This has triggered sending of pending CANUREACH to the same destination MAC address. These messages were present in the firewall queue.

DLS.111

Deleted: Message deleted.

DLS.112

Deleted: Message deleted.

DLS.113

Level: UI–ERROR
Short Syntax: DLS.113 LLC, Initialization FAILED for SAP *Sap*
Long Syntax: DLS.113 LLC, Initialization FAILED for SAP *Sap*
Description: Due to some problems, SAP initialization with LLC failed.

DLS.114

Level: UI–ERROR
Short Syntax: DLS.114 LLC, parameter validation FAILED for SAP *Sap*, rsn =*reason*
Long Syntax: DLS.114 LLC, parameter validation FAILED for SAP *Sap*, rsn =*reason*
Description: LLC tunable parameters are out of range.

DLS.115

Level: UI–ERROR
Short Syntax: DLS.115 LLC, No Memory for SAP control block for SAP *Sap*
Long Syntax: DLS.115 LLC, No Memory for SAP control block for SAP *Sap*
Description: No memory available for the SAP control block.

DLS.116

Level: UI–ERROR
Short Syntax: DLS.116 LLC, Opening of SAP *Sap* FAILED, rsn = *reason*
Long Syntax: DLS.116 LLC, Opening of SAP *Sap* FAILED, rsn = *reason*
Description: Opening of SAP failed due to problems in the LLC. Reason code is indicative of the specific problem.

DLS.117

Level: C–INFO
Short Syntax: DLS.117 LLC, Closing SAP *Sap*
Long Syntax: DLS.117 LLC, Closing SAP *Sap*
Description: Closing SAP with the LLC.

DLS.118

Level: UI–ERROR
Short Syntax: DLS.118 LLC, FAILED open stn, invalid sapcb, dst=*Destination*, src=*Source*, dsap=*Dsap*, ssap=*Ssap*
Long Syntax: DLS.118 LLC, FAILED open stn, invalid sapcb, dst=*Destination*, src=*Source*, dsap=*Dsap*, ssap=*Ssap*
Description: Opening of station for LLC data link services failed because the SAP under which station to be opened is invalid.

DLS.119

Level: UI–ERROR

Short Syntax: DLS.119 LLC, FAILED open stn, No memory, dst=*Destination*, src=*Source*, dsap=*Dsap*, ssap=*Ssap*

Long Syntax: DLS.119 LLC, FAILED open stn, No memory, dst=*Destination*, src=*Source*, dsap=*Dsap*, ssap=*Ssap*

Description: Opening of station for LLC data link services failed because there is no memory available to create control block to manage the connection.

DLS.120

Level: UI–ERROR

Short Syntax: DLS.120 LLC, FAILED open stn, rsn=*Reason*, dst=*Destination*, src=*Source*, dsap=*Dsap*, ssap=*Ssap*

Long Syntax: DLS.120 LLC, FAILED open stn, rsn=*Reason*, dst=*Destination*, src=*Source*, dsap=*Dsap*, ssap=*Ssap*

Description: Opening of station for LLC data link services failed due to some problems within LLC. The reason code is indicative of the specific problem.

DLS.121

Level: C–INFO

Short Syntax: DLS.121 LLC, opened stn, dst=*Destination*, src=*Source*, dsap=*Dsap*, ssap=*Ssap*

Long Syntax: DLS.121 LLC, opened stn, dst=*Destination*, src=*Source*, dsap=*Dsap*, ssap=*Ssap*

Description: Opening of station for LLC data link services succeeded.

DLS.122

Level: UI–ERROR

Short Syntax: DLS.122 LLC, open station failed during XCHANGE

Long Syntax: DLS.122 LLC, open station failed during XCHANGE

Description: Opening of station for LLC data link services failed during exchange operation. Exchange operation involves opening an already opened station for LLC services under a different SAP. The reason of failure is reported earlier.

DLS.123

Level: C–INFO

Short Syntax: DLS.123 LLC, xchange stn succeed for dst=*Destination*, src=*Source*, dsap=*Dsap*, ssap=*Ssap*

Long Syntax: DLS.123 LLC, exchange station succeeded for destination=*Destination*, source=*Source*, dsap=*Dsap*, ssap=*Ssap*

Description: Exchange of station for LLC data link services succeeded. Exchange operation involves opening an already opened station for LLC services under a different SAP.

DLS.124

Level: C–INFO

Short Syntax: DLS.124 LLC, closed stn by force *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.124 LLC, closed stn by force *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Closed station by force for LLC data link services.

DLS.125

Level: C–INFO

Short Syntax: DLS.125 LLC, closed stn quietly *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.125 LLC, closed stn quietly *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Closed station quietly for LLC data link services.

DLS.126

Level: UI–ERROR

Short Syntax: DLS.126 LLC, action Send failed, rsn=*reason*, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.126 LLC, action Send failed rsn=*reason*, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC was unsuccessful in sending out a frame. The reason indicative of the specific problem.

DLS.127

Level: U–INFO

Short Syntax: DLS.127 LLC, action became busy after sending, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.127 LLC, action became busy after sending, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC became busy after sending out a frame. This business is an honor system and the frame submitted to LLC does get accepted for sending. However, DLS takes note of this condition and refrains from sending more frames.

DLS.128

Level: U–INFO

Short Syntax: DLS.128 LLC BUSY, enq frm to tx pendQ, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.128 LLC BUSY, enqueue frame to pend queue,*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC is busy, frames received from TCP are being enqueued to the pending queue for that LLC session. When LLC exits busy condition, all the frames from the pending queue will be flushed.

DLS.129

Deleted: Message deleted.

DLS.130

Level: UI–ERROR

Short Syntax: DLS.130 LLC, frame refused, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.130 LLC, frame not proc, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Frame was not processed by the DLS as the SAP,as well as the station was not opened for LLC data link services.

DLS.131

Level: C–INFO

Short Syntax: DLS.131 LLC, frame refused, NOT switching for network *network, source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.131 LLC, frame refused, NOT switching for network *network,source_mac_address dest_mac_address source_sap dest_sap–>*, sap *–>*

Description: Frame was not processed by the DLS as the DSAP of the frame is not configured to be switched for the interface the frame was received on.

DLS.132

Deleted: Message deleted.

DLS.133

Deleted: Message deleted.

DLS.134

Level: UI–ERROR

Short Syntax: DLS.134 LLC, llcevent event not proc, handle is bad

Long Syntax: DLS.134 LLC, llcevent event not proc, handle is bad

Description: An LLC event was not processed by the DLS as the handle by the LLC to DLS was bad.

DLS.135

Level: UI–ERROR

Short Syntax: DLS.135 LLC, *llcevent* unknown event, not proc for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.135 LLC, *llcevent* unknown event, not proc for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: An unrecognizableLLC event occurred. This event is not processed.

DLS.136

Level: UI–ERROR

Short Syntax: DLS.136 LLC, *llcevent* req not proc, inv handle, for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.136 LLC, *llcevent* req not proc, inv handle, for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: An LLC request from DLS was not processed by the LLC interface module as the handle passed from DLS is bad.

DLS.137

Level: U–INFO

Short Syntax: DLS.137 LLC, not enabled, start_dl not honored, for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.137 LLC, not enabled, start_dl not honored, for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A Start DL request from DLS to LLC interface module was not honored as LLC portion of the DLS is not enabled. This is not an error. When a CANUREACH is received, DLS will issue START_DL request to data link service interface modules. If they are not configured to switch for the SAP or not configured at all, it is common to discard such request.

DLS.138

Level: UI–ERROR

Short Syntax: DLS.138 LLC, START_DL discard as err open stn, for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.138 LLC, START_DL discard as error happened during open station operation, for*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A Start DL request from DLS to LLC interface module was not honored as data link station could not be opened with LLC.

DLS.139

Level: UI–ERROR

Short Syntax: DLS.139 LLC, XID from DLS disc as err in xchng oper, for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.139 LLC, XID discard as error happened during exchange operation, for*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: When the first XID is received, the real SAPs get resolved. During this time, data link station is opened with LLC and resources get exchanged from data link station opened with NULL SAP. This exchange operation has failed and therefore XID is not processed.

DLS.140

Level: UI–ERROR

*Short Syntax***:** DLS.140 LLC, XID from DLS disc as new SAP can not be found/opend, for*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

*Long Syntax***:** DLS.140 LLC, XID discard as new SAP can not be found/opened, for*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: When the first XID is received, the real SAPs get resolved. During this time, DLS's interface module to LLC verifies the SAP for which XID is received. If SAP can not be found or can not be opened for some reason, XID processing is discontinued.

DLS.141

Deleted: Message deleted.

DLS.142

Level: UI–ERROR

*Short Syntax***:** DLS.142 LLC, event *eventname* received in bad state *statename*, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

*Long Syntax***:** DLS.142 LLC event *eventname* received in bad state *statename*, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS interface to LLC received an event in the bad state.

DLS.143

Deleted: Message deleted.

DLS.144

Level: C–INFO

*Short Syntax***:** DLS.144 LLC, Secondary TEST_R ign, for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

*Long Syntax***:** DLS.144 LLC Secondary TEST_R is ignored, for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: In Multipath bridge environment, it is possible to receive multiple responses to the TEST command sent earlier. The LLC interface module discards such TEST responses.

DLS.145

Deleted: Message deleted.

DLS.146

Deleted: Message deleted.

DLS.147

Deleted: Message deleted.

DLS.148

Level: U–INFO

*Short Syntax***:** DLS.148 LLC, *frametype* from LLC disc TCP is busy, for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

*Long Syntax***:** DLS.148 LLC, *frametype* from LLC disc TCP is busy, for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: The data frames received from the LLC is being discarded by the LLC interface module as TCP connection is congested.

DLS.149

Deleted: Message deleted.

DLS.150

Deleted: Message deleted.

DLS.151

Deleted: Message deleted.

DLS.152

Deleted: Message deleted.

DLS.153

Deleted: Message deleted.

DLS.154

Level: C–INFO

Short Syntax: DLS.154 LLC, *frame_type* frame drpped, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap* prt nt up

Long Syntax: DLS.154 LLC, *frame_type* frame drpped, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap* prt nt up

Description: A frame received from DLS cloud was not successfully sent – bridge port not up.

DLS.155

Deleted: Message deleted.

DLS.156

Level: C–INFO

Short Syntax: DLS.156 DLS session pool of count bytes created for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.156 DLS session pool of count bytes created for origin MAC *source_mac_address*–>Target MAC *dest_mac_address*, origin SAP *source_sap*–>Target SAP *dest_sap*

Description: A session pool has been created for a new DLS connection. This pool is used exclusively by this session for data transfer.

DLS.157

Level: C–INFO

Short Syntax: DLS.157 Global DLS pool of *count* bytes created

Long Syntax: DLS.157 Global DLS pool of *count* bytes created

Description: The global DLS memory pool has been created. This is used for SSP control messages and other nondata transfer related items.

DLS.158

Level: UE–ERROR

Short Syntax: DLS.158 Cannot create DLS session pool of *count* bytes for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.158 Cannot create DLS session pool of *count* bytes for origin MAC *source_mac_address*–>Target MAC *dest_mac_address*,origin SAP *source_sap*–>Target SAP *dest_sap*

Description: There is currently not enough memory available to support anew DLS session.

DLS.159

Level: UE–ERROR

Short Syntax: DLS.159 Cannot alloc global DLS pool of *count* bytes

Long Syntax: DLS.159 Cannot alloc global DLS pool of *count* bytes

Description: There is not enough memory available to support DLS. DLS has been disabled.

DLS.160

Level: C–INFO

Short Syntax: DLS.160 Entering *flow_ctrl_type* congestion for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.160 Entering *flow_ctrl_type* congestion for origin MAC *source_mac_address*–>Target MAC *dest_mac_address*, origin SAP *source_sap*–>Target SAP *dest_sap*

Description: The DLS session is congested due to either TCP backup, or the receipt of a DLS_ENTER_BUSY SSP message. If this happens too frequently, consider increasing the amount of memory allocated to each DLS session.

DLS.161

Level: C–INFO

Short Syntax: DLS.161 Entering GLOBAL congestion on global DLS pool

Long Syntax: DLS.161 Entering GLOBAL congestion on global DLS pool

Description: The total amount of memory allocated by all currently active DLS sessions has exceeded the amount preallocated by the user for DLS. As a result, the data links are temporarily being quiescent until some of the memory is freed up. If this happens too frequently, consider increasing the amount of memory allocated to DLSw.

DLS.162

Level: C–INFO

Short Syntax: DLS.162 Exiting *flow_ctrl_type* congestion for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.162 Exiting *flow_ctrl_type* congestion for origin MAC *source_mac_address*–>Target MAC *dest_mac_address*, origin SAP *source_sap*–>Target SAP *dest_sap*

Description: Sufficient memory has been freed up since the last time DLS was congested to allow the data links to receive data again.

DLS.163

Level: C–INFO

Short Syntax: DLS.163 Exiting GLOBAL congestion on global DLS pool

Long Syntax: DLS.163 Exiting GLOBAL congestion on global DLS pool

Description: Sufficient memory has been freed up since the last time DLS was congested to allow the data links to receive data again.

DLS.164

Level: U–INFO

Short Syntax: DLS.164 no slow buf for datalink–>transport copy for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.164 no slow buffer for datalink–>transport copy for origin MAC *source_mac_address*–>Target MAC *dest_mac_address*,origin SAP *source_sap*–>Target SAP *dest_sap*

Description: No buffer could be obtained for copying a data buffer for queueing while the transport is congested. The existing device buffer is queued.

DLS.165

Level: C–INFO

Short Syntax: DLS.165 DLS session pool deleted for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.165 DLS session pool deleted for origin MAC *source_mac_address*–>Target MAC *dest_mac_address*, origin SAP *source_sap*–>Target SAP *dest_sap*

Description: All buffers have been returned to a DLSw session pool after the DLSw session has been closed. The pool may now be removed.

DLS.166

Level: C–INFO

Short Syntax: DLS.166 DLS, SSP msg CANUREACH received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.166 DLS forwarder received a SSP CANUREACH message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of CANUREACH over TCP.

DLS.167

Level: C–INFO

***Short Syntax*:** DLS.167 DLS, SSP msg ICANREACH received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.167 DLS forwarder received a SSP ICANREACH message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of ICANREACH over TCP.

DLS.168

Level: C–INFO

***Short Syntax*:** DLS.168 DLS, SSP msg REACHACK received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.168 DLS forwarder received a SSP REACHACK message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of REACHACK over TCP.

DLS.169

Level: C–INFO

***Short Syntax*:** DLS.169 DLS, SSP msg XIDFRAME received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.169 DLS forwarder received a SSP XIDFRAME message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of XIDFRAME over TCP.

DLS.170

Level: C–INFO

***Short Syntax*:** DLS.170 DLS, SSP msg DGRMFRAME received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.170 DLS forwarder received a SSP DGRMFRAME message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of DGRMFRAME over TCP.

DLS.171

Level: C–INFO

***Short Syntax*:** DLS.171 DLS, SSP msg CONTACT received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.171 DLS forwarder received a SSP CONTACT message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of CONTACT over TCP.

DLS.172

Level: C–INFO

***Short Syntax*:** DLS.172 DLS, SSP msg CONTACTED received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.172 DLS forwarder received a SSP CONTACTED message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of CONTACTED over TCP.

DLS.173

Level: C–INFO

***Short Syntax*:** DLS.173 DLS, SSP msg DATAFRAME received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.173 DLS forwarder received a SSP DATAFRAME message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of DATAFRAME over TCP.

DLS.174

Level: C–INFO

***Short Syntax*:** DLS.174 DLS, SSP msg RESTART_DL received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.174 DLS forwarder received a SSP RESTART_DL message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of RESTART_DL over TCP.

DLS.175

Level: C–INFO

Short Syntax: DLS.175 DLS, SSP msg RESTARTED received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.175 DLS forwarder received a SSP RESTARTED message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of RESTARTED over TCP.

DLS.176

Level: C–INFO

Short Syntax: DLS.176 DLS, SSP msg HALT_DL received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.176 DLS forwarder received a SSP HALT_DL message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of HALT_DL over TCP.

DLS.177

Level: C–INFO

Short Syntax: DLS.177 DLS, SSP msg DL_HALTED received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.177 DLS forwarder received a SSP DL_HALTED message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of DL_HALTED over TCP.

DLS.178

Level: C–INFO

Short Syntax: DLS.178 DLS, SSP msg INFOFRAME received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.178 DLS forwarder received a SSP INFOFRAME message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of INFOFRAME over TCP.

DLS.179

Level: C–INFO

***Short Syntax*:** DLS.179 DLS, SSP msg ENTER_BUSY received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.179 DLS forwarder received a SSP ENTER_BUSY message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of ENTER_BUSY over TCP.

DLS.180

Level: C–INFO

***Short Syntax*:** DLS.180 DLS, SSP msg EXIT_BUSY received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.180 DLS forwarder received a SSP EXIT_BUSY message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of EXIT_BUSY over TCP.

DLS.181

Level: C–INFO

***Short Syntax*:** DLS.181 DLS, SSP msg HALT_DL_NOACK received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.181 DLS forwarder received a SSP HALT_DL_NOACK message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of HALT_DL_NOACK over TCP.

DLS.182

Level: C–INFO

***Short Syntax*:** DLS.182 DLS, SSP msg IAMOKAY received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.182 DLS forwarder received a SSP IAMOKAY message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of IAMOKAY over TCP.

DLS.183

Level: C–INFO

Short Syntax: DLS.183 DLS, UNRECOGNIZED_SSP received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.183 DLS forwarder received an UNRECOGNIZED_SSPmessage over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received an unrecognized Switch to Switch Protocol message over TCP.

DLS.184

Level: C–INFO

Short Syntax: DLS.184 DLS, DLC event DLC_CONTACTED received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.184 DLS forwarder received a DLC event of type DLC_CONTACTED for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a DLC_CONTACTED event from the underlying data link which could be LLC or SDLC.

DLS.185

Level: C–INFO

Short Syntax: DLS.185 DLS, DLC event DLC_ERROR received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.185 DLS forwarder received a DLC event of type DLC_ERROR for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a DLC_ERROR event from the underlying data link which could be LLC or SDLC.

DLS.186

Level: C–INFO

Short Syntax: DLS.186 DLS, DLC event DLC_RESET received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.186 DLS forwarder received a DLC event of type DLC_RESET for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a DLC_RESET event from the underlying data link which could be LLC or SDLC.

DLS.187

Level: C–INFO

Short Syntax: DLS.187 DLS, DLC event DLC_DL_HALTED received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.187 DLS forwarder received a DLC event of type DLC_DL_HALTED for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a DLC_DL_HALTED event from the underlying data link which could be LLC or SDLC.

DLS.188

Level: C–INFO

Short Syntax: DLS.188 DLS, DLC event DLC_DL_ENTER_BUSY received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.188 DLS forwarder received a DLC event of type DLC_DL_ENTER_BUSY for *source_mac_address–>dest_mac_address*,sap *source_sap–>dest_sap*

Description: DLS forwarder received a DLC_DL_ENTER_BUSY event from the underlying data link which could be LLC or SDLC.

DLS.189

Level: C–INFO

Short Syntax: DLS.189 DLS, DLC event DLC_DL_EXIT_BUSY received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.189 DLS forwarder received a DLC event of type DLC_DL_EXIT_BUSY for *source_mac_address–>dest_mac_address*,sap *source_sap–>dest_sap*

Description: DLS forwarder received a DLC_DL_EXIT_BUSY event from the underlying data link which could be LLC or SDLC.

DLS.190

Level: C–INFO

Short Syntax: DLS.190 DLS, DLC event DLC_DL_STARTED received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.190 DLS forwarder received a DLC event of type DLC_DL_STARTED for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a DLC_DL_STARTED event from the underlying data link which could be LLC or SDLC.

DLS.191

Level: C–INFO

Short Syntax: DLS.191 DLS, DLC event DLC_RESOLVE_C received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.191 DLS forwarder received a DLC event of type DLC_RESOLVE_C for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a DLC_RESOLVE_C event from the underlying data link which could be LLC or SDLC.

DLS.192

Level: C–INFO

Short Syntax: DLS.192 DLS, DLC event DLC_INFO received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.192 DLS forwarder received a DLC event of type DLC_INFO for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a DLC_INFO event from the underlying data link which could be LLC or SDLC.

DLS.193

Level: C–INFO

Short Syntax: DLS.193 DLS, DLC event DLC_DGRM received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.193 DLS forwarder received a DLC event of type DLC_DGRM for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a DLC_DGRM event from the underlying data link which could be LLC or SDLC.

DLS.194

Level: C–INFO

Short Syntax: DLS.194 DLS, DLC event DLC_XID received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.194 DLS forwarder received a DLC event of type DLC_XID for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a DLC_XID event from the underlying data link which could be LLC or SDLC.

DLS.195

Level: C–INFO

Short Syntax: DLS.195 DLS, DLC event DLC_DATAFRAME received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.195 DLS forwarder received a DLC event of type DLC_DATAFRAME for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a DLC_DATAFRAME event from the underlying data link which could be LLC or SDLC.

DLS.196

Level: C–INFO

Short Syntax: DLS.196 DLS, Transition to DISCONNECTED state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.196 DLS forwarder is transitioning to DISCONNECTED state for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an event, either an SSP message received over TCP or an event presented from underlying DLC, the associated DLS session is transitioning to DISCONNECTED state.

DLS.197

Level: C–INFO

Short Syntax: DLS.197 DLS, Transition to CONNECT_PENDING state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.197 DLS forwarder is transitioning to CONNECT_PENDING state for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an event, either an SSP message received over TCP or an event presented from underlying DLC, the associated DLS session is transitioning to CONNECT_PENDING state.

DLS.198

Level: C–INFO

Short Syntax: DLS.198 DLS, Transition to CONNECTED state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.198 DLS forwarder is transitioning to CONNECTED state for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an event, either an SSP message received over TCP or an event presented from underlying DLC, the associated DLS session is transitioning to CONNECTED state.

DLS.199

Level: C–INFO

Short Syntax: DLS.199 DLS, Transition to DISCONNECT_PENDING state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.199 DLS forwarder is transitioning to DISCONNECT_PENDING state for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an event, either an SSP message received over TCP or an event presented from underlying DLC, the associated DLS session is transitioning to DISCONNECT_PENDING state.

DLS.200

Level: C–INFO

Short Syntax: DLS.200 DLS, Transition to CIRCUIT_ESTABLISHED state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.200 DLS forwarder is transitioning to CIRCUIT_ESTABLISHED state for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an event, either an SSP message received over TCP or an event presented from underlying DLC, the associated DLS session is transitioning to CIRCUIT_ESTABLISHED state.

DLS.201

Level: C–INFO

Short Syntax: DLS.201 DLS, Transition to CIRCUIT_PENDING state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.201 DLS forwarder is transitioning to CIRCUIT_PENDING state for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an event, either an SSP message received over TCP or an event presented from underlying DLC, the associated DLS session is transitioning to CIRCUIT_PENDING state.

DLS.202

Level: C–INFO

Short Syntax: DLS.202 DLS, Transition to CIRCUIT_RESTART state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.202 DLS forwarder is transitioning to CIRCUIT_RESTART state for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an event, either an SSP message received over TCP or an event presented from underlying DLC, the associated DLS session is transitioning to CIRCUIT_RESTART state.

DLS.203

Level: C–INFO

Short Syntax: DLS.203 DLS, Transition to RESOLVE_PENDING state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.203 DLS forwarder is transitioning to RESOLVE_PENDING state for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an event, either an SSP message received over TCP or an event presented from underlying DLC, the associated DLS session is transitioning to RESOLVE_PENDING state.

DLS.204

Level: C–INFO

***Short Syntax*:** DLS.204 DLS, Transition to CONTACT_PENDING state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.204 DLS forwarder is transitioning to CONTACT_PENDING state for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an event, either an SSP message received over TCP or an event presented from underlying DLC, the associated DLS session is transitioning to CONTACT_PENDING state.

DLS.205

Level: C–INFO

***Short Syntax*:** DLS.205 DLS, Transition to RESTART_PENDING state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.205 DLS forwarder is transitioning to RESTART_PENDING state for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an event, either an SSP message received over TCP or an event presented from underlying DLC, the associated DLS session is transitioning to RESTART_PENDING state.

DLS.206

Level: C–INFO

***Short Syntax*:** DLS.206 DLS, Transition to HALT_PENDING state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

***Long Syntax*:** DLS.206 DLS forwarder is transitioning to HALT_PENDING state for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an event, either an SSP message received over TCP or an event presented from underlying DLC, the associated DLS session is transitioning to HALT_PENDING state.

DLS.207

Level: UE–ERROR

Short Syntax: DLS.207 DLS, DLC_CONTACTED rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.207 DLS forwarder received a DLC_CONTACTED event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DLC_CONTACTED event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.208

Level: UE–ERROR

Short Syntax: DLS.208 DLS, DLC_DGRM rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.208 DLS forwarder received a DLC_DGRM event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DLC_DGRM event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.209

Level: UE–ERROR

Short Syntax: DLS.209 DLS, DLC_ERROR rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.209 DLS forwarder received a DLC_ERROR event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DLC_ERROR event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.210

Level: UE–ERROR

Short Syntax: DLS.210 DLS, DLC_INFO rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.210 DLS forwarder received a DLC_INFO event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DLC_INFO event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.211

Level: UE–ERROR

Short Syntax: DLS.211 DLS, DLC_DL_HALTED rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.211 DLS forwarder received a DLC_DL_HALTED event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DLC_DL_HALTED event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.212

Level: UE–ERROR

Short Syntax: DLS.212 DLS, DLC_DL_STARTED rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.212 DLS forwarder received a DLC_DL_STARTED event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DLC_DL_STARTED event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.213

Level: UE–ERROR

Short Syntax: DLS.213 DLS, DLC_RESET rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.213 DLS forwarder received a DLC_RESET event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DLC_RESET event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.214

Level: UE–ERROR

Short Syntax: DLS.214 DLS, DLC_RESOLVE_C rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.214 DLS forwarder received a DLC_RESOLVE_C event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DLC_RESOLVE_C event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.215

Level: UE–ERROR

Short Syntax: DLS.215 DLS, DLC_XID rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.215 DLS forwarder received a DLC_XID event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DLC_XID event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.216

Level: UE–ERROR

Short Syntax: DLS.216 DLS, CANUREACH rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.216 DLS forwarder received a CANUREACH event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a CANUREACH event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.217

Level: UE–ERROR

Short Syntax: DLS.217 DLS, DGRMFRAME rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.217 DLS forwarder received a DGRMFRAME event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DGRMFRAME event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.218

Level: UE–ERROR

Short Syntax: DLS.218 DLS, XIDFRAME rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.218 DLS forwarder received a XIDFRAME event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a XIDFRAME event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.219

Level: UE–ERROR

Short Syntax: DLS.219 DLS, DATAFRAME rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.219 DLS forwarder received a DATAFRAME event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DATAFRAME event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.220

Level: UE–ERROR

Short Syntax: DLS.220 DLS, CONTACT rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.220 DLS forwarder received a CONTACT event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a CONTACT event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.221

Level: UE–ERROR

Short Syntax: DLS.221 DLS, CONTACTED rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.221 DLS forwarder received a CONTACTED event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a CONTACTED event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.222

Level: UE–ERROR

Short Syntax: DLS.222 DLS, RESTART_DL rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.222 DLS forwarder received a RESTART_DL event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a RESTART_DL event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.223

Level: UE–ERROR

Short Syntax: DLS.223 DLS, DL_RESTARTED rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.223 DLS forwarder received a DL_RESTARTED event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DL_RESTARTED event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.224

Level: UE–ERROR

Short Syntax: DLS.224 DLS, INFOFRAME rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.224 DLS forwarder received a INFOFRAME event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an INFOFRAME event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.225

Level: UE–ERROR

Short Syntax: DLS.225 DLS, HALT_DL rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.225 DLS forwarder received a HALT_DL event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a HALT_DL event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.226

Level: UE–ERROR

Short Syntax: DLS.226 DLS, HALT_DL_NOACK rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.226 DLS forwarder received a HALT_DL_NOACK event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a HALT_DL_NOACK event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.227

Level: UE–ERROR

Short Syntax: DLS.227 DLS, DL_HALTED rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.227 DLS forwarder received a DL_HALTED event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a DL_HALTED event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.228

Level: UE–ERROR

Short Syntax: DLS.228 DLS, ENTER_BUSY rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.228 DLS forwarder received a ENTER_BUSY event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an ENTER_BUSY event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.229

Level: UE–ERROR

Short Syntax: DLS.229 DLS, EXIT_BUSY rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.229 DLS forwarder received a EXIT_BUSY event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing an EXIT_BUSY event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.230

Level: UE–ERROR

Short Syntax: DLS.230 DLS, REACHACK rcvd in bad state *state* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.230 DLS forwarder received a REACHACK event in bad state *state* for the DLS session with data link id *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: While processing a REACHACK event, the DLS state machine discovered the event occurring in an unexpected state.

DLS.231

Level: C–INFO

Short Syntax: DLS.231 DLS, CANUREACH to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.231 DLS, CANUREACH to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a CANUREACH SSP message over TCP to its DLS peer.

DLS.232

Level: C–INFO

Short Syntax: DLS.232 DLS, ICANREACH to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.232 DLS, ICANREACH to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out an ICANREACH SSP message over TCP to its DLS peer.

DLS.233

Level: C–INFO

Short Syntax: DLS.233 DLS, REACH_ACK to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.233 DLS, REACH_ACK to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a REACH_ACK SSP message over TCP to its DLS peer.

DLS.234

Level: C–INFO

Short Syntax: DLS.234 DLS, CONTACT to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.234 DLS, CONTACT to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a CONTACT SSP message over TCP to its DLS peer.

DLS.235

Level: C–INFO

Short Syntax: DLS.235 DLS, CONTACTED to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.235 DLS, CONTACTED to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a CONTACTED SSP message over TCP to its DLS peer.

DLS.236

Level: C–INFO

Short Syntax: DLS.236 DLS, RESTART_DL to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.236 DLS, RESTART_DL to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a RESTART_DL SSP message over TCP to its DLS peer.

DLS.237

Level: C–INFO

Short Syntax: DLS.237 DLS, DL_RESTARTED to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.237 DLS, DL_RESTARTED to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a DL_RESTARTED SSP message over TCP to its DLS peer.

DLS.238

Level: C–INFO

Short Syntax: DLS.238 DLS, ENTER_BUSY to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.238 DLS, ENTER_BUSY to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out an ENTER_BUSY SSP message over TCP to its DLS peer.

DLS.239

Level: C–INFO

Short Syntax: DLS.239 DLS, EXIT_BUSY to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.239 DLS, EXIT_BUSY to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out an EXIT_BUSY SSP message over TCP to its DLS peer.

DLS.240

Level: C–INFO

Short Syntax: DLS.240 DLS, HALT_DL to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.240 DLS, HALT_DL to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a HALT_DL SSP message over TCP to its DLS peer.

DLS.241

Level: C–INFO

Short Syntax: DLS.241 DLS, DL_HALTED to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.241 DLS, DL_HALTED to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a DL_HALTED SSP message over TCP to its DLS peer.

DLS.242

Level: C–INFO

Short Syntax: DLS.242 DLS, HALT_DL_NOACK to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.242 DLS, HALT_DL_NOACK to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a HALT_DL_NOACK SSP message over TCP to its DLS peer.

DLS.243

Level: C–INFO

Short Syntax: DLS.243 DLS, TEST_CIRCUIT_RSP to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.243 DLS, TEST_CIRCUIT_RSP to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a TEST_CIRCUIT_RSP SSP message over TCP to its DLS peer.

DLS.244

Level: UI–ERROR

Short Syntax: DLS.244 DLS, FAILED to send CANUREACH to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.244 DLS, FAILED to send CANUREACH to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, a CANUREACH SSP control message could not be sent out.

DLS.245

Level: UI–ERROR

Short Syntax: DLS.245 DLS, FAILED to send ICANREACH to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.245 DLS, FAILED to send ICANREACH to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, an ICANREACH SSP control message could not be sent out.

DLS.246

Level: UI–ERROR

Short Syntax: DLS.246 DLS, FAILED to send REACH_ACK to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.246 DLS, FAILED to send REACH_ACK to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, a REACH_ACK SSP control message could not be sent out.

DLS.247

Level: UI–ERROR

Short Syntax: DLS.247 DLS, FAILED to send CONTACT to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.247 DLS, FAILED to send CONTACT to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, a CONTACT SSP control message could not be sent out.

DLS.248

Level: UI–ERROR

Short Syntax: DLS.248 DLS, FAILED to send CONTACTED to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.248 DLS, FAILED to send CONTACTED to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, a CONTACTED SSP control message could not be sent out.

DLS.249

Level: UI–ERROR

Short Syntax: DLS.249 DLS, FAILED to send RESTART_DL to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.249 DLS, FAILED to send RESTART_DL to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, a RESTART_DL SSP control message could not be sent out.

DLS.250

Level: UI–ERROR

Short Syntax: DLS.250 DLS, FAILED to send DL_RESTARTED to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.250 DLS, FAILED to send DL_RESTARTED to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, a DL_RESTARTED SSP control message could not be sent out.

DLS.251

Level: UI–ERROR

Short Syntax: DLS.251 DLS, FAILED to send ENTER_BUSY to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.251 DLS, FAILED to send ENTER_BUSY to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, an ENTER_BUSY SSP control message could not be sent out.

DLS.252

Level: UI–ERROR

Short Syntax: DLS.252 DLS, FAILED to send EXIT_BUSY to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.252 DLS, FAILED to send EXIT_BUSY to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, an EXIT_BUSY SSP control message could not be sent out.

DLS.253

Level: UI–ERROR

Short Syntax: DLS.253 DLS, FAILED to send HALT_DL to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.253 DLS, FAILED to send HALT_DL to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, a SSP control message could not be sent out.

DLS.254

Level: UI–ERROR

Short Syntax: DLS.254 DLS, FAILED to send DL_HALTED to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.254 DLS, FAILED to send DL_HALTED to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, a DL_HALTED SSP control message could not be sent out.

DLS.255

Level: UI–ERROR

Short Syntax: DLS.255 DLS, FAILED to send HALT_DL_NOACK to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.255 DLS, FAILED to send HALT_DL_NOACK to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, a HALT_DL_NOACK SSP control message could not be sent out.

DLS.256

Level: UI–ERROR

Short Syntax: DLS.256 DLS, FAILED to send TEST_CIRCUIT_RSP to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.256 DLS, FAILED to send TEST_CIRCUIT_RSP to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: Due to some problems with TCP connection to its DLS peer, a TEST_CIRCUIT_RSP SSP control message could not be sent out.

DLS.257

Deleted: Message deleted.

DLS.258

Level: C–INFO

Short Syntax: DLS.258 DLS, XIDFRAME to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.258 DLS, XIDFRAME to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a XIDFRAME SSP message over TCP to its DLS peer.

DLS.259

Level: C–INFO

Short Syntax: DLS.259 DLS, DGRMFRAME to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.259 DLS, DGRMFRAME to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a DGRMFRAME SSP message over TCP to its DLS peer.

DLS.260

Level: C–INFO

Short Syntax: DLS.260 DLS, DATAFRAME to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.260 DLS, DATAFRAME to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out a DATAFRAME SSP message over TCP to its DLS peer.

DLS.261

Level: C–INFO

Short Syntax: DLS.261 DLS, INFOFRAME to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.261 DLS, INFOFRAME to *ip_address* sent for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS successfully sent out an INFOFRAME SSP message over TCP to its DLS peer.

DLS.262

Level: C–INFO

Short Syntax: DLS.262 DLS, SSP msg TEST_CIRCUIT_REQ received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.262 DLS forwarder received a SSP TEST_CIRCUIT_REQ message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of TEST_CIRCUIT_REQ over TCP.

DLS.263

Level: C–INFO

Short Syntax: DLS.263 DLS, SSP msg TEST_CIRCUIT_RSP received from *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.263 DLS forwarder received a SSP TEST_CIRCUIT_RSP message over TCP connection to *ip_address* for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: DLS forwarder received a Switch to Switch Protocol message of TEST_CIRCUIT_RSP over TCP.

DLS.264

Level: C–INFO

Short Syntax: DLS.264 LLC, Flushed Info frame sent, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.264 LLC, Flushed info frame sent, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: An information frame received from DLS cloud was successfully flushed to a LLC end station.

DLS.265

Level: C–INFO

Short Syntax: DLS.265 LLC, TEST_C frame sent, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.265 LLC, TEST_C frame sent, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A TEST Command frame was successfully sent to a LLC end station as as result of receiving a CANUREACH from a DLSw peer router.

DLS.266

Level: C–INFO

Short Syntax: DLS.266 LLC, TEST_R frame sent, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.266 LLC, TEST_R frame sent, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A TEST Response frame was successfully sent to a LLC end station as as result of receiving an ICANREACH from a DLSw peer router.

DLS.267

Level: C–INFO

Short Syntax: DLS.267 LLC, XID frame sent, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.267 LLC, XID frame sent, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A XID frame was successfully sent to a LLC end station.

DLS.268

Level: C–INFO

Short Syntax: DLS.268 LLC, UI frame sent, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.268 LLC, UI frame sent, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: An UI frame was successfully sent to a LLC end station.

DLS.269

Level: U–INFO

Short Syntax: DLS.269 LLC, TEST_C frame refused by st mch, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.269 LLC,TEST_C frame refused by st mch, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A TEST command frame was not processed by the DLS–LLC interface module's state machine as it could not be successfully relayed through DLS. This instance is normal for TEST command frame as DLS copies the frame and still refuses the frame so that it can also be sent through the bridge path.

DLS.270

Level: U–INFO

Short Syntax: DLS.270 LLC, TEST_R frame refused by st mch, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.270 LLC,TEST_R frame refused by st mch, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A TEST response frame was not processed by the DLS–LLC interface module's state machine as it could not be successfully relayed through DLS.

DLS.271

Level: U–INFO

Short Syntax: DLS.271 LLC, XID_C frame refused by st mch, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.271 LLC,XID_C frame refused by st mch, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A XID command frame was not processed by the DLS–LLC interface module's state machine as it could not be successfully relayed through DLS.

DLS.272

Level: U–INFO

Short Syntax: DLS.272 LLC, XID_R frame refused by st mch, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.272 LLC,XID_R frame refused by st mch, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A XID response frame was not processed by the DLS–LLC interface module's state machine as it could not be successfully relayed through DLS.

DLS.273

Level: U–INFO

Short Syntax: DLS.273 LLC, UI frame refused by st mch, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.273 LLC,UI frame refused by st mch, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: An UI frame was not processed by the DLS–LLC interface module's state machine as it could not be successfully relayed through DLS.

DLS.274

Level: U–INFO

Short Syntax: DLS.274 LLC, INFO frame refused by st mch, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.274 LLC,INFO frame refused by st mch, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: An INFO frame was not processed by the DLS–LLC interface module's state machine as it could not be successfully relayed through DLS.

DLS.275

Level: UI–ERROR

Short Syntax: DLS.275 LLC, Connect Indication Passive event not proc,*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.275 LLC, Connect Indication Passive event not proc,*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC Connect Indication Passive event was not processed. Passive opens are not supported in the DLSw environment.

DLS.276

Level: C–INFO

Short Syntax: DLS.276 LLC, event LLCIM_NETUP received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.276 LLC, event LLCIM_NETUP received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_NETUP event.

DLS.277

Level: C–INFO

Short Syntax: DLS.277 LLC, event LLCIM_NETDOWN received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.277 LLC, event LLCIM_NETDOWN received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_NETDOWN event.

DLS.278

Level: C–INFO

Short Syntax: DLS.278 LLC, event LLCIM_LLC_CONNECT_IND received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.278 LLC, event LLCIM_LLC_CONNECT_IND received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_CONNECT_IND event from LLC.

DLS.279

Level: C–INFO

Short Syntax: DLS.279 LLC, event LLCIM_LLC_CONNECT_CONF received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.279 LLC, event LLCIM_LLC_CONNECT_CONF received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_CONNECT_CONF event from LLC.

DLS.280

Level: C–INFO

Short Syntax: DLS.280 LLC, event LLCIM_LLC_ENTER_BUSY received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.280 LLC, event LLCIM_LLC_ENTER_BUSY received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_ENTER_BUSY event from LLC.

DLS.281

Level: C–INFO

Short Syntax: DLS.281 LLC, event LLCIM_LLC_EXIT_BUSY received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.281 LLC, event LLCIM_LLC_EXIT_BUSY received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_EXIT_BUSY event from LLC.

DLS.282

Level: C–INFO

Short Syntax: DLS.282 LLC, event LLCIM_LLC_DISC_IND received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.282 LLC, event LLCIM_LLC_DISC_IND received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_DISC_IND event from LLC.

DLS.283

Level: C–INFO

Short Syntax: DLS.283 LLC, event LLCIM_LLC_DISC_CONF received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.283 LLC, event LLCIM_LLC_DISC_CONF received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_DISC_CONF event from LLC.

DLS.284

Level: C–INFO

Short Syntax: DLS.284 LLC, event LLCIM_LLC_RESET_IND received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.284 LLC, event LLCIM_LLC_RESET_IND received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_RESET_IND event from LLC.

DLS.285

Level: C–INFO

Short Syntax: DLS.285 LLC, event LLCIM_LLC_TEST_C received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.285 LLC, event LLCIM_LLC_TEST_C received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_TEST_C event from LLC.

DLS.286

Level: C–INFO

Short Syntax: DLS.286 LLC, event LLCIM_LLC_TEST_R received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.286 LLC, event LLCIM_LLC_TEST_R received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_TEST_R event from LLC.

DLS.287

Level: C–INFO

Short Syntax: DLS.287 LLC, event LLCIM_LLC_XID_C received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.287 LLC, event LLCIM_LLC_XID_C received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_XID_C event from LLC.

DLS.288

Level: C–INFO

Short Syntax: DLS.288 LLC, event LLCIM_LLC_XID_R received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.288 LLC, event LLCIM_LLC_XID_R received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_XID_R event from LLC.

DLS.289

Level: C–INFO

Short Syntax: DLS.289 LLC, event LLCIM_LLC_UI received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.289 LLC, event LLCIM_LLC_UI received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_UI event from LLC.

DLS.290

Level: C–INFO

Short Syntax: DLS.290 LLC, event LLCIM_DLC_START_DL received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.290 LLC, event LLCIM_DLC_START_DL received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_DLC_START_DL event from DLS.

DLS.291

Level: C–INFO

Short Syntax: DLS.291 LLC, event LLCIM_DLC_RESOLVE_R received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.291 LLC, event LLCIM_DLC_RESOLVE_R received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_DLC_RESOLVE_R event from DLS.

DLS.292

Level: C–INFO

Short Syntax: DLS.292 LLC, event LLCIM_DLC_CONTACT received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.292 LLC, event LLCIM_DLC_CONTACT received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_DLC_CONTACT event from DLS.

DLS.293

Level: C–INFO

Short Syntax: DLS.293 LLC, event LLCIM_DLC_DGRM received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.293 LLC, event LLCIM_DLC_DGRM received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_DLC_DGRM event from DLS.

DLS.294

Level: C–INFO

Short Syntax: DLS.294 LLC, event LLCIM_DLC_XID received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.294 LLC, event LLCIM_DLC_XID received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_DLC_XID event from DLS.

DLS.295

Level: C–INFO

Short Syntax: DLS.295 LLC, event LLCIM_DLC_HALT_DL received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.295 LLC, event LLCIM_DLC_HALT_DL received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_DLC_HALT_DL event from DLS.

DLS.296

Level: C–INFO

Short Syntax: DLS.296 LLC, event LLCIM_DLC_ENTER_BUSY received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.296 LLC, event LLCIM_DLC_ENTER_BUSY received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_DLC_ENTER_BUSY event from DLS.

DLS.297

Level: C–INFO

Short Syntax: DLS.297 LLC, event LLCIM_DLC_EXIT_BUSY received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.297 LLC, event LLCIM_DLC_EXIT_BUSY received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_DLC_EXIT_BUSY event from DLS.

DLS.298

Level: C–INFO

Short Syntax: DLS.298 LLC, event LLCIM_LLC_INFO received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.298 LLC, event LLCIM_LLC_INFO received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_LLC_INFO event from LLC.

DLS.299

Level: C–INFO

Short Syntax: DLS.299 LLC, event LLCIM_DLC_INFO received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.299 LLC, event LLCIM_DLC_INFO received for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS received a LLCIM_DLC_INFO event from DLS.

DLS.300

Level: C–INFO

Short Syntax: DLS.300 LLC, Transition to LLCIM_CONTACTED state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.300 LLC Transition to LLCIM_CONTACTED state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS is transitioning to the LLCIM_CONTACTED state.

DLS.301

Level: C–INFO

Short Syntax: DLS.301 LLC, Transition to LLCIM_RESOLVE_PEND state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.301 LLC Transition to LLCIM_RESOLVE_PEND state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS is transitioning to the LLCIM_RESOLVE_PEND state.

DLS.302

Level: C–INFO

Short Syntax: DLS.302 LLC, Transition to LLCIM_CONNECTED state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.302 LLC Transition to LLCIM_CONNECTED state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS is transitioning to the LLCIM_CONNECTED state.

DLS.303

Level: C–INFO

Short Syntax: DLS.303 LLC, Transition to LLCIM_CONNECT_PEND state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.303 LLC Transition to LLCIM_CONNECT_PEND state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS is transitioning to the LLCIM_CONNECT_PEND state.

DLS.304

Level: C–INFO

Short Syntax: DLS.304 LLC, Transition to LLCIM_CONTACT_PEND state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.304 LLC Transition to LLCIM_CONTACT_PEND state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS is transitioning to the LLCIM_CONTACT_PEND state.

DLS.305

Level: C–INFO

Short Syntax: DLS.305 LLC, Transition to LLCIM_DISCONNECTED state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.305 LLC Transition to LLCIM_DISCONNECTED state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS is transitioning to the LLCIM_DISCONNECTED state.

DLS.306

Level: C–INFO

Short Syntax: DLS.306 LLC, Transition to LLCIM_DISC_PEND state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.306 LLC Transition to LLCIM_DISC_PEND state for *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: LLC interface module for the DLS is transitioning to the LLCIM_DISC_PEND state.

DLS.307

Level: UE–ERROR

Short Syntax: DLS.307 DLSw disabled no mem for buffers

Long Syntax: DLS.307 DLSw forwarder disabled no memory for buffers

Description: The Data Link Switching forwarder has been disabled because there was not enough memory to create the DLS private buffer pool.

DLS.308

Level: UE–ERROR

Short Syntax: DLS.308 DLSw disabled no mem for llcim struct

Long Syntax: DLS.308 DLSw forwarder disabled no memory for llcim structures

Description: The Data Link Switching forwarder has been disabled because there was not enough memory to create necessary llcim data structures.

DLS.309

Level: UE–ERROR

Short Syntax: DLS.309 DLSw disabled no mem for tcpim struct

Long Syntax: DLS.309 DLSw forwarder disabled no memory for tcpim structures

Description: The Data Link Switching forwarder has been disabled because there was not enough memory to create necessary tcpim data structures.

DLS.310

Level: UE–ERROR

Short Syntax: DLS.310 DLSw disabled no mem for sdlcim struct

Long Syntax: DLS.310 DLSw forwarder disabled no memory for sdlcim structures

Description: The Data Link Switching forwarder has been disabled because there was not enough memory to create necessary sdlcim data structures.

DLS.311

Level: UE–ERROR

Short Syntax: DLS.311 DLSw disabled no mem for group struct

Long Syntax: DLS.311 DLSw forwarder disabled no memory for group protocol structures

Description: The Data Link Switching forwarder has been disabled because there was not enough memory to create necessary group protocol data structures.

DLS.312

Level: UE–ERROR

Short Syntax: DLS.312 DLSw disabled no mem for dl corr array

Long Syntax: DLS.312 DLSw forwarder disabled no memory for dl correlator array

Description: The Data Link Switching forwarder has been disabled because there was not enough memory to create necessary dl correlator array.

Cause: Cannot allocate necessary memory for the dl correlator array.

Action: Reduce the maximum number of DLSw sessions.

DLS.313

Level: C–INFO

Short Syntax: DLS.313 LLC, INFO frame sent, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.313 LLC, INFO frame sent, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: An INFO frame received from DLS cloud was successfully sent to a LLC end station.

DLS.314

Level: C–INFO

Short Syntax: DLS.314 TCP, cfg xmit buf too large for group *group*, clipped to *transmit_buffer_size*

Long Syntax: DLS.314 TCP, configured transmit buffer size too large for group *group*, clipped to *transmit_buffer_size*

Description: The user configured a TCP transmit buffer size in the group configuration that cannot be handled by the router. It has automatically been set to a lower value than can be allocated by the router.

DLS.315

Level: C–INFO

Short Syntax: DLS.315 TCP, cfg xmit buf too large for *neighbor_address*,clipped to *transmit_buffer_size*

Long Syntax: DLS.315 TCP, configured transmit buffer size too large for *neighbor_address*, clipped to *transmit_buffer_size*

Description: The user configured a TCP transmit buffer size that cannot be handled by the router. It has automatically been set to a lower value than can be allocated by the router.

DLS.316

Level: C–INFO

Short Syntax: DLS.316 DLS, CANUREACH frame coll, frame ign, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.316 DLS, CANUREACH frame collision, frame ignored, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A CANUREACH frame was received from the DLS cloud, but could not be processed because a CANUREACH is already outstanding from this router for the MAC addresses and SAPs specified in the CANUREACH, and the origin MAC address for the existing circuit is greater than the origin MAC address specified in the CANUREACH.

DLS.317

Level: C–INFO

Short Syntax: DLS.317 LLC, XID_C frame sent, len=*xid_data_len*, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.317 LLC, XID_C frame sent, len=*xid_data_len*, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: An XID_C frame was successfully sent to a LLC end station.

DLS.318

Level: C–INFO

Short Syntax: DLS.318 LLC, XID_R frame sent, len=*xid_data_len*,*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.318 LLC, XID_R frame sent, len=*xid_data_len*,*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: An XID_R frame was successfully sent to a LLC end station.

DLS.319

Level: C–INFO

Short Syntax: DLS.319 LLC, XID_C dropped, len=*xid_data_len*,*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.319 LLC, XID_C dropped, len=*xid_data_len*,*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A received XID_C frame was discarded because a previously received XID_C is being processed by DLSw. This XID_C is considered a duplicate. This will occur frequently since the end station retries XID_Cs.

DLS.320

Level: C–INFO

Short Syntax: DLS.320 LLC, XID_R dropped, len=*xid_data_len*,*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.320 LLC, XID_R dropped, len=*xid_data_len*, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A received XID_R frame was discarded because there is no XID command outstanding. This occurs normally since DLSw retries XID_Cs and it is possible for many XID_Rs to comeback.

DLS.321

Level: UI–ERROR

Short Syntax: DLS.321 LLC, XIDFRAME dropped – bad XID state, len=*xid_data_len*, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.321 LLC, XIDFRAME dropped – bad XID state, len=*xid_data_len*, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: An unexpected received XIDFRAME frame was discarded. The LLC does not normally expect to receive an XID in this state.

Action: None, unless you are having a problem establishing connections between the end stations described in the message.

DLS.322

Level: C–INFO

Short Syntax: DLS.322 LLC, unexpected null XID, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.322 LLC, unexpected null XID, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A null XID has been received after the SABME/UA exchange. This is not normal in this state and indicates that the end station is trying to start a new session. The current DLSw session will be terminated.

DLS.323

Level: C–INFO

Short Syntax: DLS.323 DLS, activ XIDFRAME dropped–bad state,len=*xid_data_len*, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.323 DLS, activ XIDFRAME dropped–bad state,len=*xid_data_len*, *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A received activation XIDFRAME is not allowed in the DLSw DLS_CONNECTED or DLS_CONNECT_PENDING state. The frame is discarded.

DLS.324

Level: C–INFO

Short Syntax: DLS.324 DLS, activ XID dropped–bad state, len=*xid_data_len*,*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.324 DLS, activ XID dropped–bad state, len=*xid_data_len*,*source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: A received activation XID is not allowed in the DLSw DLS_CONNECTED or DLS_CONNECT_PENDING state. The frame is discarded.

DLS.325

Level: UI–ERROR

Short Syntax: DLS.325 DLS, Session not created – Maximum Number of DLS Sessions exceeded.

Long Syntax: DLS.325 DLS, Session not created – Maximum Number of DLS Sessions exceeded.

Description: A DLSw Session is not created since the configured Maximum Number of DLSw sessions is exceeded.

DLS.326

Level: UI–ERROR

Short Syntax: DLS.326 close transport cnn to *ip_address*, unrecoverable SSP sync error

Long Syntax: DLS.326 closing transport connection to *ip_address*, unrecoverable SSP synchronization error

Description: Due to an error receiving a previous DLSw SSP message, DLSw cannot find the message header for the next message. In order to recover, the TCP session must be closed. This could possibly happen due to an invalid message length in the previous SSP message.

DLS.327

Level: UE–ERROR

Short Syntax: DLS.327 DL_HALTED timer expired, closing session *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Long Syntax: DLS.327 DL_HALTED timer expired, closing session *source_mac_address–>dest_mac_address*, sap *source_sap–>dest_sap*

Description: An expected DL_HALTED SSP message has not been received in response to a previously sent HALT_DL message. As a result,the connection is now considered to be in the disconnected state.

DN

Digital Network Architecture Phase IV

This chapter describes Digital™ Network Architecture Phase IV packet forwarder messages. For information about message content and how to use the message, refer to the Introduction.

DN.001

Level: UE–ERROR

Short Syntax: DN.001 event 4.0: Aged pkt loss; *source_area.source_node –>destination_area.destination_node*

Long Syntax: DN.001 event 4.0: Aged packet loss; packet from *source_area.source_node* to *destination_area.destination_node*

Description: A packet has had too many visits through routers going between the specified nodes. If return to sender was re–quested, the packet will be returned to the originator. Otherwise, it will be dropped.

Cause: The router's EXECUTOR MAXIMUM VISITS is too small.

Action: Increase EXECUTOR MAXIMUM VISITS to be larger the number of hops between the two most distant nodes in the network.

Cause: There is a temporary routing loop due to an unreachable node.

Action: Unless the problem is persistent, there should be no need for corrective action. Routing loops usually go away within a minute when a node goes down.

DN.002

Level: CE–ERROR

Short Syntax: DN.002 event 4.1: Node unreach pkt loss;*source_area.source_node –>destination_area.destination_node,* cir *number* net *network_name*

Long Syntax: DN.002 event 4.1: Node unreachable packet loss; packet from *source_area.source_node* to *destination_area.destination_node,* circuit *number* network *network_name*

Description: Packet was received on the specified network for unreachable destination. If return to sender was requested, the packet will be returned to the originator. Otherwise, it will be dropped.

Cause: The originator is attempting to contact a non–existent node.

Action: If the originator supplied a host address, it should be corrected. If the originator supplied a host name, the node name to the address translation may be out of date. Use the DEFINE NODE “name” ADDRESS command on the originating node to correct the permanent database.

Cause: There is no route to the destination node in the routing database.

Action: Do a SHOW ACTIVE NODES to see if the destination node is reachable. Check the circuit(s) that could be used to reach this node.

Cause: There is no route to the destination area in the routing database.

Action: Do a SHOW ACTIVE AREA to see if the area of the destination node is reachable. Check the circuit(s) that could be used to reach this area.

DN.003

Level: UI–ERROR

Short Syntax: DN.003 event 4.2: Node out–of–range pkt loss;*source_area.source_node –>destination_area.destination_node,* cir *number* net *network_name*

Long Syntax: DN.003 event 4.2: Node out–of–range packet loss; packet from *source_area.source_node* to *destination_area.destination_node,* circuit *number* network *network_name*

Description: Packet was received on the specified network for node ad–dress beyond EXECUTOR MAXIMUM ADDRESS. If return to sender was requested, the packet will be returned to the originator. Otherwise, it will be dropped.

Cause: EXECUTOR MAXIMUM ADDRESS set too low.

Action: Increase EXECUTOR MAXIMUM ADDRESS.

Cause: Destination node's EXECUTOR NODE ADDRESS set too high.

Action: Decrease destination node's EXECUTOR NODE ADDRESS.

Cause: The originator is attempting to contact a non–existent node,which also has too high an address.

Action: If the originator supplied a host address, it should be corrected. If the originator supplied a host name, the node name to address translation may be out of date. Use the DE–FINE NODE "name" ADDRESS command on the originating node to correct the permanent database.

DN.004

Level: UE–ERROR

Short Syntax: DN.004 event 4.3: Ovsize pkt loss; *source_area.source_node–> destination_area.destination_node,* cir *number* net *network_name*

Long Syntax: DN.004 event 4.3: Oversized packet loss; packet from *source_area.source_node* to *destination_area.destination_node,* circuit *number* network *network_name*

Description: Packet was received that is larger than the block size of the output circuit chosen to the destination. The packet will be dropped.

Cause: Originating host has a larger EXECUTOR BUFFER SIZE than the receiving host can accept.

Action: Correct EXECUTOR BUFFER SIZE on originating host.

Cause: Intervening circuit has too small a packet size.

Action: Ensure that originating host's EXECUTOR BUFFER SIZE is smaller than the circuit with the lowest packet size.(Since Ethernet has the smallest block size, this is not likely.)

DN.005

Level: UE–ERROR

Short Syntax: DN.005 event 4.4: Pkt format err; data packet *source_area.source_node –>destination_area.destination_node,* cir *number* net *network_name*

Long Syntax: DN.005 event 4.4: Packet format error; long data packet from *source_area.source_node* to *destination_area.destination_node,* circuit *number* network *network_name*

Description: A Long Data Packet was received with invalid header data, on the specified circuit. The packet will be dropped.

Cause: First 4 bytes of source or destination ID are not HIORD.

Action: Correct programming error in sending node, or find source of data corruption.

Cause: The reserved D–AREA or S–AREA fields of the long data packet are not zero.

Action: Correct programming error in sending node, or find source of data corruption.

DN.006

Level: UE–ERROR

Short Syntax: DN.006 event 4.4: Pkt format err; endnode hello from *source_area.source_node*, cir *number* net *network_name*

Long Syntax: DN.006 event 4.4: Packet format error; endnode hello message from *source_area.source_node*, circuit *number* network *network_name*

Description: An Endnode Hello Message was received with invalid header data on the specified circuit. The packet will be dropped.

Cause: The node type in the IINFO field is not endnode, or the first 4 bytes of the ID field are not HIORD.

Action: Correct programming error in sending node, or find source of data corruption.

DN.007

Level: UE–ERROR

Short Syntax: DN.007 event 4.4: Pkt format err; lvl *router_level* route from *source_area.source_node*, cir *number* net *network_name*

Long Syntax: DN.007 event 4.4: Packet format error; level *router_level* routing message from *source_area.source_node* circuit *number* network *network_name*

Description: A Level 1 or 2 Routing Message was received with a format–ting error within the routing data. The packet will be dropped. In the case of an error in the routing data, the data up to the error will be processed.

Cause: The packet ends with a SEGMENT that does not contain as many RTGINFO entries as the COUNT claims.

Action: Correct programming error in sending node, or find source of data corruption.

DN.008

Level: UE–ERROR

*Short Syntax***:** DN.008 event 4.4: Pkt format err; short pkt from *source_area.source_node*, cir *number* net *network_name*

*Long Syntax***:** DN.008 event 4.4: Packet format error; packet too short from *source_area.source_node*, circuit *number* network *network_name*

Description: A packet too short to contain its header was received. The packet will be dropped.

Cause: Long Data Packet less than 21 bytes long (excluding padding).

Cause: Endnode Hello Message less than 31 bytes long.

Cause: Endnode Hello Message not long enough to contain the test data indicated by the byte count in the test data.

Cause: Router Hello Message less than 27 bytes long.

Cause: Routing Message less than 6 bytes long.

Action: Correct programming error in sending node, or find source of data corruption.

DN.009

Level: UE–ERROR

*Short Syntax***:** DN.009 event 4.4: Pkt format err; router hello from *source_area.source_node* cir *number* net *network_name*

*Long Syntax***:** DN.009 event 4.4: Packet format error; router hello message from *source_area.source_node* circuit *number* network *network_name*

Description: A Router Hello Message was received with invalid header data. The packet will be dropped.

Cause: The node type in the INFO field is not level 1 or 2 router,or the first 4 bytes of the ID field are not HIORD.

Action: Correct programming error in sending node, or find source of data corruption.

DN.010

Level: UE–ERROR

*Short Syntax***:** DN.010 event 4.4: Pkt format err; unkn typ, cir *number* net *network_name*,hdr first 21 bytes

*Long Syntax***:** DN.010 event 4.4: Packet format error; unknown type, circuit *number* network *network_name*, header first 21 bytes

Description: A packet with an invalid or unsupported flags field was received. The first 21 bytes of the header are dumped.

Cause: The first byte of the message is not one of Long Data Pack–et, Endnode Hello, Router Hello, Level 1 Routing, or Level 2Routing.

Action: Correct programming error in sending node, or find source of data corruption.

DN.011

Deleted: Message deleted.

DN.012

Level: UE–ERROR

Short Syntax: DN.012 event 4.4: Pkt format err; vers skew, flags *FLAGS*, cir *number* net *network_name*

Long Syntax: DN.012 event 4.4: Packet format error; version skew in long data packet, flags *FLAGS*, circuit *number* network *network_name*

Description: A Long Format Data Packet was received with the version bit set in the flags field. The packet will be dropped.

Cause: Programming error in sending node, or data corruption.

DN.013

Level: CI–ERROR

Short Syntax: DN.013 event 4.5: Part rting upd loss; area *area_number* from *source_area.source_node*, cir *number* net *network_name*

Long Syntax: DN.013 event 4.5: Partial routing update loss; area *area_number* in level 2 routing message from *source_area.source_node*, circuit *number* network *network_name*

Description: A Level 2 Routing Message contained reachable routes to area(s) higher than this router's EXECUTOR MAXIMUM AREA. Only the highest reachable area will be logged. Routes to unreachable (infinite cost) areas are not complained about.

Cause: This routers EXECUTOR MAXIMUM AREA is lower than the highest reachable area in the network.

Action: Correct EXECUTOR MAXIMUM AREA, or change area number of of–fending area.

DN.014

Level:	CI–ERROR
Short Syntax:	DN.014 event 4.5: Part rting upd loss; node *node_number* from *source_area.source_node*, cir *number* net *network_name*
Long Syntax:	DN.014 event 4.5: Partial routing update loss; node *node_number* in level 1 routing message from *source_area.source_node*, circuit *number* network *network_name*
Description:	A Level 1 Routing Message contained reachable routes to node(s) higher than this router's EXECUTOR MAXIMUM ADDRESS. Only the highest reachable node will be logged. Routes to unreachable (infinite cost) nodes are not complained about.
Cause:	This routers EXECUTOR MAXIMUM ADDRESS is lower than the highest reachable node in the network.
Action:	Correct EXECUTOR MAXIMUM ADDRESS, or change node number of offending node.

DN.015

Level: UE–ERROR

Short Syntax: DN.015 event 4.11: Init fail; inval data from *source_area.source_node* cir *number* net *network_name*

Long Syntax: DN.015 event 4.11: Initialization failure, line fault; adjacent node listener received invalid data from node *source_area.source_node* circuit *number* network *network_name*

Description: The (optional) test data in an Endnode Hello Message was not valid, differing from the expected test pattern of 252 octal. The adjacency will not be accepted.

Cause: Data corruption on network.

DN.016

Level: UE–ERROR

Short Syntax: DN.016 event 4.13: Init fail; end node *source_area.source_node* out of range, cir *number* net *network_name*

Long Syntax: DN.016 event 4.13: Initialization failure, operator initiated; adjacent endnode *source_area.source_node* out of range,circuit *number* network *network_name*

Description: An Endnode Hello Message was received from the specified node, but its node address exceeds the EXECUTOR MAXIMUM AD–DRESS. No adjacency will be created.

Cause: Endnode node address too high.

Action: Correct endnode node address.

Cause: Router's EXECUTOR MAXIMUM ADDRESS too low.

Action: Increase router's EXECUTOR MAXIMUM ADDRESS.

DN.017

Level: UE–ERROR

Short Syntax: DN.017 event 4.13: Init fail; router *area.node* out of range,cir *number* net *network_name*

Long Syntax: DN.017 event 4.13: Initialization failure, operator initiated; adjacent router *area.node* out of range, circuit *number* network *network_name*

Description: A Router Hello Message was received from the specified node,but there is a problem with it's node address. The node address exceeds the EXECUTOR MAXIMUM ADDRESS or the area ad–dress exceeds the EXECUTOR MAXIMUM AREA or the node or area number is zero. No adjacency will be created.

Cause: Source router's node address too high.

Action: Correct source router's node address.

Cause: This router's EXECUTOR MAXIMUM ADDRESS too low.

Action: Increase this router's EXECUTOR MAXIMUM ADDRESS.

Cause: Source router's area address too high.

Action: Correct source router's area address.

Cause: This router's EXECUTOR MAXIMUM AREA too low.

Action: Increase this router's EXECUTOR MAXIMUM AREA.

Cause: Remote router using node or area 0.

Action: Correct programming error on remote node.

DN.018

Level: UE–ERROR

Short Syntax: DN.018 event 4.13: Init fail; blck sz size too sm frm *area.node*, cir *number* net *network_name*

Long Syntax: DN.018 event 4.13: Initialization failure, operator initiated; adjacent node block size size too small from router *area.node*, circuit *number* network *network_name*

Description: A router hello is offering a block size that is too small to support area routing. The block size must be large enough to receive a Level 2 Routing Message with all 63 areas in it. The adjacency will be rejected.

Cause: Adjacent router has a block size less than 80.

Action: Correct block size on adjacent router.

Cause: Software error in adjacent router.

Action: Correct software error.

Cause: Line error causing data corruption.

Action: Examine network error counters.

DN.019

Level: UE–ERROR

Short Syntax: DN.019 event 4.13: Init fail; vers skew (*Version_number.ECO_number.user_ECO_number*) node *area.node*, cir *number* net *network_name*

Long Syntax: DN.019 event 4.13: Initialization failure; version skew (*Version_number.ECO_number.user_ECO_number*) node *area.node*, cir *number* net *network_name*

Description: A Router Hello Message was received with a Routing Layer version *number* lower than 2.0.0. No adjacency will be created. (Messages with version numbers exceeding 2.0.0 are dropped silently, per the DECnet specifications.)

Cause: Attempt to have adjacency with Phase III router.

Action: Adjacencies with Phase III routers are not supported, reconfigure network.

DN.020

Level: U–INFO

Short Syntax: DN.020 event 4.14: Node reach change; node *area.node* reachable

Long Syntax: DN.020 event 4.14: Node reachability change; node *area.node* reachable

Description: The specified node is now reachable, either due to an end–node adjacency with the node, or by being included in a Lev–el 1 Routing Message.

DN.021

Level: U–INFO

Short Syntax: DN.021 event 4.14: Node reach change; node *area.node* unreachable

Long Syntax: DN.021 event 4.14: Node reachability change; node *area.node* unreachable

Description: The specified node is now unreachable.

Cause: Circuit to the node down.

Action: See if earlier message was circuit down (Event 5.0).

Cause: Endnode adjacency down.

Action: See if earlier message was adjacency down (Event 4.18).Could be due to node down, or due to failure of network connection on that machine.

Cause: Intervening node down.

Action: See if the necessary routers are reachable.

Cause: Node down.

Action: Verify whether node is up.

Cause: Cost to node exceeds EXECUTOR MAXIMUM COST.

Action: Verify that EXECUTOR MAXIMUM COST is large enough to span the network.

Cause: Cost to node exceeds EXECUTOR MAXIMUM HOPS.

Action: Verify that EXECUTOR MAXIMUM HOPS is large enough to span the network.

DN.022

Level: C–INFO

Short Syntax: DN.022 event 4.15: Adj up; new endnode *area.node* cir *number* net *network_name*

Long Syntax: DN.022 event 4.15: Adjacency up; new endnode *area.node* circuit *number* network *network_name*

Description: There is now an adjacency with the specified endnode on the specified network.

Cause: Received valid endnode hello message.

DN.023

Level: C–INFO

Short Syntax: DN.023 event 4.15: Adj up; new router *area.node* cir *number* net *network_name*

Long Syntax: DN.023 event 4.15: Adjacency up; new router *area.node* circuit *number* network *network_name*

Description: There is now an adjacency with the specified router on one of the directly connected networks. Level 1 (and 2) Routing Messages will now be accepted from this node.

Cause: Received valid router hello message containing this router's node–id in the R/S–LIST.

DN.024

Level: UI–ERROR

Short Syntax: DN.024 event 4.16: Adj rej; table full for end node *area.node*, cir *number* net *network_name*

Long Syntax: DN.024 event 4.16: Adjacency rejected; table too full for end node *area.node*, circuit *number* network *network_name*

Description: An Endnode Hello Message has been received from a new end–node, but there are too many endnode adjacencies, and the table is full. No adjacency will be created until another endnode adjacency times out.

Cause: There are more than EXECUTOR MAXIMUM BROADCAST NONROUTERS endnodes with adjacencies to this router.

Action: Increase EXECUTOR MAXIMUM BROADCAST NONROUTERS.

DN.025

Level: UI–ERROR

Short Syntax: DN.025 event 4.16: Adj rej; table full for rtr *source*, cir *number* net *network_name*

Long Syntax: DN.025 event 4.16: Adjacency rejected; table too full for router *source*, circuit *number* network *network_name*

Description: A Router Hello Message has been received from a new router,but there are too many router adjacencies, and the table is full. No adjacency will be created until another router adjacency times out. No routes will be accepted from this router, since there is no adjacency.

Cause: There are more than EXECUTOR MAXIMUM BROADCAST ROUTERS end–nodes with adjacencies to this router.

Action: Increase EXECUTOR MAXIMUM BROADCAST ROUTERS.

DN.026

Level: UI–ERROR

Short Syntax: DN.026 event 4.16: Adj rej; too many rtrs for node *source*, cir *number* net *network_name*

Long Syntax: DN.026 event 4.16: Adjacency rejected; too many routers for node *source*, circuit *number* network *network_name*

Description: A Router Hello Message has been received from a new router on the specified circuit, but there are too many router adjacencies on this circuit, and the table is full. No adjacency will be created until another router adjacency on this circuit times out. No routes will be accepted from this router, since there is no adjacency.

Cause: There are more than CIRCUIT MAXIMUM ROUTERS endnodes with adjacencies to this router.

Action: Increase CIRCUIT MAXIMUM ROUTERS.

DN.027

Level: U–INFO

Short Syntax: DN.027 event 4.17: Area reach change; area *area* reachable

Long Syntax: DN.027 event 4.17: Area reachability change; area *area* reachable

Description: The specified area is now reachable due to being included in a Level 2 Routing Message.

DN.028

Level:	U–INFO
Short Syntax:	DN.028 event 4.17: Area reach change; area *area* unreachable
Long Syntax:	DN.028 event 4.17: Area reachability change; area *area* unreachable
Description:	The specified area is now unreachable, due to a circuit going down, a router adjacency timing out, an endnode adjacency timing out, or by the cost to that node exceeding EXECUTOR MAXIMUM COST. A preceding message should indicate the cause.
Cause:	Circuit to the area down.
Action:	See if earlier message was circuit down (Event 5.0).
Cause:	Adjacent router down.
Action:	See if earlier message was adjacency down (Event 4.18) for the router to the area.
Cause:	Intervening router down.
Action:	See if the necessary routers are reachable.
Cause:	Cost to area exceeds EXECUTOR AREA MAXIMUM COST.
Action:	Verify that EXECUTOR AREA MAXIMUM COST is large enough to span the network.
Cause:	Hops to area exceeds EXECUTOR AREA MAXIMUM HOPS.
Action:	Verify that EXECUTOR AREA MAXIMUM HOPS is large enough to span the network.

DN.029

Level: UE–ERROR

Short Syntax: DN.029 event 4.18: Adj dwn; invalid data from *area.node* cir *number* net *network_name*

Long Syntax: DN.029 event 4.18: Adjacency down, line fault; adjacent node listener received invalid data from node *area.node* circuit *number* network *network_name*

Description: The (optional) test data in an Endnode Hello Message was not valid, differing from the expected test pattern of 252 octal. The adjacency will be taken down.

Cause: Data corruption on network, or software error in remote node.

DN.030

Level: UE–ERROR

Short Syntax: DN.030 event 4.18: Adj dwn: node *area.node* chng to endnode,cir *number* net *network_name*

Long Syntax: DN.030 event 4.18: Adjacency down: node *area.node* changed to end node, circuit *number* network *network_name*

Description: An Endnode Hello Message has been received from a node that had previously been a router adjacency. The existing router adjacency will be taken down, and an endnode adjacency created.

Cause: This would occur if the type of the of the adjacent node was changed quickly.

Action: Do not change node types without taking them down first.

Cause: Two nodes of different type at the same address.

Action: Ensure that node ID’s are unique.

DN.031

Level: UE–ERROR

Short Syntax: DN.031 event 4.18: Adj dwn: router *area.node* chng type, cir *number* net *network_name*

Long Syntax: DN.031 event 4.18: Adjacency down: router *area.node* changed TM type, cir *number* net *network_name*

Description: A Router Hello Message has been received from a node whose existing adjacency was for the other type of router (level 1versus level 2). The existing router adjacency will be taken down, and a new router adjacency created.

Cause: The type of the adjacent node was changed quickly.

Action: Do not change node types without taking them down first.

Cause: Two nodes of different type at the same address.

Action: Ensure that node ID's are unique.

DN.032

Level: C–INFO

Short Syntax: DN.032 event 4.18: Adj dwn; cir *number* net *network_name* down to node *area.node*

Long Syntax: DN.032 event 4.18: Adjacency down; circuit *number* network *network_name* down to node *area.node*

Description: The specified adjacency has gone down. All routes through this adjacency will be deleted.

Cause: The associated circuit has gone down.

Action: See if earlier message was circuit down (Event 5.0).

Cause: A Router Hello Message was received from a higher priority router.

Action: See if earlier message was adjacency reject (Event 4.16).

DN.033

Level: C–INFO

Short Syntax: DN.033 event 4.18: Adj dwn; node *area.node*, cir *number* net *network_name* timed out

Long Syntax: DN.033 event 4.18: Adjacency down; node *area.node*, circuit *number* network *network_name* timed out

Description: The specified adjacency has gone down because a Router Hello Message has not been heard from the adjacency for three times the hello time presented in the adjacency's Router Hello Message (the adjacency's CIRCUIT HELLO TIMER). All routes through this adjacency will be deleted.

Cause: Node down.

Action: Check status of node.

Cause: Node disconnected from network.

Action: Check circuit and line status on node.

DN.034

Level: UE–ERROR

Short Syntax: DN.034 event 4.18: Adj dwn; lvl 1 route from *area.node*, cir *number* net *network_name*, *cksum received_checksum*, expct *correct_checksum*

Long Syntax: DN.034 event 4.18: Adjacency down; lvl 1 route from *area.node*, circuit *number* network *network_name*, checksum *received_checksum*, expected *correct_checksum*

Description: A Level 1 Routing Message was received with an invalid checksum. The packet will be dropped, and the adjacency with the router will be taken down.

Cause: Data corruption error.

Action: Check network error counters.

Cause: Programming error at remote node.

Action: See if error is consistent from a particular node.

DN.035

Level: UE–ERROR

Short Syntax: DN.035 event 4.18: Adj dwn; lvl 2 route from *area.node*, cir *number* net *network_name*, *cksum received_checksum*, expct *correct_checksum*

Long Syntax: DN.035 event 4.18: Adjacency down; lvl 2 route from *area.node*, circuit *number* network *network_name*, checksum *received_checksum*, expected *correct_checksum*

Description: A Level 2 Routing Message was received with an invalid checksum. The packet will be dropped, and the adjacency with the router will be taken down.

Cause: Data corruption error.

Action: Check network error counters.

Cause: Programming error at remote node.

Action: See if error is consistent from a particular node.

DN.036

Level: C–INFO

Short Syntax: DN.036 event 4.19: Adj dwn: dropped by rtr *area.node*, cir *number* net *network_name*

Long Syntax: DN.036 event 4.19: Adjacency down, operator initiated:dropped by router *area.node*, circuit *number* network *network_name*

Description: A Router Hello Message has been received from a router that we have an adjacency with, but does not include our address in the router state list. The adjacency will be taken down,and will not come back up until our address is in the router state list.

Cause: Adjacent router restarted.

Cause: One–way communication. While this router can receive packets from the adjacent router, the adjacent router cannot receive packets from this router.

Action: Ensure that there is two–way communication on the circuit.

DN.037

Level: U–INFO

Short Syntax: DN.037 event 5.0: Circ dwn; cir *number* net *network_name*

Long Syntax: DN.037 event 5.0: Circuit down; cir *number* net *network_name*

Description: A circuit has gone down. All adjacencies through this circuit will be taken down.

Cause: self-test failure.

Action: Look for self-test error messages, check status of interface.

Cause: Disabling circuit through CGWCON, by the SET CIRCUIT STATE OFF command, or by the SET EXECUTOR STATE OFF command.

DN.038

Level: U–INFO

Short Syntax: DN.038 event 5.0: Circ up; cir *number* net *network_name*

Long Syntax: DN.038 event 5.0: Circuit up; cir *number* net *network_name*

Description: A circuit has gone up, due either to enabling the circuit through CGWCON, due to a self-test success, by the NCP SET CIRCUIT STATE ON command, or by the NCP SET EXECUTOR STATE ON command. The router will start sending router hellos on the circuit.

DN.039

Level: UI–ERROR

Short Syntax: DN.039 event 5.14: Send fail; rsn *reason_code*, *source* –>destination cir *number* net *network_name*

Long Syntax: DN.039 event 5.14: Send failure on line; reason *reason_code*, packet from *source* to *destination* cir *number* net *network_name*

Description: The sending of a packet being forwarded failed. The *reason_code* is the internal error code for the failure.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network_name.*

Cause: Output queue overflow, or other flow control. (Reason code2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast.(Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

DN.040

Level: P–TRACE

Short Syntax: DN.040 *source –> destination*

Long Syntax: DN.040 Forwarding packet from *source* to *destination*

Description: Forwarding a packet from one node to another.

DN.041

Level: P–TRACE

Short Syntax: DN.041 MOP Req ID pkt rcvd frm *MAC_address* cir *number* net *network_name*

Long Syntax: DN.041 MOP Request ID packet received from node *MAC_address* circuit *number* network *network_name*

Description: A DECnet Maintenance Operations Protocol MOP Request System ID packet was received from the specified node. A MOP System ID packet will be sent to the requester’s address.

DN.042

Level: P–TRACE

Short Syntax: DN.042 MOP Sys ID pkt rcvd frm *MAC_address* cir *number* net *network_name*

Long Syntax: DN.042 MOP System ID packet received from node *MAC_address* circuit *number* network *network_name*

Description: A DECnet MOP System ID packet was received from the specified node.

DN.043

Level: UE–ERROR

Short Syntax: DN.043 MOP pkt rcvd unk opc *opcode* frm *MAC_address* cir *number* net *network_name*

Long Syntax: DN.043 MOP packet received unknown opcode *opcode* from node *MAC_address* cir *number* net *network_name*

Description: DECnet MOP (Maintenance Operations Protocol) packet received with unsupported opcode from specified node. The packet will be ignored.

Cause: Programming error on remote note.

Cause: Data corruption.

DN.044

Deleted: Message deleted.

DN.045

Level: UI–ERROR

Short Syntax: DN.045 acc cnt bad rec, cir *number* net *network_name*, purge

Long Syntax: DN.045 Access control bad SRAM record, circuit *number* network *network_name*, purge

Description: There is a faulty access control record in the permanent database for this circuit.

Action: Do a PURGE MODULE ACCESS CONTROL CIRCUIT.

DN.046

Level: C–INFO

Short Syntax: DN.046 acc cont fail *source –> destination* cir *number* net *network_name*

Long Syntax: DN.046 Access control failed, packet from *source* to *destination* circuit *number* network *network_name*

Description: A packet was not forwarded between the two hosts due to access control restrictions. If Request Return to Sender was set in the header, the packet will be returned to the sender, otherwise it will be dropped.

Cause: User attempting to contact host is restricted by access control.

DN.047

Level: C–INFO

Short Syntax: DN.047 desig router chng frm *old_router* to *new_router,* cir *number* net *network_name*

Long Syntax: DN.047 Designated router changed from *old_router* to *new_router,* circuit *number* network *network_name*

Description: Designated router for this circuit has changed.

Cause: New router adjacency with higher router priority on circuit,or same router priority and higher node address.

DN.048

Level: C–INFO

Short Syntax: DN.048 desig router *address* select, cir *number* net *network_name*

Long Syntax: DN.048 Designated router *address* selected, circuit *number* network *network_name*

Description: There is now a designated router for this circuit, where there had not been one before.

DN.049

Level: P–TRACE

Short Syntax: DN.049 endnode hello len *packet_length* from node, cir *number* net *network_name*

Long Syntax: DN.049 endnode hello length *packet_length* from node, circuit *number* network *network_name*

Description: Received endnode hello message from specified endnode.

DN.050

Level: ALWAYS

Short Syntax: DN.050 executor node address *area.node* exceeds MAX ADDRESS *max_address*

Long Syntax: DN.050 executor node address *area.node* exceeds EXECUTOR MA*address* *max_address*

Description: The EXECUTOR ADDRESS stored in the permanent database exceeds the EXECUTOR MAXIMUM ADDRESS stored in the permanent database. DECnet will be left off, but the database will be allocated.

Action: Either correct EXECUTOR ADDRESS or EXECUTOR MAX ADDRESS.

DN.051

Level: ALWAYS

Short Syntax: DN.051 executor node address *area.node* exceeds MAX AREA *max_node*

Long Syntax: DN.051 executor node address *area.node* exceeds EXECUTOR MAXAREA *max_node*

Description: The area of the EXECUTOR ADDRESS stored in the permanent database exceeds the EXECUTOR MAXIMUM AREA stored in the permanent database. DECnet will be left off, but the database will be allocated.

Action: Either correct EXECUTOR ADDRESS or EXECUTOR MAX AREA.

DN.052

Deleted: Message deleted at Release 9.0.

DN.053

Level: CI–ERROR

*Short Syntax***:** DN.053 inp que ovflow data *source –> destination* cir *number* net *network_name*

*Long Syntax***:** DN.053 Input queue overflow data packet from *source* to *destination* circuit *number* network *network_name*

Description: The DECnet input queue overflowed for incoming Long Format Data packet. The packet will be dropped.

Cause: Too much traffic for forwarder to forward.

Action: Adjust circuit costs to balance traffic between paths. Re–configure network. Increase speed of router.

Cause: Inadequate buffer resources.

Action: Examine memory statistics in GWCON. More buffers can be made available by ensuring that DECnet configuration does not have excess adjacency memory allocated.

Action: Increase memory.

DN.054

Level: CI–ERROR

*Short Syntax***:** DN.054 inp que ovflow multicast from *source* cir *number* net *network_name*

*Long Syntax***:** DN.054 Input queue overflow multicast from *source* circuit *number* network *network_name*

Description: The DECnet input queue overflowed for incoming routing or hello multicast packet. The packet will be dropped.

Cause: Too much traffic for forwarder to forward.

Action: Adjust circuit costs to balance traffic between paths. Re–configure network. Increase speed of router.

Cause: Inadequate buffer resources.

Action: Examine memory statistics in GWCON. More buffers can be made available by ensuring that DECnet configuration does not have excess adjacency memory allocated.

Action: Increase memory.

DN.055

Level: U–TRACE

Short Syntax: DN.055 lvl 1 rte pkt from *source* ign, cir *number* net *network_name*, no adjacency

Long Syntax: DN.055 Level 1 routing message from *source* ignored, circuit *number* network *network_name*, no adjacency with router

Description: A Level 1 Routing Message was received from a router that does not have an active adjacency with this router. The routing packet will not be processed.

Cause: This will happen occasionally when the other router develops an adjacency with this router before this one does.

Action: No action needed unless message is persistent.

DN.056

Level: P–TRACE

Short Syntax: DN.056 lvl 1 rte pkt len *received_length* from *source*, cir *number* net *network_name*

Long Syntax: DN.056 Level 1 routing packet length *received_length* from *source*, circuit *number* network *network_name*

Description: A Level 1 Routing Message was received from the specified router.

DN.057

Level: U–TRACE

Short Syntax: DN.057 lvl 2 rte pkt from *source* ign, cir *number* net *network_name*, no adjacency

Long Syntax: DN.057 Level 2 routing message from *source* ignored, circuit *number* network *network_name*, no adjacency with router

Description: A Level 2 Routing Message was received from a router that does not have an active adjacency with this router. The routing packet will not be processed.

Cause: This will happen occasionally when the other router develops an adjacency with this router before this one does.

Action: No action needed unless message is persistent.

Cause: Level 2 routing message sent by level 1 router.

Action: Correct software error at sending router.

DN.058

Level: P–TRACE

Short Syntax: DN.058 lvl 2 rte pkt len *received_length* from *source*, cir *number* net *network_name*

Long Syntax: DN.058 Level 2 routing packet length *received_length* from *source*, circuit *number* network *network_name*

Description: A Level 2 Routing Message was received from the specified router.

DN.059

Level: UI–ERROR

Short Syntax: DN.059 no buffer for hello on cir *number* net *network_name*

Long Syntax: DN.059 No buffer to build hello packet to send on circuit *number* network *network_name*

Description: No packet buffer was available to construct and send a Router Hello Message.

Cause: Severe packet buffer shortage.

Action: Check memory statistics in GWCON to verify packet buffer level. If possible, make routing tables smaller. In DEC–net, this is done by minimizing the number of adjacencies allowed. (Configure EXECUTOR MAXIMUM BROADCAST ROUTERS and EXECUTOR MAXIMUM BROADCAST NONROUTERS to minimum appropriate values.) If routing tables cannot be made smaller, increase memory size.

Cause: Traffic peak using all available buffers.

Action: This is the problem if this message occurs very infrequently.

DN.060

Level: UI–ERROR

*Short Syntax***:** DN.060 no buffer for lvl 1 rte on cir *number* net *network_name*

*Long Syntax***:** DN.060 No buffer to build level 1 routing message to send on circuit *number* network *network_name*

Description: No packet buffer was available to construct and send a Level1 Routing Message.

Cause: Severe packet buffer shortage.

Action: Check memory statistics in GWCON to verify packet buffer level. If possible, make routing tables smaller. In DEC–net, this is done by minimizing the number of adjacencies allowed. (Configure EXECUTOR MAXIMUM BROADCAST ROUTERS and EXECUTOR MAXIMUM BROADCAST NONROUTERS to minimum appropriate values.) If routing tables cannot be made smaller, increase memory size.

Cause: Traffic peak using all available buffers.

Action: This is the problem if this message occurs very infrequently.

DN.061

Level: UI–ERROR

*Short Syntax***:** DN.061 no buffer for lvl 2 rte on cir *number* net *network_name*

*Long Syntax***:** DN.061 No buffer to build level 2 routing message to send on circuit *number* network *network_name*

Description: No packet buffer was available to construct and send a Level2 Routing Message.

Cause: Severe packet buffer shortage.

Action: Check memory statistics in GWCON to verify packet buffer level. If possible, make routing tables smaller. In DEC–net, this is done by minimizing the number of adjacencies allowed. (Configure EXECUTOR MAXIMUM BROADCAST ROUTERS and EXECUTOR MAXIMUM BROADCAST NONROUTERS to minimum appropriate values.) If routing tables cannot be made smaller, increase memory size.

Cause: Traffic peak using all available buffers.

Action: This is the problem if this message occurs very infrequently.

DN.062

Level: ALWAYS

Short Syntax: DN.062 no memory for NCP circuit name table

Long Syntax: DN.062 No memory for building NCP circuit name table

Description: No memory was available to build the circuit name table for NCP at start time.

Cause: There is some configuration error causing a grave memory shortage.

Action: Reduce memory demand by making routing tables smaller, or getting more memory.

DN.063

Level: ALWAYS

Short Syntax: DN.063 no memory for routing tables (*number* bytes req), DEC–net disabled

Long Syntax: DN.063 No Memory for building routing tables (*number* bytes required), DECnet disabled

Description: The routing tables required more memory than was available. DECnet is disabled.

Cause: Parameters that determine size of routing database are too large for actual network configuration.

Action: The following parameters should be reduced as appropriate using the DEFINE commands, and the gateway restarted: EXECUTOR MAXIMUM BROADCAST ROUTERS, EXECUTOR MAXIMUM BROADCASTN ON ROUTERS, CIRCUIT MAXIMUM ROUTERS, EXECUTOR MAXIMUM ADDRESS, EXECUTOR MAXIMUM AREA,

Cause: Inadequate memory size.

Action: Upgrade for more memory.

DN.064

Level: CI–ERROR

Short Syntax: DN.064 packet received on down cir *number* net *network_name*,dropped

Long Syntax: DN.064 Packet received on down circuit *number* network *network_name*, packet dropped

Description: Received a data packet on a circuit or router that does not have DECnet enabled. The packet will be dropped.

DN.065

Level: P–TRACE

Short Syntax: DN.065 DECnet pkt ign

Long Syntax: DN.065 DECnet packet ignored, no DECnet forwarder

Description: A DECnet packet was received, but no DECnet forwarder is in–stalled in the gateway.

DN.066

Level: U–TRACE

Short Syntax: DN.066 returning packet to sender *sender* <– *original_destination*

Short Syntax: DN.066 returning packet to sender *sender* from *original_destination*

Description: A data packet could not reach the destination, and had the Request Return to Sender bit set in the header. It is being returned to the sender.

Cause: Should be explained by a previous message, such as Events4.1, 4.2, and 4.3.

Action: See action in causative message.

DN.067

Level: P–TRACE

Short Syntax: DN.067 routerhello len *received_length* from *source*, cir *number* net *network_name*

Long Syntax: DN.067 Router hello length *received_length* received from *source*, circuit *number* network *network_name*

Description: A Router Hello Message was received from the specified router.

DN.068

Level: P–TRACE

Short Syntax: DN.068 sending desig rtr hello on cir *number* net *network_name*

Long Syntax: DN.068 Sending designated router hello on circuit *number* network *network_name*

Description: A Router Hello Message is being sent to the ALLENDNODES address, as this router is the designated router on the specified circuit.

DN.069

Level: P–TRACE

Short Syntax: DN.069 sending hello on cir *number* net *network_name*

Long Syntax: DN.069 Sending router hello on circuit *number* network *network_name*

Description: A Router Hello Message is being sent to the ALLROUTERS address on the specified circuit.

DN.070

Level: P–TRACE

Short Syntax: DN.070 sending lvl 1 rte on cir *number* net *network_name*

Long Syntax: DN.070 Sending level 1 routing message on circuit *number* network *network_name*

Description: A Level 1 Routing Message is being sent to the ALLROUTERS address on the specified circuit.

DN.071

Level: P–TRACE

Short Syntax: DN.071 sending lvl 2 rte on cir *number* net *network_name*

Long Syntax: DN.071 Sending level 2 routing message on circuit *number* network *network_name*

Description: A Level 2 Routing Message is being sent to the ALLROUTERS address on the specified circuit.

DN.072

Level: ALWAYS

Short Syntax: DN.072 too many router adjacencies *total_adjacencies*, NBRA = *maximum_adjacencies*

Long Syntax: DN.072 Too many router adjacencies configured, sum = *total_adjacencies*, NBRA = *maximum_adjacencies*

Description: The permanent database has been configured such that the sum of CIRCUIT MAXIMUM ROUTERS for all circuits exceeds EXECUTORMAXIMUM BROADCAST ROUTERS. This error is non-fatal, but new values should be DEFINED, and the gateway restarted.

Cause: CIRCUIT MAXIMUM ROUTERS too large.

Action: This is the usual problem, especially on Serial Line interfaces, where there can only be one router adjacency.

Cause: EXECUTOR MAXIMUM BROADCAST ROUTERS too small.

Action: This is not normally the problem, as the default is 32,which is quite generous.

DN.073

Level: C–INFO

Short Syntax: DN.073 new 1–way adj *sender* cir *number* net *network_name*

Long Syntax: DN.073 new 1–way adjacency with node *sender* on circuit *number* network *network_name*

Description: We have just received a router hello message from the specified router, but our address is not in the router/state list of the hello message. We have a one–way adjacency with this router, it will not be two–way until our address is in the router/state list.

Cause: New node came up.

Action: None required unless adjacency never reaches two way. This should happen shortly. If it does not, it may indicate that our address is beyond the other routers EXECUTOR MAXIMUM AD–DRESS.

DN.074

Level: C–INFO

Short Syntax: DN.074 1–way adj *sender* timed out cir *number* net *network_name*

Long Syntax: DN.074 1–way adjacency with node *sender* timed out on circuit *number* network *network_name*

Description: We have stopped receiving router hellos without our node ad–dress in the router/state list from the specified router. The time out is three times the hello timer that was specified in the last router hello from this router. The partial adjacency with this router will be eliminated.

Cause: New node never came up all the way.

DN.075

Level: P–TRACE

Short Syntax: DN.075 Pkt for me frm *sender*

Long Syntax: DN.075 Packet for me from node *sender*

Description: We have received a packet addressed to us. It will be checked to see what transport protocol it is for.

DN.076

Level: U–TRACE

Short Syntax: DN.076 NSP unsupp msg type *msgflg* frm *sender*

Long Syntax: DN.076 NSP unsupported message type *msgflg* from node *sender*

Description: We have received an NSP packet of a message type that we do not process. Only Connect Initiate Messages are processed.

DN.077

Level: CE–ERROR

Short Syntax: DN.077 Unk trans type *msgflg* from *sender*

Long Syntax: DN.077 Unknown transport protocol type *msgflg* from node *sender*

Description: We have received a data packet that is not for the NSP transport protocol.

DN.078

Level: C–INFO

Short Syntax: DN.078 NSP conn init from *sender*, reject

Long Syntax: DN.078 NSP Connect Initiate Message received from node *sender*, rejecting

Description: An NSP Connect Initiate or Retransmitted Connect Initiate Message was received from the specified node. A Disconnect Initiate message will be sent in return, with a Session Reject error code of 4 (destination end user does not exist).

Cause: User on remote machine attempted to initiate an NSP connection, but there are no Session clients supported in the router.

DN.079

Level: UE–ERROR

Short Syntax: DN.079 endnode hello from *sender* cir *number* net *network_name*d up addr w/self, ign

Long Syntax: DN.079 endnode hello from node *sender* circuit *number* network *network_name*, duplicate address with self, ignoring

Description: An Endnode Hello Message was received from a node with the same DECnet address as this router. Since duplicate node addresses are not allowed, and the router is more important,the hello message will be ignored.

Cause: User configuration error.

Action: Change DECnet node address.

DN.080

Level: P–TRACE

Short Syntax: DN.080 MOP Req Cnt pkt rcvd frm *MAC_address* cir *number* net *network_name*

Long Syntax: DN.080 MOP Request Counters packet received from node *MAC_address* circuit *number* network *network_name*

Description: A DECnet Maintenance Operations Protocol (MOP) Request Counters packet was received from the specified node. A MOP Counters packet will be sent to the requester’s address.

DN.081

Level: P–TRACE

Short Syntax: DN.081 MOP Cnt pkt snt to *MAC_address* cir *number* net *network_name*

Long Syntax: DN.081 MOP Counters packet sent to node *MAC_address* circuit *number* network *network_name*

Description: A DECnet Maintenance Operations Protocol (MOP) Counters packet is being sent to the specified address.

DN.082

Level: P–TRACE

Short Syntax: DN.082 MOP Sys ID pkt snt to *MAC_address* cir *number* net *network_name*

Long Syntax: DN.082 MOP System ID packet sent to node *MAC_address* circuit *number* network *network_name*

Description: A DECnet Maintenance Operations System ID packet is being sent to the specified address.

DN.083

Level: P–TRACE

Short Syntax: DN.083 MOP Sys ID pkt snt to MOP cir *number* net *network_name*

Long Syntax: DN.083 MOP System ID packet sent to MOP circuit *number* network *network_name*

Description: A DECnet Maintenance Operations Protocol System ID packet is being sent to the MOP multicast address AB–00–00–02–00–00.

DN.084

Level: UI–ERROR

Short Syntax: DN.084 MOP Cnt Req frm *MAC_address* not supp on cir *number* net *network_name*

Long Syntax: DN.084 MOP Cnt Req from node *MAC_address* not supported on circuit *number* network *network_name*

Description: A DECnet Maintenance Operations Protocol (MOP) Request Counters was received from the specified host, but there is no support for the MOP Counters message on this circuit.

DN.085

Level: UI–ERROR

Short Syntax: DN.085 Ph IV rtr hlo wo bilingual rtr frm *node_number* on cir *number* net *network_name*

Long Syntax: DN.085 Ph IV router hello without bilingual router from *node_number* on circuit *number* network *network_name*

Description: A DECnet Phase IV broadcast router hello was received on a circuit that was configured for Phase IV' only.

Cause: Router is receiving Phase IV broadcast router hello packets on a network that should only have Phase IV' packets

Action: Router must be configured for both Phase IV and Phase IV' to receive the broadcast router hello packets from a Phase IV router.

DN.086

Level: UI–ERROR

Short Syntax: DN.086 Ph IV ennd hlo wo bilingual rtr frm *node_number* oncir *circut_number* net *node_name*

Long Syntax: DN.086 Ph IV endnode hello without bilingual router from *node_number* on circuit *circut_number* network *node_name*

Description: A DECnet Phase IV broadcast endnode hello was received on a circuit that was configured for Phase IV' only.

Cause: The router is receiving Phase IV broadcast endnode hello packets on a network that should only have Phase IV' packets.

Action: The router must be configured for both Phase IV and Phase IV to receive the broadcast endnode hello packets from a Phase IV end node.

DN.087

Level: UI–ERROR

Short Syntax: DN.087 Ph IV' rtr hlo wo bilingual or ama rtr frm *node_number* on cir *circut_number* net *node_name*

Long Syntax: DN.087 Ph IV' router hello without bilingual or ama router from *node_number* on circuit *circut_number* network *node_name*

Description: A DECnet Phase IV' broadcast router hello was received on a circuit that was configured for Phase IV only.

Cause: The router is receiving Phase IV' broadcast router hello packets on a network that should only have Phase IV packets.

Action: The router must be configured for Phase IV' to receive the broadcast endnode hello packets from a Phase IV' endnode.

DN.088

Level: UI–ERROR

*Short Syntax***:** DN.088 Ph IV’ ennd hlo wo bilingual or ama rtr frm *node_number* on cir *circut_number* net *node_name*

*Long Syntax***:** DN.088 Ph IV’ endnode hello without bilingual or ama router from *node_number* on circuit *circut_number* network *node_name*

Description: A DECnet Phase IV’ broadcast endnode hello was received on a circuit that was configured for Phase IV only.

Cause: The router is receiving Phase IV’ broadcast endnode hello packets on a network that should only have Phase IV packets.

Action: The router must be configured for Phase IV’ to receive the broadcast endnode hello packets from a Phase IV’ endnode.

DN.089

Level: UI–ERROR

*Short Syntax***:** DN.089 Unkn ennd hlo format frm *node_number* on cir *circut_number* net *node_name*

*Long Syntax***:** DN.089 Unknown endnode hello message format from *node_number* on circuit *circut_number* network *node_name*

Description: The router received an Endnode Hello Message with unknown format.

Cause: Some station is sending a message with this format.

Action: Determine the errant node from this message and inform the manufacturer that this node is sending hello messages of unknown format.

DN.090

Level: UI–ERROR

*Short Syntax***:** DN.090 Cannot bld lvl 1 rte on cir *number* net *network_name*,blk sz too small – *block_size*

*Long Syntax***:** DN.090 Cannot build level 1 routing message on circuit *number*, network *network_name*, block size too small –*block_size*

Description: A Level 1 Routing Message cannot be built because the circuit’s minimum block size is too small.

DN.091

Level:	UI–ERROR
Short Syntax:	DN.091 Send fail for hello, rsn *reason_code*, cir *number* net *network_name*
Long Syntax:	DN.091 Send failed for router hello packet, reason *reason_code*, on circuit *number* network *network_name*
Description:	The transmission of a router hello packet failed on the specified circuit for the reason number given in *reason_code*. Occasional occurrences of this will not disrupt the protocol, but continuing occurrences will disrupt the protocol.
Cause:	Miscellaneous handler error. (Reason code 1.)
Action:	Check for error messages from handler for *network_name.*
Cause:	Output queue overflow, or other flow control. (Reason code2.)
Action:	Alleviate congestion.
Cause:	Network down. (Reason code 3.)
Action:	See why handler thinks network is down.
Cause:	Dropped by handler to avoid looping, or bad broadcast.(Reason code 4.)
Action:	Check configuration.
Cause:	Host down. (Reason code 5.)
Action:	See why handler thinks host is down.

DN.092

Level:	UI–ERROR
Short Syntax:	DN.092 Send fail for lvl 1 rte, rsn *reason_code*, cir *number* net *network_name*
Long Syntax:	DN.092 Send failed for level 1 routing message, reason *reason_code*, on circuit *number* network *network_name*
Description:	The transmission of a Level 1 Routing Message failed on the specified circuit for the reason number given in *reason_code*. Occasional occurrences of this will not disrupt the protocol, but continuing occurrences will disrupt the protocol.
Cause:	Miscellaneous handler error. (Reason code 1.)
Action:	Check for error messages from handler for *network_name.*
Cause:	Output queue overflow, or other flow control. (Reason code2.)
Action:	Alleviate congestion.
Cause:	Network down. (Reason code 3.)
Action:	See why handler thinks network is down.
Cause:	Dropped by handler to avoid looping, or bad broadcast.(Reason code 4.)
Action:	Check configuration.
Cause:	Host down. (Reason code 5.)
Action:	See why handler thinks host is down.

DN.093

Level:	UI–ERROR
Short Syntax:	DN.093 Send fail for lvl 2 rte, rsn *reason_code*, cir *number* net *network_name*
Long Syntax:	DN.093 Send failed for level 2 routing message, reason *reason_code*, on circuit *number* network *network_name*
Description:	The transmission of a Level 2 Routing Message failed on the specified circuit for the reason number given in *reason_code*. Occasional occurrences of this will not disrupt the protocol, but continuing occurrences will disrupt the protocol.
Cause:	Miscellaneous handler error. (Reason code 1.)
Action:	Check for error messages from handler for *network_name.*
Cause:	Output queue overflow, or other flow control. (Reason code2.)
Action:	Alleviate congestion.
Cause:	Network down. (Reason code 3.)
Action:	See why handler thinks network is down.
Cause:	Dropped by handler to avoid looping, or bad broadcast.(Reason code 4.)
Action:	Check configuration.
Cause:	Host down. (Reason code 5.)
Action:	See why handler thinks host is down.

DN.094

Level: UI–ERROR

Short Syntax: DN.094 Send fail for MOP *message_type*, rsn *reason_code*, cir *number* net *network_name*

Long Syntax: DN.094 Send failed for MOP *message_type* message, reason *reason_code*, on circuit *number* network *network_name*

Description: The transmission of a MOP message failed on the specified circuit for the reason number given in *reason_code*. The *message_type* is one of ‘‘System ID’’ or ‘‘Counters.’’ Occasional occurrences of this will not disrupt the protocol,but continuing occurrences will disrupt the protocol.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network_name*.

Cause: Output queue overflow, or other flow control. (Reason code2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast.(Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

Check “dnrouttype”

Short Syntax: Unknown circuit router type.

Description: The circuit router type is unknown.

Cause: Data corruption, probably from coding error.

Check “dnrfgtl”

Short Syntax: DN routes() called with first > last

Description: The dnroute routine was called with an invalid node address range.

Cause: Internal consistency error.

Action: Report to customer service, preferably with a core dump.

Check “dnrbeaf”

Short Syntax: DN routes() BEA optimization failed

Description: The dnroute routine has computed a route through a broadcast circuit, rather than through a router or endnode.

Cause: Internal consistency error.

Action: Report to customer service, preferably with a core dump.

Check “dnarfgtl”

Short Syntax: DNA routes() called with first > last

Description: The DNA route routine was called with an invalid area range.

Cause: Internal consistency error.

Action: Report to customer service, preferably with a core dump.

Check “dnmembug”

Short Syntax: DN no memory for table

Description: An allocation of memory for the routing tables failed, but acheck of free memory had indicated that there should be enough memory.

Cause: Internal consistency check.

Action: Report to customer service, preferably with a core dump.

Fatal “dnadbadarg”

Short Syntax: DN bad arg to dnadjdown()

Description: The dnadjdown routine was asked to remove an adjacency that was not a router or an endnode.

Cause: Internal consistency error.

Action: Report to customer service, preferably with a core dump.

Fatal “dnacnmr”

Short Syntax: DN no mem to read acc cntl

Description: There is no memory available to read the access control lists from the permanent database.

Cause: Severe memory shortage.

Action: Reduce sizes of routing tables to use less memory, or add additional memory.

Fatal “dnacnmsac”

Short Syntax: DN no mem to store acc cntl

Description: There is no memory available to store the access control lists for use.

Cause: Severe memory shortage.

Action: Reduce sizes of routing tables to use less memory, or add additional memory.

Fatal “dnacnmcac”

Short Syntax: DN no mem for acc cntl

Description: There is no memory available to build the access control list.

Cause: Severe memory shortage.

Action: Reduce sizes of routing tables to use less memory, or add additional memory.

Fatal “dncnmrfi”

Short Syntax: DN no mem for dnrfin

Description: There is no memory available to build the circuit input routing filter table.

Cause: Severe memory shortage.

Action: Reduce sizes of routing tables to use less memory, or add additional memory.

Fatal “dncnmrfo”

Short Syntax: DN no mem for dnrfout

Description: There is no memory available to build the circuit output routing filter table.

Cause: Severe memory shortage.

Action: Reduce sizes of routing tables to use less memory, or add additional memory.

Fatal “dncnmci”

Short Syntax: DN no mem for dnccti init

Description: There is no memory available to build the circuit volatile database.

Cause: Severe memory shortage.

Action: Reduce sizes of routing tables to use less memory, or add additional memory.

Panic “dnrtcrtos”

Short Syntax: DN routing table corrupt: routes to self

Description: The routing database consistency checker has detected an in–consistency in the routing database. The router will be restarted.

Cause: Memory corruption.

Action: Configure for core dump, and report to customer service.

Cause: Internal software error.

Action: Configure for core dump, and report to customer service.

Panic “dnrtcarts”

Short Syntax: DN routing table corrupt: area routes to self

Description: The routing database consistency checker has detected an in–consistency in the routing database. The router will be restarted.

Cause: Memory corruption.

Action: Configure for core dump, and report to customer service.

Cause: Internal software error.

Action: Configure for core dump, and report to customer service.

Panic “dnrtcrths”

Short Syntax: DN routing table corrupt: routes through self

Description: The routing database consistency checker has detected an in–consistency in the routing database. The router will be restarted.

Cause: Memory corruption.

Action: Configure for core dump, and report to customer service.

Cause: Internal software error.

Action: Configure for core dump, and report to customer service.

Panic “dnrtcrtas”

Short Syntax: DN routing table corrupt: route to area self

Description: The routing database consistency checker has detected an in–consistency in the routing database. The router will be restarted.

Cause: Memory corruption.

Action: Configure for core dump, and report to customer service.

Cause: Internal software error.

Action: Configure for core dump, and report to customer service.

Panic “dnrtcartas”

Short Syntax: DN routing table corrupt: area route to area self

Description: The routing database consistency checker has detected an in–consistency in the routing database. The router will be restarted.

Cause: Memory corruption.

Action: Configure for core dump, and report to customer service.

Cause: Internal software error.

Action: Configure for core dump, and report to customer service.

DNAV

Digital Network Architecture Phase V

This chapter describes Digital™ Network Architecture (DNA) Phase V messages. These messages come from the software that provides the convergence layer between the OSI–CLNP forwarder and the DNA Phase IV forwarder to provide the DNA Phase V extensions to the CLNP protocol. For information about message content and how to use the message, refer to the Introduction.

DNAV.001

Level: P–TRACE

Short Syntax: DNAV.001 DNA pkt forwarded through OSI at level *rtg_lvl*

Long Syntax: DNAV.001 DNA packet forwarded through OSI at level *rtg_lvl*

Description: A DNA packet was received and then passed to OSI for for–warding.

DNAV.002

Level: P–TRACE

Short Syntax: DNAV.002 DNA pkt translated to OSI pkt *source_NSAP–> destination_NSAP*

Long Syntax: DNAV.002 DNA pkt translated to OSI pkt: *source_NSAP –>destination_NSAP*

Description: A DNA data packet was successfully translated to an OSI data packet.

DNAV.003

Level: P–TRACE

Short Syntax: DNAV.003 Translation of DNA pkt to OSI pkt failed

Long Syntax: DNAV.003 Translation of DNA pkt to OSI pkt failed

Description: An attempt to translate a DNA data packet to an OSI data packet failed.

DNAV.004

Level: P–TRACE

Short Syntax: DNAV.004 OSI pkt translated to DNA pkt *src –> dst*

Long Syntax: DNAV.004 OSI pkt translated to DNA pkt: *src –> dst*

Description: An OSI data packet was successfully translated to a DNA data packet.

DNAV.005

Level: P–TRACE

Short Syntax: DNAV.005 Translation of OSI pkt to DNA pkt failed

Long Syntax: DNAV.005 Translation of OSI pkt to DNA pkt failed

Description: An attempt to translate an OSI data packet to a DNA data packet failed.

DNAV.006

Level: P–TRACE

Short Syntax: DNAV.006 OSI pkt forwarded through DNA at level *rtg_lvl*

Long Syntax: DNAV.006 OSI packet forwarded through DNA at level *rtg_lvl*

Description: An OSI packet was received and then passed to DNA for for–warding.

DNAV.007

Level: UE–ERROR

Short Syntax: DNAV.007 timed out route to DNA IV ES reactivated *src_area.src_node*

Long Syntax: DNAV.007 timed out route to DNA IV ES reactivated *src_area.src_node*

Description: A DNA endnode hello packet was received with a route that had been previously timed out in the OSI database.

DNAV.008

Level: P–TRACE

Short Syntax: DNAV.008 ISIS hello from distance vector router funnelled to DNA

Long Syntax: DNAV.008 ISIS hello from distance vector router funnelled to DNA

Description: An ISIS hello was received from a router running distance vector – the hello was passed to DNA IV to establish a router adjacency.

DNAV.009

Level: C–INFO

Short Syntax: DNAV.009 new 1–way adj w/ phase V dist vect router *sender* cir *number* net *network_name*

Long Syntax: DNAV.009 new 1–way adjacency with phase V distance vector router *sender* on circuit *number* network *network_name*

Description: We have just received an ISIS Hello Message from the specified router, but our address is not in the IS neighbor list of the hello message. We have a one–way adjacency with this router, it will not be two–way until our address is in the IS neighbor list.

DNAV.010

Level: C–INFO

Short Syntax: DNAV.010 Adj up; new phase V dist vect rtr *area.node* cir *number* net *network_name*

Long Syntax: DNAV.010 Adjacency up; new phase V distance vector router *area.node* circuit *number* network *network_name*

Description: There is now an adjacency with the specified router on one of the directly connected networks. Level 1 (and 2) Routing Messages will now be accepted from this node.

DNAV.011

Level: C–INFO

Short Syntax: DNAV.011 Adj dwn: dropped by phase V dist vect rtr*area.node*, cir *number* net *network_name*

Long Syntax: DNAV.011 Adjacency down, operator initiated: dropped by phase V distance vector router *area.node*, circuit *number* network *network_name*

Description: An ISIS Hello Message has been received from a router that we have an adjacency with, but does not include our address in the IS neighbor list. The adjacency will be taken down,and will not come back up until our address is in the IS neighbor list.

Cause: Adjacent router restarted.

Cause: One–way communication. While this router can receive packets from the adjacent router, the adjacent router cannot receive packets from this router.

Action: Ensure that there is two–way communication on the circuit.

DNAV.012

Level: UE–ERROR

Short Syntax: DNAV.012 pkt trans V to IV err – segmentation needed but not permitted

Long Syntax: DNAV.012 packet translation V to IV error – segmentation needed but not permitted

Description: An OSI data packet could not be translated to a DNA IV data packet because it needs to be segmented – segmentation of it is not permitted.

DNAV.013

Level: UE–ERROR

Short Syntax: DNAV.013 pkt trans V to IV err – src or dst addr not translatable

Long Syntax: DNAV.013 packet translation V to IV error – source or destination address not translatable

Description: An OSI data packet could not be translated to a DNA IV data packet because either the source or destination address is not Phase IV translatable.

DNAV.014

Level: UE–ERROR

Short Syntax: DNAV.014 Validation of phase IV info in ISIS hello PDU failed

Long Syntax: DNAV.014 Validation of phase IV info in ISIS hello PDU failed

Description: An ISIS hello PDU was received with an invalid Phase IV in–formation option.

DNAV.015

Level: UE–ERROR

Short Syntax: DNAV.015 Phase IV hello from Phase V system dropped

Long Syntax: DNAV.015 Phase IV hello from Phase V system dropped

Description: A Phase IV hello PDU is dropped because it was sent by a Phase V system – adjacencies with Phase V systems are established using Phase V hellos.

DNAV.016

Level: UE–ERROR

Short Syntax: DNAV.016 L1 LSP from DNA system dropped – running dist vect at level 1

Long Syntax: DNAV.016 L1 LSP from DNA system dropped – running dist vect at level 1

Description: A level 1 link state packet received from a DNA system is dropped because this router is running distance vector at level 1.

DNAV.017

Level: UE–ERROR

Short Syntax: DNAV.017 L2 LSP from DNA system dropped – running dist vect at level 2

Long Syntax: DNAV.017 L2 LSP from DNA system dropped – running dist vect at level 2

Description: A level 2 link state packet received from a DNA system is dropped because this router is running distance vector at level 2.

DNAV.018

Level: UE–ERROR

Short Syntax: DNAV.018 ISIS hello dropped – nonmatching Phase IV areas

Long Syntax: DNAV.018 ISIS hello dropped – nonmatching Phase IV areas

Description: An ISIS hello PDU is dropped because the Phase IV area ad–dress in the area address option does not match this router's Phase IV area address.

DNAV.019

Level: C–INFO

Short Syntax: DNAV.019 Adj up; new DNA V endnode *area.node* cir *number* net *network_name*

Long Syntax: DNAV.019 Adjacency up; new DNA V endnode *area.node* circuit *number* network *network_name*

Description: There is now an adjacency with the specified DNA Phase V end node on the specified network.

Cause: Received valid ISO ESIS hello message.

DNAV.020

Level: UE–ERROR

Short Syntax: DNAV.020 Trans DNIV pkt not forwarded – mapping ofout adj ID *area.node* to SNPA add failed

Long Syntax: DNAV.020 Translated DECnet IV packet not forwarded – mapping of output adjacency's Phase IV ID *area.node* to an OSI SNPA address failed.

Description: The translation of a DECnet IV packet to a DECnet V packet failed because a mapping couldn't be found between the out–put adjacency's DECnet IV ID and an OSI SNPA address.

Cause: An end system adjacency doesn't exist in the OSI database for the next hop system.

DNAV.021

Level: C–INFO

Short Syntax: DNAV.021 Adj up; new DNA IV VAXcluster alias *area.node* cir *number* net *network_name*

Long Syntax: DNAV.021 Adjacency up; new DNA IV VAXcluster alias *area.node* circuit *number* network *network_name*

Description: There is now a DNA Phase IV endnode adjacency representing a VAX cluster alias address on the specified network.

Cause: Received a valid DNA IV Level 1 Routing message which advertises a VAXcluster alias address.

DNAV.022

Level: C–INFO

Short Syntax: DNAV.022 Adj dwn; DNA IV VAXcluster alias *area.node* cir *number* net *network_name*

Long Syntax: DNAV.022 Adjacency down; DNA IV VAXcluster alias *area.node* circuit *number* network *network_name*

Description: A DNA Phase IV endnode adjacency representing a VAX cluster alias address has gone down.

Cause: The adjacency to the DNA IV router that was advertising the alias address has timed out.

Cause: Received a valid DNA IV Level 1 Routing message from the router that was advertising the alias address, which now advertises a different alias address or no alias address.

DVM

Distance-Vector Multicast Routing Protocol

This chapter describes messages for the Distance–Vector Multicast Routing Protocol for the IP routing protocol. For information about message content and how to use the message, refer to the Introduction.

DVM.001

Level: UE–ERROR

Short Syntax: DVM.001 Unknown DVMRP code from *IP_source*

Long Syntax: DVM.001 Received unknown DVMRP code from *IP_source*

Description: A DVMRP message was received from the specified source, how–ever it has an unrecognized IGMP code value. The packet is discarded.

DVM.002

Level: UE–ERROR

Short Syntax: DVM.002 No matching VIF for pkt from *IP_source*

Long Syntax: DVM.002 No matching DVMRP interface for packet from *IP_source*

Description: A DVMRP message was received from the specified source, how–ever, no matching DVMRP interface could be found. This probably indicates a configuration error (either in the source, or in the logging router). The packet is discarded.

DVM.003

Level: P–TRACE

Short Syntax: DVM.003 Rcvd DVMRP Report from *IP_source*

Long Syntax: DVM.003 Received DVMRP Report from *IP_source*

Description: A DVMRP report (routing update) has been received from the specified source. This is a normal, periodic event, and can cause additions to the DVMRP routing table.

DVM.004

Level: U–TRACE

Short Syntax: DVM.004 Rcvd DVMRP probe from *IP_source*

Long Syntax: DVM.004 Received DVMRP probe from *IP_source*

Description: A DVMRP probe has been received from the specified source. This is somewhat unusual, and should only happen when the DVMRP conversation on the interface is just beginning.

DVM.005

Level: UE–ERROR

Short Syntax: DVM.005 Rcvd bad DVMRP update from *IP_source*

Long Syntax: DVM.005 Received bad DVMRP update from *IP_source*

Description: A DVMRP update has been received from the specified source. The update was improperly formatted, and at least part of its contents were discarded.

DVM.006

Level: U–TRACE

Short Syntax: DVM.006 Add phy int *IP_interface_address* cost *cost* thresh *threshold*

Long Syntax: DVM.006 Add physical interface *IP_interface_address* cost *cost* thresh *threshold*

Description: DVMRP has been enabled on the specified physical interface,with the given cost and threshold parameters.

DVM.007

Level: U–TRACE

Short Syntax: DVM.007 Add tunnel *tunnel_source–>tunel_destination* cost *cost* thresh *threshold*

Long Syntax: DVM.007 Add tunnel *tunnel_source–>tunel_destination* cost *cost* thresh *threshold*

Description: A DVMRP tunnel has been configured between the given source and destination, with the specified cost and threshold parameters.

DVM.008

Level: U–TRACE

Short Syntax: DVM.008 Add MOSPF cost *cost* thresh *threshold*

Long Syntax: DVM.008 Add MOSPF cost *cost* thresh *threshold*

Description: Tunneling of DVMRP through the MOSPF cloud has been enabled,with the given cost and threshold parameters.

DVM.009

Level: U–TRACE

Short Syntax: DVM.009 Add/update route to *source_network* via *neighbor_IP_address*

Long Syntax: DVM.009 Add route to source *source_network* via neighbor *neighbor_IP_address*

Description: Processing a DVMRP update, or the fact that an interface came up, has caused us to either create or revise a routing table entry for the particular source.

DVM.010

Level: U–TRACE

Short Syntax: DVM.010 Delete route to *source_network*

Long Syntax: DVM.010 Delete route to source *source_network*

Description: A neighbor has informed us that the source is no longer reachable.

DVM.011

Level: U–TRACE

Short Syntax: DVM.011 Add neighbor *neighbor_IP_address*

Long Syntax: DVM.011 Add neighbor *neighbor_IP_address*

Description: A new DVMRP neighbor has been discovered, through the receipt of a probe or update message.

DVM.012

Level: U–TRACE

Short Syntax: DVM.012 Delete neighbor *neighbor_IP_address*

Long Syntax: DVM.012 Delete neighbor *neighbor_IP_address*

Description: Neighbor is no longer reachable. It has either timed out or its associated interface has gone down.

DVM.013

Level: C–TRACE

Short Syntax: DVM.013 Sending DVMRP probe to *neighbor_IP_address*, VIF:*VIF_index*

Long Syntax: DVM.013 Sending probe to *neighbor_IP_address*, VIF *VIF_index*

Description: Sent a DVMRP neighbor probe to the specified address.

DVM.014

Level: C–TRACE

***Short Syntax*:** DVM.014 Sending DVMRP update to *neighbor_IP_address*, VIF:*VIF_index*

***Long Syntax*:** DVM.014 Sending probe to *neighbor_IP_address*, VIF: *VIF_index*

Description: Sent a DVMRP routing update to the specified address.

DVM.015

Level: U–TRACE

***Short Syntax*:** DVM.015 Route to *source_network* timed out

***Long Syntax*:** DVM.015 Route to source *source_network* timed out

Description: Route to a particular source has timed out.

DVM.016

Level: U–TRACE

***Short Syntax*:** DVM.016 Neighbor *neighbor_IP_address* timed out

***Long Syntax*:** DVM.016 Neighbor *neighbor_IP_address* has timed out

Description: A neighbor has timed out. We did not get any updates from it lately.

DVM.017

Level: UI–ERROR

***Short Syntax*:** DVM.017 No mem for source *source_network*

***Long Syntax*:** DVM.017 No memory for source network *source_network*

Description: Either a) we don't have enough heap memory to allocate a DVMRP routing table entry or b) the IP routing table has overflowed. In any case, we cannot recognize the new source. If this source is a directly connected subnet, we won't be able to run IGMP on the subnet either.

DVM.018

Level: U–TRACE

***Short Syntax*:** DVM.018 Added MOSPF route *source_network*

***Long Syntax*:** DVM.018 Added MOSPF route *source_network*

Description: Started advertising a MOSPF route through DVMRP.

DVM.019

Level: U–TRACE

Short Syntax: DVM.019 Deleted MOSPF route *source_network*

Long Syntax: DVM.019 Deleted MOSPF route *source_network*

Description: Stopped advertising a MOSPF route through DVMRP.

DVM.020

Level: UI–ERROR

Short Syntax: DVM.020 No room for neighbor *neighbor_IP_address*

Long Syntax: DVM.020 No room for neighbor *neighbor_IP_address*

Description: There was no room to allocate the data structure for a new neighbor. DVMRP routes form the neighbor will be ignored.

DVM.021

Level: P–TRACE

Short Syntax: DVM.021 Packet rcvd from mis/unconfigured tunnel *source_IP_address*

Long Syntax: DVM.021 Packet received from mis/unconfigured tunnel *source_IP_address*

Description: A packet has been received through protocol 4 (IP encapsulation). The packet should be source by the other end of a tunnel. Either the tunnel has not been configured, or it has been configured to be source–routed instead of encapsulated.

EGP

Exterior Gateway Protocol

This chapter describes Exterior Gateway Protocol (EGP) messages. EGP is a routing protocol used by the IP protocol family. For information about message content and how to use the message, refer to the Introduction.

EGP.001

Level: U–TRACE
Short Syntax: EGP.001 pkt *source_ip_address –> destination_ip_address* no srvr
Long Syntax: EGP.001 packet *source_ip_address –> destination_ip_address* no server
Description: An EGP packet was sent to this station. However, EGP is not in the load.

EGP.002

Level: U–TRACE
Short Syntax: EGP.002 pkt *source_ip_address –> destination_ip_address* no srvr
Long Syntax: EGP.002 packet *source_ip_address –> destination_ip_address* no server
Description: An EGP packet was sent to this station. However, EGP is disabled.

EGP.003

Level: P–TRACE
Short Syntax: EGP.003 pkt in *type code* frm *source_ip_address*
Long Syntax: EGP.003 packet in *type code* from *source_ip_address*
Description: An EGP packet was received.

EGP.004

Level: UE–ERROR
Short Syntax: EGP.004 rcv acq from *source_ip_address* rjct bd AS AS_number
Long Syntax: EGP.004 received acquire from *source_ip_address* reject bad AS AS_number
Description: An EGP acquire packet was received but the Autonomous System number did not match what was expected.

EGP.005

Level: UE–ERROR
Short Syntax: EGP.005 rcv err pkt frm *source_ip_address info reason*
Long Syntax: EGP.005 received error packet from *source_ip_address info reason*
Description: An EGP error packet was received.

EGP.006

Level: UE–ERROR
Short Syntax: EGP.006 rcv err pkt frm *source_ip_address info reason*
Long Syntax: EGP.006 received error packet from *source_ip_address info reason*
Description: An EGP error packet was received.

EGP.007

Level: UE–ERROR
Short Syntax: EGP.007 rcv bd pkt *type code* frm *source_ip_address*
Long Syntax: EGP.007 received bad packet *type code* from *source_ip_address*
Description: An EGP packet was received with a bad type field.

EGP.008

Level: P–TRACE
Short Syntax: EGP.008 snd err pkt error to *destination_ip_address type code* rsn *reason*
Long Syntax: EGP.008 snd error packet error to *destination_ip_address type code* reason *reason*
Description: An EGP error packet is being sent to an EGP neighbor.

EGP.009

Level: UI–ERROR
Short Syntax: EGP.009 no pkt typ *type error* to *destination_ip_address*
Long Syntax: EGP.009 no packet type *type error* to *destination_ip_address*
Description: No buffer was available to send an EGP error packet.

EGP.010

Level: UI–ERROR
Short Syntax: EGP.010 rtng tbl ovfl rt to *destination_ip_address*
Long Syntax: EGP.010 routing tbl ovfl rt to *destination_ip_address*
Description: There were no routing table entries available to add an EGP route.

EGP.011

Level: C–INFO

Short Syntax: EGP.011 nw dst *new_metric* old *old_metric* to *destination_ip_address* through next_hop_address

Long Syntax: EGP.011 new distance *new_metric* old *old_metric* to *destination_ip_address* through next_hop_address

Description: An EGP route was received with a new metric.

EGP.012

Level: UE–ERROR

Short Syntax: EGP.012 gw *source_ip_address* bd addr *destination_ip_address*

Long Syntax: EGP.012 gateway *source_ip_address* bad addr *destination_ip_address*

Description: An EGP update was received with an IP address of 0.0.0.0.

EGP.013

Level: C–INFO

Short Syntax: EGP.013 nw rt to *destination_ip_address* through *netxthop_ip_address* at metric old at *old_metric* typ *rte_type*

Long Syntax: EGP.013 nw rt to *destination_ip_address* through *netxthop_ip_address* at *metric* old at *old_metric* type *rte_type*

Description: A new EGP route has been added to the routing database.

EGP.014

Level: UE–ERROR

Short Syntax: EGP.014 bd acq *code* frm *source_ip_address*

Long Syntax: EGP.014 bad acquire *code* from *source_ip_address*

Description: A bad EGP acquire packet was received.

EGP.015

Level: P–TRACE

Short Syntax: EGP.015 rcv *type info source_ip_address*

Long Syntax: EGP.015 received *type info source_ip_address*

Description: An EGP acquire or cease packet was received.

EGP.016

Level: UE–ERROR
Short Syntax: EGP.016 old rfs frm *source_ip_address* state *state*
Long Syntax: EGP.016 old rfs from *source_ip_address* state *state*
Description: A refuse packet was received but the state was not acquired.

EGP.017

Level: UE–ERROR
Short Syntax: EGP.017 gw nt repnd *destination_ip_address*
Long Syntax: EGP.017 gateway nt repnd *destination_ip_address*
Description: The maximum number of acquire packets have been sent but there has been no response.

EGP.018

Level: UI–ERROR
Short Syntax: EGP.018 no pkt typ *type type* to *destination_ip_address*
Long Syntax: EGP.018 no packet type *type type* to *destination_ip_address*
Description: No packet buffer available to send EGP acquire packet.

EGP.019

Level: P–TRACE
Short Syntax: EGP.019 snd acq *destination_ip_address*
Long Syntax: EGP.019 snd acquire *destination_ip_address*
Description: An acquire packet is being sent.

EGP.020

Level: UI–ERROR
Short Syntax: EGP.020 no pkt typ *type type* to *destination_ip_address*
Long Syntax: EGP.020 no packet type *type type* to *destination_ip_address*
Description: No packet buffer available to send EGP cease packet.

EGP.021

Level: P–TRACE
Short Syntax: EGP.021 snd cse *destination_ip_address*
Long Syntax: EGP.021 snd cease *destination_ip_address*
Description: An EGP cease packet was sent.

EGP.022

Level: UI–ERROR
Short Syntax: EGP.022 no acq pkt *code info*
Long Syntax: EGP.022 no acquire packet *code info*
Description: No packet buffer to send EGP packet.

EGP.023

Level: P–TRACE
Short Syntax: EGP.023 rcv hlo *source_ip_address*
Long Syntax: EGP.023 received hello *source_ip_address*
Description: An EGP hello packet was received.

EGP.024

Level: UE–ERROR
Short Syntax: EGP.024 old IHU frm *source_ip_address*
Long Syntax: EGP.024 old IHU from *source_ip_address*
Description: An “I Heard You” packet with a bad ID was received from an EGP neighbor.

EGP.025

Level: P–TRACE
Short Syntax: EGP.025 rcv ihu *source_ip_address*
Long Syntax: EGP.025 received ihu *source_ip_address*
Description: An “I Heard You” packet was received from an EGP neighbor.

EGP.026

Level: UE–ERROR
Short Syntax: EGP.026 rcv bd hello frm *source_ip_address* cd code
Long Syntax: EGP.026 received bad hello from *source_ip_address* cd code
Description: A bad hello packet (invalid type) was received from an EGP neighbor.

EGP.027

Level: UI–ERROR
Short Syntax: EGP.027 no pkt typ *type type* to *destination_ip_address*
Long Syntax: EGP.027 no packet type *type type* to *destination_ip_address*
Description: No packet buffer to send EGP hello packet.

EGP.028

Level: P–TRACE
Short Syntax: EGP.028 snd hlo *destination_ip_address*
Long Syntax: EGP.028 snd hello *destination_ip_address*
Description: A EGP hello packet was sent.

EGP.029

Level: UE–ERROR
Short Syntax: EGP.029 rcv poll *source_ip_address*
Long Syntax: EGP.029 received poll *source_ip_address*
Description: Poll packet received.

EGP.030

Level: UE–ERROR
Short Syntax: EGP.030 no resp NR poll *source_ip_address*
Long Syntax: EGP.030 no response NR poll *source_ip_address*
Description: No response to EGP poll, maximum retries exceeded.

EGP.031

Level: UE–ERROR
Short Syntax: EGP.031 no pkt fr poll *destination_ip_address*
Long Syntax: EGP.031 no packet to send poll to *destination_ip_address*
Description: No packet buffer available to send poll message.

EGP.032

Level: P–TRACE
Short Syntax: EGP.032 snd poll *destination_ip_address*
Long Syntax: EGP.032 snd poll *destination_ip_address*
Description: An EGP poll packet is being sent.

EGP.033

Level: UE–ERROR
Short Syntax: EGP.033 old rt pkt *source_ip_address id seq retry*
Long Syntax: EGP.033 old rt packet *source_ip_address id seq retry*
Description: An old EGP update has been received.

EGP.034

Level: P–TRACE
Short Syntax: EGP.034 rcv upd *source_ip_address*
Long Syntax: EGP.034 received update *source_ip_address*
Description: An EGP update has been received.

EGP.035

Level: C–INFO
Short Syntax: EGP.035 del rt to *destination_ip_address* through *nexthop_ip_address*
Long Syntax: EGP.035 delete route to *destination_ip_address* through *nexthop_ip_address*
Description: An EGP route has timed out.

EGP.036

Level: C–INFO
Short Syntax: EGP.036 del rtng db
Long Syntax: EGP.036 delete routing db
Description: EGP routing database deleted.

EGP.037

Level: UI–ERROR
Short Syntax: EGP.037 bd init nbr *destination_ip_address* rsn *reason*, ign
Long Syntax: EGP.037 bad init neighbor *destination_ip_address* reason *reason*, ign
Description: An EGP neighbor relationship could not be formed.

EGP.038

Level: C–INFO
Short Syntax: EGP.038 new nbr *neighbor_ip_address*
Long Syntax: EGP.038 new neighbor *neighbor_ip_address*
Description: A new EGP neighbor relationship has been created.

EGP.039

Level: C–INFO
Short Syntax: EGP.039 gw aq st *neighbor_ip_address*
Long Syntax: EGP.039 gateway aquire state *neighbor_ip_address*
Description: Go into acquire state.

EGP.040

Level: C–INFO
Short Syntax: EGP.040 nbr up *neighbor_ip_address*
Long Syntax: EGP.040 neighbor up *neighbor_ip_address*
Description: Went into neighbor state.

EGP.041

Level: C–INFO
Short Syntax: EGP.041 nbr cse *neighbor_ip_address*
Long Syntax: EGP.041 neighbor cease *neighbor_ip_address*
Description: An EGP cease packet is to be sent.

EGP.042

Level: UI–ERROR
Short Syntax: EGP.042 nbr up cnt err
Long Syntax: EGP.042 neighbor up count error
Description: The count of the number of up EGP neighbors has gone negative.

EGP.043

Level: UI–ERROR
Short Syntax: EGP.043 nbr core cnt err
Long Syntax: EGP.043 neighbor core count error
Description: The count of the number of up EGP core neighbors has gone negative.

EGP.044

Level: C–INFO
Short Syntax: EGP.044 nbr dsc *neighbor_ip_address*
Long Syntax: EGP.044 neighbor discard *neighbor_ip_address*
Description: An EGP neighbor is being shut down.

EGP.045

Level: UE–ERROR
Short Syntax: EGP.045 gw nw unreach *destination_ip_address*
Long Syntax: EGP.045 gateway now unreach *destination_ip_address*
Description: An EGP neighbor has become unreachable.

EGP.046

Level: C–INFO

Short Syntax: EGP.046 gw nw reach *destination_ip_address*

Long Syntax: EGP.046 gateway nw reach *destination_ip_address*

Description: An EGP neighbor is reachable.

EGP.047

Level: UE–ERROR

Short Syntax: EGP.047 nbr dsc *destination_ip_address*

Long Syntax: EGP.047 neighbor dsc *destination_ip_address*

Description: An EGP neighbor has become unreachable.

EGP.048

Level: UE–ERROR

Short Syntax: EGP.048 rcv max hello or acq from *source_ip_address num_hellos num_aquires*

Long Syntax: EGP.048 received max hello or acquire from *source_ip_address num_hellos num_aquires*

Description: Received maximum number of hellos or acquires from EGP neighbor.

EGP.049

Level: P–TRACE

Short Syntax: EGP.049 rcv pkt *type code* nt frm neighbor *source_ip_address*

Long Syntax: EGP.049 received packet *type code* nt from neighbor *source_ip_address*

Description: An EGP packet was received from a neighbor.

EGP.050

Level: UE–ERROR

Short Syntax: EGP.050 rcv bd pkt *type code* frm *source_ip_address*

Long Syntax: EGP.050 received bad packet *type code* from *source_ip_address*

Description: A bad EGP packet was received from a neighbor.

EGP.051

Level: UE–ERROR

***Short Syntax*:** EGP.051 nbr *source_ip_address* bd gw *destination_ip_address*

***Long Syntax*:** EGP.051 neighbor *source_ip_address* bad gateway *destination_ip_address*

Description: An EGP packet was received with an unreachable destination.

EGP.052

Level: UE–ERROR

***Short Syntax*:** EGP.052 bad cse frm *source_ip_address* state *state*

***Long Syntax*:** EGP.052 bad cease from *source_ip_address* state *state*

Description: A bad acknowledgment to a cease packet was received.

EGP.053

Level: UE–ERROR

***Short Syntax*:** EGP.053 old acq frm *source_ip_address* state *state*

***Long Syntax*:** EGP.053 old acquire from *source_ip_address* state *state*

Description: A bad acquire packet was received.

EGP.054

Level: UI–ERROR

***Short Syntax*:** EGP.054 no pkt fr poll response *destination_ip_address*

***Long Syntax*:** EGP.054 no packet to send EGP update to *destination_ip_address*

Description: No packet buffer available to send poll response (EGP up–date) message. Either the update is too large, or there is a temporary buffer shortage.

EGP.055

Level: P–TRACE

***Short Syntax*:** EGP.055 snd update *destination_ip_address, EGP_data_length* bytes

***Long Syntax*:** EGP.055 Send NR update packet to *destination_ip_address,EGP_data_length* bytes

Description: An EGP neighbor reachability update packet is being sent. This is normally done in response to a poll packet.

ESIS

End System–Intermediate System Protocol

This chapter describes End System–Intermediate System (ESIS) routing protocol messages. ESIS is part of the OSI–CLNP forwarder. For information about message content and how to use the message, refer to the Introduction.

ESIS.001

Level: UE–ERROR
Short Syntax: ESIS.001 ESIS input que ovflw
Long Syntax: ESIS.001 ESIS input queue overflow
Description: The ESIS task input queue has overflowed, packet is dropped.

ESIS.002

Level: UE–ERROR
Short Syntax: ESIS.002 rcvd incmplt pkt
Long Syntax: ESIS.002 received incomplete packet
Description: A packet fragment recognized as an ESIS packet was received.

ESIS.003

Level: UE–ERROR
Short Syntax: ESIS.003 rcvd pkt bad chksm=*pkt_chksum*
Long Syntax: ESIS.003 received packet with a bad checksum = *pkt_chksum*
Description: An ESIS packet was received but had a bad checksum.

ESIS.004

Level: UE–ERROR
Short Syntax: ESIS.004 rcvd pkt bad vers # =*version_number*
Long Syntax: ESIS.004 received packet with a bad version number (vers =*version_number*)
Description: An ESIS packet was received but had a bad or unsupported version number.

ESIS.005

Level: UE–ERROR

Short Syntax: ESIS.005 rcvd pkt bad typ # =*type_field*

Long Syntax: ESIS.005 received packet with a bad type field (vers =*type_field*)

Description: An ESIS packet was received but had a bad or unsupported type field.

ESIS.006

Level: UE–ERROR

Short Syntax: ESIS.006 no iob avail to snd hello

Long Syntax: ESIS.006 no i/o buffer available to send hello

Description: An attempt to send an ESIS hello failed because of a lack of system i/o buffers.

ESIS.007

Level: UE–ERROR

Short Syntax: ESIS.007 cnnt snt hello pkt hndlr err

Long Syntax: ESIS.007 cannot send a hello packet, handler error

Description: An ESIS hello packet could not be sent because of a handler error.

ESIS.008

Level: P–TRACE

Short Syntax: ESIS.008 sent hello *source_NSAP* on int *interface_#*

Long Syntax: ESIS.008 sent hello packet with source nsap *source_NSAP* on int *interface_#*

Description: An ESIS hello packet was sent out on an interface.

ESIS.009

Level: UE–ERROR

Short Syntax: ESIS.009 rcvd hello packet with a bad header

Long Syntax: ESIS.009 rcvd hello packet with a bad header

Description: Received hello packet with a holding time or reserved field.

ESIS.010

Level: UE–ERROR

Short Syntax: ESIS.010 rcvd hello bad nsap *source_NSAP*

Long Syntax: ESIS.010 received hello with bad nsap *source_NSAP*

Description: An ESIS hello packet was received with a bad nsap or one that overran the packet.

ESIS.011

Level: UE–ERROR

***Short Syntax*:** ESIS.011 rcvd hello pkt bad opt

***Long Syntax*:** ESIS.011 received packet with a bad optional parameter

Description: An ESIS CLNP data packet was received with bad option parameter(s).

ESIS.012

Level: P–TRACE

***Short Syntax*:** ESIS.012 rcvd hello from *source_NSAP* int *interface* net *network_name*

***Long Syntax*:** ESIS.012 rcvd hello packet with source nsap *source_NSAP* on int *interface*, net *network_name*

Description: An ESIS hello packet was received on the specified interface.

ESIS.013

Level: UE–ERROR

***Short Syntax*:** ESIS.013 rcvd hello unsp dom src *source_NSAP*

***Long Syntax*:** ESIS.013 rcvd hello packet unsupported domain *source_NSAP*

Description: An ESIS hello packet was received with an unrecognized IDI.

ESIS.014

Level: UE–ERROR

***Short Syntax*:** ESIS.014 no rsrc to instl rt

***Long Syntax*:** ESIS.014 no resources to install route

Description: An ESIS hello packet was received but there were no resources available to install the route.

ESIS.015

Level: UE–ERROR

***Short Syntax*:** ESIS.015 rcvd hello ng cnfltng rt *source_NSAP*

***Long Syntax*:** ESIS.015 received hello no good conflicting route *source_NSAP*

Description: An ESIS hello packet was received but could not be entered into the database since there was a static or dynamic route already defined that conflicted with the route in the hello.

ESIS.016

Level: UE–ERROR

Short Syntax: ESIS.016 tmd out rte reac *source_NSAP*

Long Syntax: ESIS.016 timed out route reactivated *source_NSAP*

Description: An ESIS hello packet was received with a route that had been previously timed out.

ESIS.017

Level: UE–ERROR

Short Syntax: ESIS.017 no rsrc to snd rdrct

Long Syntax: ESIS.017 no resources to send redirect

Description: An ESIS redirect packet could not be sent due to a lack of resources.

ESIS.018

Level: UE–ERROR

Short Syntax: ESIS.018 rdrct nt snt hndlr err

Long Syntax: ESIS.018 redirect not sent, handler error

Description: An ESIS redirect packet could not be sent due to a handler error.

ESIS.019

Level: P–TRACE

Short Syntax: ESIS.019 sent rdrct to:*dest_NSAP*

Long Syntax: ESIS.019 sent redirect packet to:*dest_NSAP*

Description: An ESIS redirect packet was sent out on an interface.

ESIS.020

Level: UE–ERROR

Short Syntax: ESIS.020 tmd out rte *source_NSAP*

Long Syntax: ESIS.020 timed out route *source_NSAP*

Description: An ESIS hello route has been timed out.

ESIS.021

Level: UI_ERROR

Short Syntax: ESIS.021 Unable to allocate resources for a new ES adjacency

Long Syntax: ESIS.021 Unable to allocate resources for a new ES adjacency

Description: We were unable to get an adjacency structure for a new end system adjacency.

ESIS.022

Level: UE_ERROR

Short Syntax: ESIS.022 hello PDU dropped, rcvd over p–to–p cir *cct_num*

Long Syntax: ESIS.022 hello PDU dropped, received over point–to–point circ *cct_num*

Description: An ESIS hello PDU was received over a point–to–point circuit– the packet was dropped because ESIS does not run over point–to–point circuits.

ESIS.023

Level: UE_ERROR

Short Syntax: ESIS.023 hello PDU dropped, no matching area address

Long Syntax: ESIS.023 ESIS hello PDU dropped, no matching area address

Description: An ESIS hello PDU was dropped because the area address portion of its source NSAP didn't match one of the router's manual area addresses.

ESIS.024

Level: P–TRACE

Short Syntax: ESIS.024 dropped hello from *source_NSAP* int *interface* net *network_name* manual ES adjacency exists

Long Syntax: ESIS.024 dropped hello packet with source nsap *source_NSAP* on int *interface*, net *network_name* – manual ES adjacency exists

Description: An ESIS hello packet was dropped on the specified interface because a manual adjacency exists for the ES.

ETH

Ethernet Network Interface

This chapter describes Ethernet™ Network Interface messages. For information about message content and how to use the message, refer to the Introduction.

ETH.001

Level: P–TRACE

Short Syntax: ETH.001 brd rcv unkwn typ *packet_type source_Ethernet_address –> destination_Ethernet_address* nt *network*

Long Syntax: ETH.001 broadcast packet received with unknown Ethernet type *packet_type* from host *source_Ethernet_address* to *destination_Ethernet_address* network *network*

Description: A broadcast packet was received with an unknown or unsupported Ethernet type field.

ETH.002

Level: UE–ERROR

Short Syntax: ETH.002 rcv unkwn typ *packet_type source_Ethernet_address –>destination_Ethernet_address* nt *network*

Long Syntax: ETH.002 packet received with unknown Ethernet type field *packet_type* from *source_Ethernet_address* to *destination_Ethernet_address* network *network*

Description: A non–broadcast packet was received with an unknown or un–supported Ethernet type field.

ETH.003

Level: P–TRACE

Short Syntax: ETH.003 brd 802.3 bd ln *actual_length claimed_length source_Ethernet_address –> destination_Ethernet_address* nt *network*

Long Syntax: ETH.003 broadcast packet received with a bad 802.3 length field actual *actual_length* claimed *claimed_length* from *source_Ethernet_address* to *destination_Ethernet_address* network *network*

Description: A broadcast packet was received with a type field that indicated 802.3 but was shorter than data length claimed in the802.3 header.

ETH.004

Level: UE–ERROR

Short Syntax: ETH.004 802.3 bd ln *actual_length claimed_length source_Ethernet_address* –> *destination_Ethernet_address* nt *network*

Long Syntax: ETH.004 packet received with a bad 802.3 length field actual *actual_length* claimed *claimed_length* from *source_Ethernet_address* to *destination_Ethernet_address* network *network*

Description: A non–broadcast packet was received with a type field that indicated 802.3 but was shorter than data length claimed in the 802.3 header.

ETH.005

Level: UE–ERROR

Short Syntax: ETH.005 DN bd ln *actual_length claimed_length source_Ethernet_address* –> *destination_Ethernet_address* nt *network*

Long Syntax: ETH.005 DECnet packet received with a bad length actual *actual_length* claimed *claimed_length* from *source_Ethernet_address* to *destination_Ethernet_address* network *network*

Description: A DECnet packet was received with a length field that was larger than the actual length of the packet.

ETH.006

Deleted: Message deleted.

ETH.007

Deleted: Message deleted.

ETH.008

Deleted: Message deleted.

ETH.009

Deleted: Message deleted.

ETH.010

Level: C–INFO

Short Syntax: ETH.010 LLC unk SAP *DSAP source_Ethernet_address* –>*destination_Ethernet_address* nt *network*

Long Syntax: ETH.010 802.2 LLC packet received with unknown DSAP *DSAP* from host *source_Ethernet_address* to *destination_Ethernet_address* network *network*

Description: An 802.2 LLC packet was received from the network with an inactive (unrecognized) DSAP.

ETH.011

Level: C–INFO

Short Syntax: ETH.011 LLC nt typ 1 *LLC_control_type* nt *network*

Long Syntax: ETH.011 802.2 LLC packet received, not Type 1*LLC_control_type* network *network*

Description: A packet was received from the network that had an LLC but was not a Type 1 LLC.

ETH.012

Level: C–INFO

Short Syntax: ETH.012 LLC RSP *LLC_SSAP* nt *network*

Long Syntax: ETH.012 LLC RESPONSE packet received *LLC_SSAP* network *network*

Description: An LLC response was received from the network.

ETH.013

Level: C–INFO

Short Syntax: ETH.013 LLC XID *LLC_SSAP* nt *network*

Long Syntax: ETH.013 LLC XID packet received *LLC_SSAP* network *network*

Description: An LLC XID packet was received from the network.

ETH.014

Level: C–INFO

Short Syntax: ETH.014 LLC TEST *LLC_SSAP* nt *network*

Long Syntax: ETH.014 LLC TEST packet received *LLC_SSAP* network *network*

Description: An LLC TEST packet was received from the network.

ETH.015

Level: U–INFO

Short Syntax: ETH.015 unrec ctl *LLC_control_field* nt *network*

Long Syntax: ETH.015 packet received with unrecognized control field *LLC_control_field* network *network*

Description: A packet was received from the network that had an illegal control field or UI.

ETH.016

Deleted: Message deleted.

ETH.017

Level: P–TRACE

Short Syntax: ETH.017 LOOP rcv *source_Ethernet_address –> destination_Ethernet_address*, nt *network*

Long Syntax: ETH.017 Loopback Protocol frame received from *source_Ethernet_address* to *destination_Ethernet_address*,network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet was received.

ETH.018

Level: UE–ERROR

Short Syntax: ETH.018 LOOP odd skp *count*,*source_Ethernet_address –>destination_Ethernet_address*, nt *network*

Long Syntax: ETH.018 Loopback Protocol, odd skipCount *count* from *source_Ethernet_address* to *destination_Ethernet_address*, network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet had an odd skip Count in the packet. It will be discarded.

Cause: Programming error on remote node.

ETH.019

Level: UE–ERROR

Short Syntax: ETH.019 LOOP func *function* not forw,*source_Ethernet_address –> destination_Ethernet_address*, nt *network*

Long Syntax: ETH.019 Loopback Protocol, function *function* not Forward Data from *source_Ethernet_address* to *destination_Ethernet_address*, network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet did not have a function code of forward (2). It will be discarded.

Cause: Function code was reply (1), because we were the ultimate destination of this packet.

Action: None.

Cause: Undefined function code, due to programming error in remote node.

ETH.020

Level: UE–ERROR

Short Syntax: ETH.020 LOOP mc fwd dst *forward_Ethernet_address*, *source_Ethernet_address –> destination_Ethernet_address*, nt *network*

Long Syntax: ETH.020 Loopback Protocol, multicast forward address *forward_Ethernet_address* from *source_Ethernet_address* to *destination_Ethernet_address*, network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet has a forward address that is a multicast. It will be discarded.

Cause: Programming error in remote node.

ETH.021

Level: P–TRACE

Short Syntax: ETH.021 LOOP fwd *source_Ethernet_address –>forward_Ethernet_address*, nt *network*

Long Syntax: ETH.021 Loopback Protocol, forwarding from *source_Ethernet_address* to *forward_Ethernet_address*, network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet is being forwarded to the specified next hop.

ETH.022

Level: UI–ERROR

Short Syntax: ETH.022 LOOP fwd to *forward_Ethernet_address* dsc, rsn *code*,nt *network*

Long Syntax: ETH.022 Loopback protocol, forward to *forward_Ethernet_address* discarded, for reason *code*, network *network*

Description: A Ethernet Loopback Protocol (Configuration Testing Protocol) packet could not be forwarded to the specified address,for the reason specified by code.

ETH.023

Level: UI–ERROR

Short Syntax: ETH.023 LLC RSP to *destination_Ethernet_address* dsc, rsn *code*, nt *network*

Long Syntax: ETH.023 LLC response to *destination_Ethernet_address* discarded, for reason *code*, network *network*

Description: An LLC response (XID or TEST) could not be transmitted to the specified address, for the reason specified by code.

ETH.024

Level: UE–ERROR

*Short Syntax***:** ETH.024 MOP bd ln *actual_length claimed_length source_Ethernet_address –> destination_Ethernet_address* nt *network*

*Long Syntax***:** ETH.024 DECnet MOP packet received with a bad length actual *actual_length* claimed *claimed_length* from *source_Ethernet_address* to *destination_Ethernet_address* network *network*

Description: A DECnet MOP packet was received with a length field that was larger than the actual length of the packet.

ETH.025

Level: UE–ERROR

*Short Syntax***:** ETH.025 LOOP bd skp *count, source_Ethernet_address –>destination_Ethernet_address,* nt *network*

*Long Syntax***:** ETH.025 Loopback Protocol, bad skipCount *count* from *source_Ethernet_address* to *destination_Ethernet_address*,network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet had a skip Count in the packet that points to beyond the end of the packet. It will be discarded.

Cause: Programming error on remote node.

ETH.026

Deleted: Message deleted.

ETH.027

Deleted: Message deleted.

ETH.028

Deleted: Message deleted.

ETH.029

Deleted: Message deleted.

ETH.030

Deleted: Message deleted.

ETH.031

Deleted: Message deleted.

ETH.032

Deleted: Message deleted.

ETH.033

Deleted: Message deleted.

ETH.034

Deleted: Message deleted.

ETH.035

Deleted: Message deleted.

ETH.036

Deleted: Message deleted.

ETH.037

Deleted: Message deleted.

ETH.038

Deleted: Message deleted.

ETH.039

Deleted: Message deleted.

ETH.040

Deleted: Message deleted.

ETH.041

Deleted: Message deleted.

ETH.042

Level: UI–ERROR

*Short Syntax***:** ETH.042 Unable to get buf for ethernet packet.

*Long Syntax***:** ETH.042 Unable to get buffer for ethernet packet.

Description: A buffer to set an Ethernet address, or to copy an Ethernet packet couldn't be gotten because of a buffer shortage.

ETH.043

Level: ALWAYS

*Short Syntax***:** ETH.043 CMD596 Command Timeout. Interface *network* being restarted.

*Long Syntax***:** ETH.043 CMD596 Command Timeout. Interface *network* being restarted.

Description: The 82596 chip on the interface card has failed to clear the command field for this interface. The interface will be reinitialized.

ETH.044

Level: ALWAYS

*Short Syntax***:** ETH.044 I5IOCTL Bad Command *network* being restarted.

*Long Syntax***:** ETH.044 I5IOCTL Bad Command *network* being restarted.

Description: An Incorrect command field has been sent to the driver. The interface will be re–initialized.

ETH.045

Level: UI–ERROR

*Short Syntax***:** ETH.045 Eth self-test *selftest_phase* fld *error_condition* nt *network*

*Long Syntax***:** ETH.045 Ethernet self-test phase *selftest_phase* failed:*error_condition*, network *network*

Description: The selftest for the Ethernet card has reported an error during selftest. The phases are "Reset board", "Reset delay", "Check reset done", "Check reset delay", "Init SCB","Init SCB delay", "Init SCB completion", "Read hardware ad–dress", "Set bus throttle timers", "Internal loopback", "Set hardware address", "Enable receive", "Internal loopback(output)", "Check internal loopback data", "External loop–back", "External loopback delay", "External loopback (out–put)", "Check external loopback data", "Network loopback", "Network loopback delay", "Network loopback (output)", "Check network loopback data", "Clear loopback", and "Operational test".

Cause: In the "Reset board" phase, the error "Packetsize of < 1500 bytes" indicates that the interface has been provided with buffers that are too small.

Action: Correct configuration of system that is artificially reducing packet size below Ethernet requirement of 1500 bytes.

Cause: In all phases, the error “No buffers” indicates that there is a severe packet buffer shortage in the router.

Action: Increase buffer memory size, decrease buffer size on configurable networks.

Cause: In phase “Init SCB completion”, the error “ISCP busy not 0” indicates that the BUSY byte of the 82596 Intermediate System Configuration Pointer (ISCP) did not clear after the CA signal was sent.

Action: This indicates a probable hardware problem with the interface or router. Run diagnostics.

Cause: The error “Unexpected receive pkt” indicates that the interface received a packet in a self-test state where it did not expect to receive a packet.

Action: This indicates a possible hardware problem with the interface. Run diagnostics.

Cause: The error “Loop back count error”, indicates that the received loopback packet was not of the same length as the transmitted one.

Action: This indicates a possible hardware problem with the interface. Run diagnostics.

Cause: The error “Loop back stat error” indicates that the receive of the loopback packet had an unsuccessful error status.

Action: This indicates a possible hardware problem with the interface. Run diagnostics.

Cause: The error “Loop back data error” indicates that there was a data mismatch in the loopback packet.

Action: This indicates a possible hardware problem with the interface. Run diagnostics.

Cause: In the “Operational test” phase, the error “maintenance failure” indicates that the interface could not perform a successful maintenance test. (The maintenance test sends one packet and checks for carrier sense.)

Action: Check the transceiver cabling and hardware.

Cause: In all phases, the error “timeout” indicates that the entire selftest did not complete within one-eighth of a second.

ETH.046

Level: UE–ERROR

Short Syntax: ETH.046 IPX pkt in *received_encapsulation* encap ign, using *configured_encapsulation* encaps, nt *network*

Long Syntax: ETH.046 IPX pkt in encapsulation *received_encapsulation* ignored, using encapsulation *configured_encapsulation* on network *network*

Description: This message is generated when an IPX packet is received in a data-link encapsulation (frame) other than the one configured for IPX on this interface. The packet will be ignored. The *received_encapsulation* and *configured_encapsulation* are one of “ETHERNET_802.3”, “ETHERNET_II”, “ETHERNET_802.2”, or “ETHERNET_SNAP”. ETHERNET_802.3 is also known as “Novell”,and ETHERNET_II is also known as “Ethernet”.

Cause: If only one encapsulation is being used on this network,this node’s encapsulation is not the same as all other IPX nodes on the network.

Action: Configure all nodes on network to use same encapsulation.

Cause: If multiple encapsulations are being used on this network, a packet has been received from a node using an encapsulation different from this node.

ETH.047

Level: UI–ERROR

Short Syntax: ETH.047 Eth self-test *selftest_phase* fld *error_condition* nt *network*

Long Syntax: ETH.047 Ethernet port self-test phase *selftest_phase* failed:*error_condition*, network *network*

Description: The self-test for the SCC Ethernet port has reported an error during selftest. The phases are “Reset port”, “Set media selection”, “Set hardware address”, “Network loop–back”, “Enable receive”, “Network loopback (output)”, “Check network loopback data”, “Clear loopback”, “Set multicast ad–dresses”, and “Operational test”.

Cause: In the “Reset port” phase, the error “Packet size of < 1500 bytes” indicates that the interface has been provided with buffers that are too small.

Action: Correct configuration of system that is artificially reducing packet size below Ethernet requirement of 1500 bytes.

Cause: In all phases, the error “No buffers” indicates that there is a severe packet buffer shortage in the router.

Action: Increase memory size, decrease size of routing tables, decrease buffer allocations to networks, decrease buffer size on configurable networks.

Cause: The error “Loop back data error” indicates that there was a data mismatch in the loopback packet.

Action: This indicates a possible hardware problem with the interface. Run diagnostics.

Cause: The error “Loop back count error”, indicates that the received loopback packet was not of the same length as the transmitted one.

Action: This indicates a possible hardware problem with the interface. Run diagnostics.

Cause: The error “Loop back status error” indicates that the receive of the loopback packet had an unsuccessful error status.

Action: This indicates a possible hardware problem with the interface. Run diagnostics.

Cause: In all phases, the error “Timeout” indicates that the entire selftest did not complete within one–eighth of a second.

Action: This indicates a possible hardware problem with the interface. Run diagnostics.

Cause: In the “Operational test” phase, the error “maintenance failure” indicates that the interface could not perform a successful maintenance test. (The maintenance test sends one packet and checks for carrier sense.)

Action: Check the transceiver cabling and hardware. The router is probably not connected to the Ethernet correctly, or there is a hardware failure.

Panic “ethbdtbl”

Short Syntax: ethbdtbl: eth_llc tbl out of date

Description: The Ethernet LLC table is out of date.

Action: Contact customer service.

Panic “ethintm”

Short Syntax: ethintm: net intf mismtch

Description: The Ethernet data structure “net” is not Ethernet related.

Action: Contact customer service.

Panic “ethbprt”

Short Syntax: ethbprt: bad prot init

Description: An unsupported Network Layer protocol tried to initialize Ethernet handler.

Action: Contact customer service.

Panic “ethbipx”

Short Syntax: ethbipx: bad IPX rqst shd be 8137

Description: An unsupported IPX packet was given to the Ethernet handler for transmission.

Action: Contact customer service.

Panic “ethbreq”

Short Syntax: ethbreq: bad xmit rqst

Description: An unsupported protocol packet was given to the Ethernet handler for transmission.

Action: Contact customer service.

Panic “ethtbig”

Short Syntax: ethtbig: bad xmit rqst pkt too lg

Description: A packet was given to the Ethernet handler for transmission that was too large.

Action: Contact customer service.

Panic “ethnbuf”

Short Syntax: ethnbuf: no buf to set addr

Description: A buffer to set an Ethernet address could not be allocated.

Action: Contact customer service.

Panic “ethsrtmcr”

Short Syntax: ethsrtmcr: multicast address previously reserved

Description: One of the multicast addresses enabled on this interface is one of the multicast addresses in the range 01–80–C2–00–00–00 through 01–80–C2–00–00–0F.

Cause: Possibly one of these addresses that is being used by a protocol where the user can select the multicast address, such as the ES–IS and IS–IS protocols in ISO.

Action: Don’t use the reserved addresses.

Panic “ethsrtnm”

Short Syntax: ethsrtnm: no memory to register own MAC addr

Description: The learning database is so small that there are not enough free entries to learn the address of this interface.

Action: Increase the size of the learning database.

Panic “ethsrtnmm”

Short Syntax: ethsrtnmm: no memory to register mutlicast address

Description: The learning database is so small that there are not enough free entries to one of the multicast addresses of this interface.

Action: Increase the size of the learning database.

Fatal “ethsrtun”

Short Syntax: ethsrtun: unsupported command

Description: An unsupported command was given by the SRT protocol

Panic “ilnie”

Deleted: Message deleted.

Panic “ilcai”

Deleted: Message deleted.

Panic “penpe”

Deleted: Message deleted.

Panic “pecap”

Deleted: Message deleted.

Panic “npnnp”

Deleted: Message deleted.

Panic “npcap”

Deleted: Message deleted.

FDDI

Fiber Distributed Data Interface

This chapter describes Fiber Distributed Data Interface (FDDI) messages. For information about message content and how to use the message, refer to the Introduction.

FDDI.001

Level: UI–ERROR

Short Syntax: FDDI.001 *setup_phase* fld – bff unav:nt *network*

Long Syntax: FDDI.001 *setup_phase* failed, no buffer available:net *network*

Description: There were no iorbs available for an ioctl–type function such as starting the self-test or updating statistics. The net may be marked down.

FDDI.002

Level: C–INFO

Short Syntax: FDDI.002 unexp LLC hdr *DSAP SSAP control_field source_FDDI_address –> dest_FDDI_address* nt *network*

Long Syntax: FDDI.002 unexpected LLC header DSAP:*DSAP SSAP*:SSAPCTL:*control_field* from *source_FDDI_address* to *dest_FDDI_address* net *network*

Description: The LLC header has a bad control field or bad DSAP/control field combination

FDDI.003

Level: C–INFO

Short Syntax: FDDI.003 unk SNAP org code *organization_code source_FDDI_address –> dest_FDDI_address* nt *network*

Long Syntax: FDDI.003 unknown SNAP organization code *organization_code* from *source_FDDI_address* to *dest_FDDI_address* net *network*

Description: The 3 byte LLC organization code is unknown. That is, it is neither 0, an Appletalk/Apple ARP packet,Apple's OUI and an Appletalk packet, nor DEC's OUI and an Apple ARP packet.

FDDI.004

Level: C–INFO

Short Syntax: FDDI.004 unk SNAP type *SNAP_type_code source_FDDI_address –>dest_FDDI_address* nt *network*

Long Syntax: FDDI.004 unknown SNAP *type code SNAP_type_code* from *source_FDDI_address* to *dest_FDDI_address* net *network*

Description: The LLC has a bad SNAP *type code*.

FDDI.005

Level: C–INFO

Short Syntax: FDDI.005 unk DSAP *DSAP source_FDDI_address –>dest_FDDI_address* nt *network*

Long Syntax: FDDI.005 unknown DSAP *DSAP* from *source_FDDI_address* to *dest_FDDI_address* net *network*

Description: The LLC has a bad DSAP.

FDDI.006

Level: UE–ERROR

Short Syntax: FDDI.006 DN bd ln *actual_length claimed_length source_FDDI_address –> destination_FDDI_address* nt *network*

Long Syntax: FDDI.006 DECnet packet received with a bad length actual *actual_length* claimed *claimed_length* from *source_FDDI_address* to *destination_FDDI_address* network *network*

Description: A DECnet packet was received with a length field that was larger than the actual length of the packet.

FDDI.007

Level: UE–ERROR

Short Syntax: FDDI.007 No ALP for *function* nt *network*

Long Syntax: FDDI.007 *function* failed – Cannot ALP table. net *network*

Description: Need some permanent data memory and cannot get it, for example, for group addresses.

FDDI.008

Level: UE–ERROR

Short Syntax: FDDI.008 Bad SMT xmt rs:*reason_string* ln:*length –>dest_FDDI_addr* nt *network*

Long Syntax: FDDI.008 Bad SMT xmt reason *reason_string* length *length* dest *dest_FDDI_addr* net *network*

Description: SMT transmit failed due to bad length (reason 0), device overflow (2), or net down (3).

FDDI.009

Level: C–INFO

Short Syntax: FDDI.009 *Info_string* at *address* nt *network*

Long Syntax: FDDI.009 *Info_string* at *address* net *network*

Description: Information, such as BSI register values, is in Buffer memory at the given address. THE BUFFER is NOT FREED. This message should appear only when one is using the DDT debugger.

FDDI.010

Level: C–INFO

Short Syntax: FDDI.010 Bad *smt_fr_class reason_string reason_va*l fr *source_FDDI_addr* nt *network*

Long Syntax: FDDI.010 Bad *smt_fr_class* frame *reason_string reason_va*l received from *source_FDDI_addr* on net *network*

Description: Bad SMT frame received. This may well be due to an error at the sending station. This station will not use the information.

FDDI.011

Level: C–INFO

Short Syntax: FDDI.011 NSA fr *source_FDDI_addr* UNA:*UNA_FDDI_addr* nt *network*

Long Syntax: FDDI.011 NIF NSA received from *source_FDDI_addr* whose UNA is *UNA_FDDI_addr* on net *network*

Description: Not an error. A NIF NSA frame was received from this station’s upstream neighbor with a report of that station’s upstream neighbor.

FDDI.012

Level: C–INFO

Short Syntax: FDDI.012 GET rsp ln length *lw_0 lw_1 lw_2 lw_3* to *dest_FDDI_addr* nt *network*

Long Syntax: FDDI.012 GET rsp length bytes *lw_0 lw_1 lw_2 lw_3* to *dest_FDDI_addr* on net *network*

Description: Not an error. This station is sending a GET response start–ing with the 16 bytes shown.

FDDI.013

Level: C–INFO

Short Syntax: FDDI.013 SMT xmted *smt_fr_class frame_type #trans_id* to *dest_FDDI_addr_hi dest_FDDI_addr*_mid *dest_FDDI_addr*_low nt *network*

Long Syntax: FDDI.013 SMT transmitted *smt_fr_class frame_type #trans_id* to *dest_FDDI_addr_hi dest_FDDI_addr*_mid *dest_FDDI_addr*_low on net *network*

Description: Not an error. SMT is transmitting a frame. Note that self–addressed ESF announcements constitute the CNX500's maintenance packets.

FDDI.014

Level: C–INFO

Short Syntax: FDDI.014 SMT rcvd *smt_fr_class frame_type* ln *length* fm *source_FDDI_addr* nt *network*

Long Syntax: FDDI.014 SMT received a *smt_fr_class frame_type* frame len length from *source_FDDI_addr* on net *network*

Description: Not an error. SMT has received an "unusual" frame, which it cannot parse further.

FDDI.015

Level: C–INFO

Short Syntax: FDDI.015 SMT rcv *dest_FDDI_addr–>source_FDDI_addr #trans_number* nt *network*

Long Syntax: FDDI.015 SMT received a frame addressed to *dest_FDDI_addr* from *source_FDDI_addr #trans_number* on net *network*

Description: No error. SMT has received a frame.

FDDI.016

Level: UI–ERROR

Short Syntax: FDDI.016 SMT DAD rcv *dest_FDDI_addr–>source_FDDI_addr* nt *network*

Long Syntax: FDDI.016 SMT Duplicate Address Detected through frame addressed to *dest_FDDI_addr* from *source_FDDI_addr* on net *network*

Description: A big error. SMT received a frame addressed to it with the A bit set. This is a serious FDDI event. If there is another station with the same MAC address on the ring, ring initialization and frame stripping will be adversely affected. Indeed, if the ring goes down it may never recover. This station will mark the net as down, note in memory that a duplicate address was detected and NEVER come up unless the router is restarted. This ensures that the rest of the ring is not periodically harmed.

FDDI.017

Level: UI–ERROR

Short Syntax: FDDI.017 SMT Q ovf *dest_FDDI_addr–>source_FDDI_addr* nt *network*

Long Syntax: FDDI.017 SMT Queue overflow Detected through frame addressed to *dest_FDDI_addr* from *source_FDDI_addr* on net *network*

Description: An error. SMT receive queue overflowed. This frame will not be queued for SMT processing, but otherwise no action is taken.

FDDI.018

Level: C–INFO

Short Syntax: FDDI.018 Rcvd FC FC ln *length* fm *source_FDDI_addr* nt *network*

Long Syntax: FDDI.018 Received a FC FC frame len length from *source_FDDI_addr* on net *network*

Description: Not necessarily an error. Received aframe with FC neither LLC nor SMT. No further processing is done on the frame.

FDDI.019

Level: UI–ERROR

Short Syntax: FDDI.019 Mcast tbl ovf *multicast_FDDI_addr* not in nt *network*

Long Syntax: FDDI.019 Multicast table overflow with address *multicast_FDDI_addr* on net *network*

Description: A big error. Multicast table overflowed. Since the maximum number of multicast addresses needed can be determined in advance (at release time), this should never happen.

FDDI.020

Deleted: Message deleted.

FDDI.021

Deleted: Message deleted.

FDDI.022

Level: UI–ERROR

Short Syntax: FDDI.022 Pwr–Up Diag Fail: *group_no*/*test_no* CSR *CSR*

Long Syntax: FDDI.022 Power–up diagnostic failure in group *group_no*, test *test_no*, net's CSR is *CSR*.

Description: A hardware error. The on–interface–board power–up diagnostics are run as part of initialization. If they fail, this report is given, but the board will continue to come up if possible. The group/test numbers may help field service in debugging the board.

FDDI.023

Level: C–INFO

Short Syntax: FDDI.023 *state_machine old_state–>new_state* CSR *CSR*

Long Syntax: FDDI.023 Connection state machine *state_machine* went from state *old_state* to state *new_state* on the net with CSR *CSR*

Description: No error. A state machine changed state on the net with the board whose CSR is noted. Certain states are to be expected,others may not be, such as RMT's DETECTstates. Entry to these generally reflect problems in neighboring stations or the cables.

FDDI.024

Level: UE–ERROR

Short Syntax: FDDI.024 drop IPX pkt w/*encap_seen* encaps – using *encap_used* encaps on int *intnum*

Long Syntax: FDDI.024 dropped IPX pkt with encaps *encap_seen* using *encap_used* on interface *intnum*

Description: This message is generated when an IPX packet is received with an encapsulation other than that which has been selected for this interface.

Cause: Normal for networks using multiple encapsulations on a single wire.

Action: None needed.

FDDI.025

Level: UI–ERROR

***Short Syntax*:** FDDI.025 FD_IN got uni IP: *fddi_hdr_longword* nt *network*

***Long Syntax*:** FDDI.025 FDDI input got unicast IP: *fddi_hdr_longword* nt *network*

Description: This packet should not have been received; packet has been dropped.

Cause: Bad hardware, nonsense data on network.

FDDI.026

Level: UI–ERROR

***Short Syntax*:** FDDI.026 FD_IN got uni VN: *fddi_hdr_longword* nt *network*

***Long Syntax*:** FDDI.026 FDDI input got unicast Vines: *fddi_hdr_longword* nt *network*

Description: This packet should not have been received; packet has been dropped.

Cause: Bad hardware, nonsense data on network.

FDDI.027

Level: UI–ERROR

***Short Syntax*:** FDDI.027 BDG_IN got SMT: *fddi_hdr_longword* nt *network*

***Long Syntax*:** FDDI.027 Bridge input got SMT packet: *fddi_hdr_longword* nt *network*

Description: This packet should not have been received; packet has been dropped.

Cause: Bad hardware, nonsense data on network.

Panic “fddibdtbl”

***Short Syntax*:** fddibdtbl: fd_llc tbl out of date

Description: The FDDI LLC table is out of date.

Action: Contact customer service.

Panic “fddialp”

***Short Syntax*:** fddialp: Can’t allocate fddi per net structure

Description: Cannot allocate the network specific FDDI structure.

Panic “fddibprt”

Short Syntax: fddibprt: bad prot init

Description: An unsupported Network Layer protocol tried to initialize the FDDI handler.

Action: Contact customer service.

Panic “fddibreq”

Short Syntax: fddibreq: bad xmit rqst

Description: An unsupported protocol packet was given to the FDDI handler for transmission.

Action: Contact customer service.

FLT

Generic Packet Filter

This chapter describes the Generic Packet Filter facility messages. For information about message content and how to use the message, refer to the Introduction.

FLT.001

Level: UI–ERROR

Short Syntax: FLT.001 no free mem to create *structure_type*

Long Syntax: FLT.001 No free memory to create a *structure_type*

Description: This message is generated when the filtering subsystem can–not allocate the memory to hold a data structure to hold filtering information. This results in a filter not being built.

FLT.002

Level: U–TRACE

Short Syntax: FLT.002 cant apply fltr (offset *filter_offset*), pkt too shrt (ln *packet_offset*)

Long Syntax: FLT.002 Cannot apply filter (offset *filter_offset*), to pack–et of length *packet_offset*

Description: This message is generated when the maximum offset in a filter is larger than the length of a packet. The filter is not applied to the packet.

FLT.003

Level: U–TRACE

Short Syntax: FLT.003 no mem to cache pkt (max *cache_entries_allocated*)

Long Syntax: FLT.003 No memory to cache packet (maximum *cache_entries_allocated*)

Description: This message is generated if a filter is attempting to create a cache entry but cannot do so because there is no available memory on the heap. Instead, an existing entry is reused from the filter.

FLT.004

Level: C–INFO

***Short Syntax*:** FLT.004 crtng flt, sys *system_name*

***Long Syntax*:** FLT.004 Creating filter for system *system_name*

Description: A filter is being created for the router system identified by system_name

FLT.005

Level: C–INFO

***Short Syntax*:** FLT.005 flt che hit, sys *system_name*

***Long Syntax*:** FLT.005 Filter cache hit, system *system_name*

Description: A filter produced a cache hit. System_name is the system name of a filter that was previously created.

FLT.006

Level: C–INFO

***Short Syntax*:** FLT.006 flt match, sys *system_name*

***Long Syntax*:** FLT.006 Filter match, system *system_name*

Description: A filter produced a match, but with no cache hit. System_name is the system name of a filter that was previously created.

FLT.007

Level: C–INFO

***Short Syntax*:** FLT.007 flt miss, sys *system_name*

***Long Syntax*:** FLT.007 Filter miss, system *system_name*

Description: A filter was applied to a block a data, but not match was found. System_name is the system name of a filter that was previously created.

FR

Frame Relay

This chapter describes Frame Relay interface messages. For information about message content and how to use the message, refer to the Introduction.

FR.001

Level: C–INFO

Short Syntax: FR.001 Frame received, PVC = *circuit* protocol = *protocol* nt *network ID*

Long Syntax: FR.001 Frame received, PVC = *circuit* protocol type = *protocol*, on network *network ID*

Description: A LAPD frame had been received on the FR interface.

FR.002

Level: C–INFO

Short Syntax: FR.002 Frame transmitted PVC = *circuit* protocol = *protocol* nt *network ID*

Long Syntax: FR.002 Frame transmitted PVC = *circuit* protocol type = *protocol*, on network *network ID*

Description: A LAPD frame had been transmitted on the FR interface.

FR.003

Level: C–INFO

Short Syntax: FR.003 Transmit frame discarded PVC = *circuit* protocol =*protocol* nt *network ID*

Long Syntax: FR.003 Transmit frame discarded PVC = *circuit* protocol type= *protocol*, on network *network ID*

Description: A protocol frame had been discarded due to the PVC congested condition.

Cause: Protocol frames are backing up on a congested PVC.

FR.004

Level: C–INFO

Short Syntax: FR.004 Circuit outbound congestion PVC = *circuit* nt *network ID*

Long Syntax: FR.004 Circuit outbound congestion PVC = *circuit*, on network *network ID*

Description: The circuit is now experiencing congestion in the outbound direction.

FR.005

Level: C–INFO

Short Syntax: FR.005 Circuit outbound uncongested PVC = *circuit* nt *network ID*

Long Syntax: FR.005 Circuit outbound uncongested PVC = *circuit*, on network *network ID*

Description: The circuit is now not experiencing congestion in the out–bound direction.

FR.006

Level: C–INFO

Short Syntax: FR.006 Circuit active PVC = *circuit* nt *network ID*

Long Syntax: FR.006 Circuit enters active state PVC = *circuit*, on network *network ID*

Description: The circuit enters the active state.

FR.007

Level: C–INFO

Short Syntax: FR.007 Orphan circuit joins net PVC = *circuit* nt *network ID*

Long Syntax: FR.007 An orphan circuit not statically configured has joined the network PVC = *circuit*, on network *network ID*

Description: The LMI signalled present and active a circuit which had not been statically configured.

FR.008

Level: C–INFO

Short Syntax: FR.008 Circuit inactive PVC = *circuit* nt *network ID*

Long Syntax: FR.008 Circuit enters inactive state PVC = *circuit*, on network *network ID*

Description: The circuit enters the inactive state.

Cause: The remote end–point on the circuit either is down or is disabled.

FR.009

Level: C–INFO

***Short Syntax*:** FR.009 Circuit unavailable PVC = *circuit* nt *network ID*

***Long Syntax*:** FR.009 Circuit is unavailable PVC = *circuit*, on network *network ID*

Description: The circuit is no longer available on the network.

Cause: The remote end–point on the circuit either is down or is disabled.

FR.010

Level: C–INFO

***Short Syntax*:** FR.010 Circuit available PVC = *circuit* nt *network ID*

***Long Syntax*:** FR.010 Circuit is available PVC = *circuit*, on network *network ID*

Description: The circuit is now available on the network.

FR.011

Level: C–INFO

***Short Syntax*:** FR.011 LMI seq exchange requested rcv seq = *rcvseq* xmt seq =*xmtseq* nt *network ID*

***Long Syntax*:** FR.011 LMI sequence number exchange requested, last received sequence = *rcvseq* current transmit sequence = *xmtseq*, on network *network ID*

Description: A LMI sequence number exchange has been requested.

FR.012

Level: C–INFO

***Short Syntax*:** FR.012 LMI Status Enquiry requested nt *network ID*

***Long Syntax*:** FR.012 LMI Status Enquiry requested on network *network ID*

Description: A LMI full Status Enquiry has been requested.

FR.013

Level: C–INFO

***Short Syntax*:** FR.013 LMI solicited Status Enquiry response received nt *network ID*

***Long Syntax*:** FR.013 LMI solicited Status Enquiry response had been received on network *network ID*

Description: A solicited LMI Status Enquiry response has been received.

FR.014

Level: C–INFO

Short Syntax: FR.014 LMI Full Status Enquiry response received nt *network ID*

Long Syntax: FR.014 LMI Full Status Enquiry response had been received on network *network ID*

Description: A LMI full Status Enquiry response has been received.

FR.015

Level: C–INFO

Short Syntax: FR.015 Modem status change, DCD = *dcd* CTS = *cts* nt *network ID*

Long Syntax: FR.015 Modem status changed DCD = *dcd* CTS = *cts* on network *network ID*

Description: A modem status change has occurred. The present state is described.

FR.016

Level: C–INFO

Short Syntax: FR.016 Multicast frame transmitted PVC = *circuit* protocol =*protocol* nt *network ID*

Long Syntax: FR.016 Multicast frame transmitted PVC = *circuit* protocol type = *protocol*, on network *network ID*

Description: A LAPD frame had been transmitted on the FR interface.

FR.017

Level: C–INFO

Short Syntax: FR.017 Circuit remains outbound congested PVC = *circuit* nt *network ID*

Long Syntax: FR.017 Circuit remains congested in the outbound direction PVC = *circuit*, on network *network ID*

Description: The circuit is remaining in the outbound congested state to–ward the network.

FR.018

Level: C–INFO

Short Syntax: FR.018 CIR exceeded, transmit discarded PVC = *circuit* protocol = *protocol* nt *network ID*

Long Syntax: FR.018 CIR exceeded, transmit frame discarded PVC = *circuit* protocol type = *protocol*, on network *network ID*

Description: A protocol frame had been discarded due to the PVC exceeding the CIR on the circuit.

Cause: CIR monitor is enabled.

FR.019

Level: C–INFO

Short Syntax: FR.019 Orphan circuit ignored PVC = *circuit* nt *network ID*

Long Syntax: FR.019 An disallowed orphan circuit not statically configured has been ignored the network PVC = *circuit*, on network *network ID*

Description: The LMI signalled present and active a circuit which had not been statically configured or allowed.

FR.020

Level: C–INFO

Short Syntax: FR.020 Circuits exceeded, orphan circuit discarded PVC =*circuit* nt *network ID*

Long Syntax: FR.020 The total circuits allowed has been exceeded, an orphan circuit has been ignored PVC = *circuit*, on network *network ID*

Description: The LMI signalled present and active a circuit which cannot join the interface, maximum circuits have been exceeded.

FR.021

Level: C–INFO

Short Syntax: FR.021 No memory for orphan, circuit discarded PVC = *circuit* nt *network ID*

Long Syntax: FR.021 No available memory for orphan circuit, the *circuit* has been ignored PVC = *circuit*, on network *network ID*

Description: The LMI signalled present and active a circuit which cannot join the interface, memory is unavailable.

FR.022

Level: UE–ERROR

Short Syntax: FR.022 Unsupported LMI IE, type = *type* on nt *network ID*

Long Syntax: FR.022 Unsupported LMI information element, type = *type* on network *network ID*

Description: An unsupported management information element has been en–countered.

Cause: Software out of date, contact customer service.

FR.023

Level: UE–ERROR

Short Syntax: FR.023 Unsupported LMI message, type = *type* nt *network ID*

Long Syntax: FR.023 Unsupported LMI message type, type encountered = *type*, on network *network ID*

Description: An unsupported management message has been encountered.

Cause: Software out of date, contact customer service.

FR.024

Level: C–INFO

Short Syntax: FR.024 Multicast circuit joins net PVC = *circuit*, group =*group* nt *network ID*

Long Syntax: FR.024 An multicast circuit has joined the network PVC =*circuit*, in multicast group = *group* on network *network ID*

Description: The LMI signalled present and active a multicast circuit.

FR.025

Level: C–INFO

Short Syntax: FR.025 Multicast circuit leaves net PVC = *circuit*, group = *group* nt *network ID*

Long Syntax: FR.025 An multicast circuit has left the network PVC = *circuit*, from multicast group = *group* on network *network ID*

Description: The LMI signalled present and active a multicast circuit.

FR.026

Level: UE–ERROR

Short Syntax: FR.026 Unsupported NLPID, type = *type*, PVC = *circuit* on nt *network ID*

Long Syntax: FR.026 Unsupported Network Layer Protocol ID, type = *type* from PVC = *circuit* on network *network ID*

Description: An unsupported network layer protocol NLPID has been encountered.

Cause: Software out of date or incompatible, contact customer service.

FR.027

Level: UE–ERROR

Short Syntax: FR.027 Unsupported ether type, type = *type*, PVC = *circuit* on nt *network ID*

Long Syntax: FR.027 Unsupported ethernet type, type = *type* from PVC =*circuit* on network *network ID*

Description: An unsupported ethernet type has been encountered.

Cause: Software out of date or incompatible, contact customer service.

FR.028

Level: C–INFO

Short Syntax: FR.028 Unsupported multicast circuit ignored PVC = *circuit*,group = *group* nt *network ID*

Long Syntax: FR.028 An unsupported multicast circuit has been ignored,PVC = *circuit*, in multicast group = *group* on network *network ID*

Description: The LMI signalled present and active a multicast circuit that is not supported on the interface.

FR.029

Level: UE–ERROR

Short Syntax: FR.029 Received data on invalid circuit, PVC = *circuit* on nt *network ID*

Long Syntax: FR.029 Data received on invalid or nonconfigured circuit,PVC = *circuit* on network *network ID*

Description: Data has been received on a circuit not configured or learned dynamically but not yet active on network.

Cause: Network mis–configuration or mis–timing.

FR.030

Level: C–INFO

Short Syntax: FR.030 LMI seq exchange received rcv seq = *rcvseq* xmt seq =*xmtseq* nt *network ID*

Long Syntax: FR.030 LMI sequence number exchange received, last received sequence = *rcvseq* current transmit sequence = *xmtseq*, on network *network ID*

Description: A LMI sequence number exchange has been received.

FR.031

Level: C–INFO

Short Syntax: FR.031 LMI unsolicited PVC Status Update received nt *network ID*

Long Syntax: FR.031 LMI unsolicited single Status Update had been received on network *network ID*

Description: An LMI unsolicited single status update message has been received.

FR.032

Level: UE–ERROR

Short Syntax: FR.032 Circuit address length to short nt *network ID*

Long Syntax: FR.032 Circuit address length less than the 2 octet minimum received on network *network ID*

Description: A LAPD frame containing less than the 2 octet minimum ad–dress length had been encountered.

FR.033

Level: UE–ERROR

Short Syntax: FR.033 Circuit address length to large nt *network ID*

Long Syntax: FR.033 Circuit address length greater than the 2 octet maximum received on network *network ID*

Description: A LAPD frame containing greater than the 2 octet maximum ad–dress length supported had been encountered. Currently, only2 octet addressing is supported.

FR.034

Level: UE–ERROR

Short Syntax: FR.034 Circuit status message using reserved address, PVC =*circuit* nt *network ID*

Long Syntax: FR.034 Circuit status update message contained a reserved management channel address, PVC = *circuit,* on network *network ID*

Description: The LMI status message contained a reserved management channel address.

FR.035

Level: UE–ERROR

Short Syntax: FR.035 Unsupported control frame, type = *type,* PVC = *circuit* on nt *network ID*

Long Syntax: FR.035 Unsupported Link Layer control frame encountered,type = *type* from PVC = *circuit* on network *network ID*

Description: An unsupported link layer control frame encountered.

Cause: Software out of date or incompatible, contact customer ser–vice.

FR.036

Level: UE–ERROR

Short Syntax: FR.036 Unsupported management protocol descriptor, type = *type* on nt *network ID*

Long Syntax: FR.036 Unsupported layer management protocol descriptor encountered, type = *type* on network *network ID*

Description: An unsupported network layer protocol descriptor has been encountered.

Cause: Software out of date or incompatible, contact customer service.

FR.037

Level: UE–ERROR

Short Syntax: FR.037 Unsupported management call reference encountered on nt *network ID*

Long Syntax: FR.037 Unsupported layer management call reference encountered on network *network ID*

Description: An unsupported network layer call reference field has been encountered.

Cause: Software out of date or incompatible, contact customer ser–vice.

FR.038

Level: UE–ERROR

Short Syntax: FR.038 No lock shift encountered in ANSI LMI message on nt *network ID*

Long Syntax: FR.038 No lock shift encountered in received ANSI LMI message on network *network ID*

Description: The received ANSI management frame did not include required locking shift information element.

Cause: Error in network switch management frame, contact site administrator.

FR.039

Level: UE–ERROR

Short Syntax: FR.039 Incorrect formatted information element encountered on nt *network ID*

Long Syntax: FR.039 Incorrect formatted information element encountered on network *network ID*

Description: The received management frame information element was in–correctly formatted.

Cause: Error in network switch management frame, contact site administrator.

FR.040

Level: UE–ERROR

Short Syntax: FR.040 LMI rcv seq number in error seq = *rcvseq* expected seq= *xmtseq* nt *network ID*

Long Syntax: FR.040 LMI receive sequence number in error, receive sequence = *rcvseq* expected sequence = *xmtseq*, on network *network ID*

Description: An incorrect LMI receive sequence number has been received.

FR.041

Level: C–INFO

Short Syntax: FR.041 Circuit leaves net PVC = *circuit* nt *network ID*

Long Syntax: FR.041 An circuit has been removed from the network PVC =*circuit*, on network *network ID*

Description: The LMI failed to signal the mentioned circuit is last network status update message. Circuit assumed removed from network.

FR.042

Level: C–INFO

Short Syntax: FR.042 Circuit inbound congestion PVC = *circuit* nt *network ID*

Long Syntax: FR.042 Circuit inbound congestion PVC = *circuit*, on network *network ID*

Description: The circuit is now experiencing congestion in the inbound direction.

FR.043

Level: UE–ERROR

Short Syntax: FR.043 Incorrect formatted addr hdr for LMI packet encountered on nt *network ID*

Long Syntax: FR.043 Incorrect formatted address header for LMI packet en–countered on network *network ID*

Description: The address header on received management frame had BECN,FECN, DE or CR bits set.

Cause: Error in network switch management frame, contact site administrator.

FR.044

Level: UE–ERROR

Short Syntax: FR.044 Unsolicited LMI LIV received on nt *network ID*

Long Syntax: FR.044 Unsolicited LMI Link Integrity Verification received on network *network ID*

Description: LMI Link Integrity Verification message was received from the network without the router polling for it.

Cause: Duplicate packet may have been sent. Monitor LMI link and contact site administrator.

FR.045

Level: UE–ERROR

Short Syntax: FR.045 Unsolicited LMI FULL STATUS received on nt *network ID*

Long Syntax: FR.045 Unsolicited LMI FULL STATUS response received on network *network ID*

Description: LMI Full Status message was received from the network without the router polling for it.

Cause: Duplicate packet may have been sent. Monitor LMI link and contact site administrator.

Panic "frimem"

Short Syntax: Frame Relay interface initialization failed, no memory.

Description: The Frame Relay interface failed to allocate sufficient memory to complete initialization.

Action: Contact customer service.

Panic "friprt"

Short Syntax: FR: unsupported protocol during initialization

Description: The frame relay network handler detected an unsupported protocol during initialization.

Action: Contact customer service.

Panic "frfprt"

Short Syntax: FR: unsupported protocol during frame forward

Description: The frame relay network handler detected an unsupported protocol during the protocol frame forward phase.

Action: Contact customer service.

GW

Gateway

This chapter describes Gateway messages. The Gateway subsystem is the core of the router software and includes status code and the network library.For information about message content and how to use the message, refer to the Introduction.

GW.001

Level: ALWAYS

*Short Syntax***:** GW.001Copyright 1995 Digital Equipment Corporation, Copyright 1984 Massachusetts Institute of Technology, Copyright1989 The Regents of the University of California

*Long Syntax***:** GW.001Copyright 1995 Digital Equipment Corporation, Copyright 1984 Massachusetts Institute of Technology, Copyright1989 The Regents of the University of California

Description: Portions of the original code on which this system was based bear the following copyright notice: Permission to use,copy, modify, and distribute this program for any purpose and without fee is hereby granted, provided that this copy–right and permission notice appear on all copies and sup–porting documentation, the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the program without specific prior permission, and notice be given in supporting documentation that copying and distribution is by permission of M.I.T. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided “as is” without express or implied warranty. Copyright (c) 1989 The Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1.Redistributions of source code must retain the above copy–right notice, this list of conditions and the

following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by the University of California, Berkeley and its contributors. 4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

GW.002

Level: ALWAYS

Short Syntax: GW.002 *router name*, release *level* Started

Long Syntax: GW.002 *router name*, started

Description: Prints the host name of the router (as indicated in the router), and the release level of the software load which has just started in the router.

GW.003

Level: ALWAYS

*Short Syntax***:** GW.003 Unus pkt len *unused_length* nt *network ID*

*Long Syntax***:** GW.003 Unused packet length *unused_length* net *network ID*

Description: The router will not be able to send or receive the last[unused length] bytes of maximum size packets.

Cause: The configuration for the router has dictated a maximum packet size that the software will handle, which is smaller than the Maximum Transmission Unit (MTU) of the network.

Action: If the the buffer size setting on the router has been manually set, modify or remove the buffer size setting in the router. If the message persists, contact customer service.

GW.004

Level: ALWAYS

*Short Syntax***:** GW.004 Sys *queue type* q adv alloc *advisable queue length* excd actual queue length

*Long Syntax***:** GW.004 System *queue type* queue advisory allocation of exceeded *advisable queue length*

Description: The system has detected that there are probably an insufficient number of buffers for optimal operation. On startup,the maximum number of buffers allocated to either the permanent device input queue or the transient device output queue had exceeded an advisable allocation of the entire buffer pool.

Cause: The router has been configured with overly large routing tables for some protocol.

Action: Ensure that the routing tables for each protocol are of a reasonable size for the network configuration. Memory allocated to routing tables cannot be used for packet buffers.

Cause: The router in question has too many network interfaces for the amount of buffer memory available.

Action: Reduce the number of network interfaces on the router. If there are only a reasonable number of interfaces on the router, or if a (buffer) memory upgrade is available, consider expanding the amount of memory on the router. If the message persists, contact customer service.

Cause: The number of buffers has been manually set to a low number.

Action: Modify or remove the number of buffers setting in the router. If the message persists, contact customer service.

GW.005

Level: ALWAYS

Short Syntax: GW.005 Bffrs: *total created* avail *initially free* idle fair *fair share amount* low *high water mark*

Long Syntax: GW.005 Buffers: *total created* available idle fair share *initially free* low water

Description: The message gives information about the number of buffers created by the initialization procedure, as well as some in–formation on parameters used by the buffer allocation system. As long as the number of buffers currently free in the router is above the low water mark, any user can allocate buffers. Below that point, any user can allocate buffers, as long as the number is less than the 'fair share'.

GW.006

Level: C–INFO

Short Syntax: GW.006 Pkt frm nt *network ID* for uninit prt, disc

Long Syntax: GW.006 Packet from net *network ID* for uninitialized protocol discarded

Description: An incoming packet was in a protocol which, although recognized, did not have a handler loaded and enabled.

GW.007

Level: C–INFO

Short Syntax: GW.007 Ip err *error_code* nt *network ID*

Long Syntax: GW.007 Input error *error_code* net *network ID*

Description: A device input operation returned an error, along with a device specific error code. The input error counter for that network was incremented, and any packet associated with that error was probably discarded.

Action: Refer to the Router Hardware Manual under the appropriate interface to see what the specific error codes for this type of interface mean.

GW.008

Deleted: Message deleted.

GW.009

Level: UI–ERROR

*Short Syntax***:** GW.009 Nt dwn ip rstrt nt *network ID*

*Long Syntax***:** GW.009 Net down for input restart net *network ID*

Description: When the router attempted to queue additional input operations for the network, the network had been disabled for in–put.

Cause: This is caused by timing windows in the internal operation of the router; an input restart operation was requested, but when the time came to do it, input on the interface had been internally disabled. The condition is detected, and is harm–less.

Action: If the message persists, contact customer service.

GW.010

Level: UI–ERROR

*Short Syntax***:** GW.010 Ip q len *queue_length* no ip buf nt *network ID*

*Long Syntax***:** GW.010 Input queue length *queue_length* no input buffer net *network ID*

Description: When the router attempted to queue additional input operations for the network, not enough free buffers were avail–able to refill the input queue to the level desired; the actual level attained is listed.

Cause: If the message occurs on an occasional basis, a traffic peak is causing the router to run short of buffers.

Action: No action is necessary.

Cause: The router is short of buffers for some reason. This may be because there is not enough memory on the router to provide enough buffers.

Action: If there is a shortage of memory for buffers, either add memory to the router or reduce the number of network interfaces on the router. If the message persists, contact customer service.

Cause: The number of buffers may have been manually set low.

Action: Modify or remove the number of buffers setting in the router. If the message persists, contact customer service.

GW.011

Deleted: Message deleted.

GW.012

Deleted: Message deleted.

GW.013

Deleted: Message deleted.

GW.014

Level: UI–ERROR

Short Syntax: GW.014 Nt dwn op rstrt nt *network ID*

Long Syntax: GW.014 Net down for output restart net *network ID*

Description: When the router attempted to queue additional output operations for the network, the network had been disabled for output.

Cause: This is caused by timing windows in the internal operation of the router. An output restart operation was requested,but when the time came to do it, output on the interface had been internally disabled. The condition is detected, and is harmless.

Action: If the message persists, contact Customer Service.

GW.015

Deleted: Message deleted.

GW.016

Deleted: Message deleted.

GW.017

Level: UE–ERROR

Short Syntax: GW.017 Intfc hdw mssng nt *network ID*

Long Syntax: GW.017 Interface hardware missing net *network ID*

Description: When the router software went to initialize the network interface for the first time, the board did not respond on the bus. The board is marked ’not present’ and ignored from then on.

Cause: The board is not plugged in.

Action: With the power off, insert the board and restart the router.

Cause: The board is misconfigured.

Action: With the power off, and referring to the appropriate router hardware manual and configuration information, check and correct the board configuration.

Cause: The board is broken.

Action: Contact Customer Service.

GW.018

Level: U–TRACE

Short Syntax: GW.018 Strt nt slf tst nt *network ID*

Long Syntax: GW.018 Start network self test network *network ID*

Description: A network self-test (see Software Operator's Manual for more information on self-test) has been requested.

GW.019

Level: C–INFO

Short Syntax: GW.019 Slf tst nt *network ID*

Long Syntax: GW.019 Self test network *network ID*

Description: A network self-test (see Software Operator's Manual for more information on self-test) has been started.

GW.020

Level: U–TRACE

Short Syntax: GW.020 Nt pss slf tst nt *network ID*

Long Syntax: GW.020 Network passed self test network *network ID*

Description: A network undergoing self-test (see Software Operator's Manual for more information on self-test) has passed the selftest.

GW.021

Level: UE–ERROR

Short Syntax: GW.021 Nt up nt *network ID*

Long Syntax: GW.021 Network up network *network ID*

Description: After passing self-test (see Software Operator's Manual for more information on self-test), a network that was previously down has come up.

GW.022

Level: U–TRACE

Short Syntax: GW.022 Nt fld slf tst nt *network ID*

Long Syntax: GW.022 Network failed selftest network *network ID*

Description: A network undergoing self-test (see Software Operator's Manual for more information on self-test) has failed the selftest.

Cause: The exact cause is network dependant. Use any trap messages printed by the network handler, along with network specific information as revealed by the CGWCON 'Interface' command,to isolate the problem.

GW.023

Level: UE–ERROR

Short Syntax: GW.023 Nt dwn nt *network ID*

Long Syntax: GW.023 Network down network *network ID*

Description: After failing self-test (see Software Operator's Manual for more information on self-test), a network that was previously up has gone down.

GW.024

Deleted: Message deleted.

GW.025

Level: UE–ERROR

Short Syntax: GW.025 Nt fld mnt nt *network ID*

Long Syntax: GW.025 Network failed maintenance network *network ID*

Description: The number of maintenance checks (see Software Operator's Manual for more information on maintenance) failed in a given interval has exceeded the allowed limit (see the appropriate Router Hardware Manual for more detail on what the exact numbers are for each interface). A self test (see Software Operator's Manual for more information on maintenance) will be started on the interface; if it fails, the interface will be marked down.

Cause: The exact cause is network dependant. Use any trap messages printed by the network handler, along with network specific information as revealed by the CGWCON 'Interface' command,to isolate the problem. A self-test of the network may re–veal additional information.

GW.026

Level: C–TRACE

Short Syntax: GW.026 Mnt nt *network ID*

Long Syntax: GW.026 maintenance network *network ID*

Description: A maintenance check (see Software Operator's Manual for more information on maintenance) has been started for the indicated interface.

GW.027

Level: CI–ERROR

Short Syntax: GW.027 No pkt fr mnt nt *network ID*

Long Syntax: GW.027 No packet for maintenance network *network ID*

Description: A buffer could not be allocated when needed by network maintenance.

Cause: This may be caused by temporary traffic loads. Many other causes are possible.

Action: If the message persists, contact customer service.

GW.028

Level: U–INFO

Short Syntax: GW.028 Snk dsc pkt prt *protocol* to *next_hop_host*

Long Syntax: GW.028 Sink network discarding packet protocol *protocol* to host *next_hop_host*

Description: A buffer was sent to the sink network, which discarded it with no indication of error to the forwarder.

GW.029

Level: U–INFO

Short Syntax: GW.029 Int dis nt *network ID*

Long Syntax: GW.029 Interface disabled in configuration net *network ID*

Description: The interface in question was disabled in the configuration and will not come up; it can be started at any time by test–ing it.

GW.030

Level: U–INFO

Short Syntax: GW.030 *heap_bytes* bytes reserved by *subsystem*

Long Syntax: GW.030 *heap_bytes* bytes of heap reserved by subsystem *subsystem*

Description: At start–up time, one of the router's subsystems has reserved so many bytes of heap memory. This will be subtracted from the router's free memory before the remainder is carved into packet buffers.

GW.031

Level: ALWAYS

Short Syntax: GW.031 IP q alloc fl nt *network ID* avl *number of buffers*

Long Syntax: GW.031 Input queue allocation failed net *network ID* available

Description: The system has detected that there are probably an insufficient number of buffers for optimal operation. On startup,each of the fast devices are allocated a fixed number of buffers. If these buffers are not available, the particular interface may not perform well.

GW.032

Deleted: Message deleted.

GW.033

Deleted: Message deleted.

GW.034

Deleted: Message deleted.

GW.035

Level: UI–ERROR

Short Syntax: GW.035 Nt dwn to hst *next_hop_host* nt *network ID*

Long Syntax: GW.035 Net down transmitting to host *next_hop_host* net *network ID*

Description: When the router went to send a packet to a given host, the network interface it had been told to send the packet over to was not up. The output discard counter for that network was incremented, and the packet was returned to the protocol forwarder for attention. Usually, *next_hop_host* will be the MAC layer address of the next hop router or host that this packet is being sent to. The format of this MAC address depends on the network type. If *next_hop_host* starts with an @ (at–sign), then that network does not provide a format–ted display of MAC addresses, and the following number is the hex address in router memory that the next hop host ad–dress is stored at.

Cause: This is caused by timing windows in the internal operation of the router; a packet was queued for output, but when the time came to send it, the interface was down. The condition is detected, and is harmless.

Action: If the message persists, contact customer service.

GW.036

Level: U–INFO

Short Syntax: GW.036 Op ovfl to hst *next_hop_host* nt *network ID*

Long Syntax: GW.036 Output overflow when transmitting to host *next_hop_host* net *network ID*

Description: When the router went to send a packet to a given host, the network output queue was too full, and the packet had to be discarded. The output overflow counter for that network was incremented, and the packet was returned to the protocol forwarder for attention. Usually, *next_hop_host* will be the MAC layer address of the next hop router or host that this packet is being sent to. The format of this MAC address depends on the network type. If *next_hop_host* starts with an @ (at–sign), then that network does not provide a format–ted display of MAC addresses, and the following number is the hex address in router memory that the next hop host ad–dress is stored at.

Cause: This is caused by the offered load in the network being higher than the bandwidth available in the output network. Since the router itself is keeping up with the traffic,there is little it can do; the hosts generating the traffic are simply sending more data than the output network can accommodate.

Action: Increase the speed of the network in question (particularly if it is a slow speed leased line), or take measures to restrict the offered load.

GW.037

Level: C–INFO

Short Syntax: GW.037 Nt dwn, disc pkt to hst *next_hop_host* nt *network ID*

Long Syntax: GW.037 Network down, discarding packet to host *next_hop_host* network *network ID*

Description: Packets waiting for transmission on the network in question were discarded when the network went down. The discard counter for the network in question is incremented. Usually, *next_hop_host* will be the MAC layer address of the next hop router or host that this packet is being sent to. The format of this MAC address depends on the network type. If *next_hop_host* starts with an @ (at–sign), then that network does not provide a formatted display of MAC addresses, and the following number is the hex address in router memory that the next hop host address is stored at.

GW.038

Level: C–INFO

Short Syntax: GW.038 User *default name* has logged on

Long Syntax: GW.038 User *default name* has logged on

Description: A new user has logged on to the system.

GW.039

Level: CE–ERROR

Short Syntax: GW.039 Failed logon: ID = *default name*

Long Syntax: GW.039 A logon attempt has failed: user ID = *default name*

Description: Someone attempted to log onto the system but did not supply a correct user–name and password.

GW.040

Level: C–INFO

Short Syntax: GW.040 ot cl dnd nt *network ID*

Long Syntax: GW.040 Outbound calls denied network *network ID*

Description: Router would like to place outbound call, but configuration prevents it.

GW.041

Level: C–INFO

Short Syntax: GW.041 in cl dnd nt *network ID*

Long Syntax: GW.041 Inbound calls denied network *network ID*

Description: Router rejected incoming call because configuration doesn't allow inbound calls.

GW.042

Level: C–INFO

Short Syntax: GW.042 in cl unk addr "*dial_address,/subdial_address,*" rj nt *network ID*

Long Syntax: GW.042 Inbound call from unknown address"*dial_address,/subdial_address,*" rejected, network *network ID*

Description: An inbound call was not accepted because the caller's ad–dress didn't match our configured remote address

. GW.043

Level: C–TRACE

Short Syntax: GW.043 CML state *state_string,*, event *event_string,* nt *network ID*

Long Syntax: GW.043 CML state *state_string,*, event *event_string,*, net *network ID*

Description: FSM trace event.

GW.044

Level: UI–ERROR

Short Syntax: GW.044 No cnfg nt *network ID*

Long Syntax: GW.044 No configuration found for net *network ID*

Description: No SR_VRTBLK record found in SR_VNET block.

GW.045

Level: UI–ERROR

Short Syntax: GW.045 bd dl net on nt *network ID*

Long Syntax: GW.045 Bad dial network specified in config, net *network ID*

Description: The dialer net configured is either not present, or not a dial net.

GW.046

Level: UI–ERROR

Short Syntax: GW.046 bd dl dst on nt *network ID*

Long Syntax: GW.046 Bad dialer destination name specified in config, net *network ID*

Description: The specified destination name was not added with the “add address” command.

GW.047

Level: C–INFO

Short Syntax: GW.047 idle exp nt *network ID*

Long Syntax: GW.047 idle timer expired and call cleared, net *network ID*

Description: The idle timer of a demand–based net expired, and the call was cleared.

GW.048

Level: U–INFO

Short Syntax: GW.048 Int rsvd for rst nt *network ID*

Long Syntax: GW.048 Interface reserved for WAN restoral in configuration net *network ID*

Description: The interface in question has been reserved for WAN restoral in the configuration and will not come up until needed by the WAN restoral process.

GW.049

Level: U–INFO

Short Syntax: GW.049 Patched *variable_name* to *new_value*

Long Syntax: GW.049 Variable *variable_name* has been patched to value *new_value*

Description: The user has patched the value of one of the router's data items accordingly.

GW.050

Level: U–INFO

Short Syntax: GW.050 Patch to *variable_name* failed

Long Syntax: GW.050 Attempt to patch variable *variable_name* has failed

Description: The user has attempted to patch the value of one of the router's data items. The patch failed.

GW.051

Level: UI_ERROR

Short Syntax: GW.051 Wrt SRAM failed blk *block_num*, typ *rec_type_num*

Long Syntax: GW.051 Attempt to write block *block_num*,, record type *rec_type_num* to SRAM has failed

Description: Some code which cannot put out a console message (typically during startup) tried to write SRAM and failed.

GW.052

Level: UI_ERROR

Short Syntax: GW.052 No UDP port avail to sync time

Long Syntax: GW.052 No UDP port available to send time sync request

Description: udp_notify returned 0. Probably, IP is not configured.

GW.053

Level: UI_ERROR

Short Syntax: GW.053 No UDP port avail to srvc time req

Long Syntax: GW.053 No UDP port available to receive time sync requests

Description: udp_notify returned 0. Probably, IP is not configured, or else software error.

GW.054

Level: U–INFO

*Short Syntax***:** GW.054 Ip ovfl nt *network ID*, *count* pkts disc

*Long Syntax***:** GW.054 Input overflow net *network ID*, *count* packets discarded

Description: Packets are arriving on the stated interface too quickly for the router's forwarders to process them; they are discarded before being examined by the router software because of the overload. The count of packets is the number of packets this has happened to since the last time it was attempted to log this message. The input overflow counter for this *network ID* is incremented.

Cause: This may sometimes be caused by ''broadcast storms'', which are network events caused by combinations of buggy and/or out–of–date software running on network hosts which spread in a chain reaction, typically causing the network to be consumed with back to back packets (often broadcast) for a period of seconds, or occasionally, a minute or two.

Action: If a broadcast storm is happening, fix or disable the responsible hosts.

Cause: It may be simply caused by very heavy load.

Action: If heavy load is the cause, and this message happens frequently, you may be using one of the slower routers in the product line. If there is a faster CPU option available for the router you are using, consider upgrading.

GW.055

Level: UI–ERROR

*Short Syntax***:** GW.055 Nt dwn trans on nt *network ID*

*Long Syntax***:** GW.055 Net down transmitting on net *network ID*

Description: When the router went to send a packet, the network interface it had been told to send the packet over to was not up. The output discard counter for that network was incremented, and the packet was returned to the protocol forwarder for attention.

Cause: This is caused by timing windows in the internal operation of the router; a packet was queued for output, but when the time came to send it, the interface was down. The condition is detected, and is harmless.

Action: If the message persists, contact customer service.

GW.056

Level: UI–ERROR

Short Syntax: GW.056 Nt out dis trans on nt *network ID*

Long Syntax: GW.056 Net output disabled, transmitting on net *network ID*

Description: When the router went to send a packet, the network interface it had been told to send the packet over had packet transmission disabled. The output discard counter for that network was incremented.

Cause: This is caused by timing windows in the internal operation of the router; a packet was queued for output, but when the time came to send it, output on the interface was disabled. The condition is detected, and is harmless.

Action: If the message persists, contact customer service.

GW.057

Level: U–INFO

Short Syntax: GW.057 Op ovfl nt *network ID*

Long Syntax: GW.057 Output overflow when transmitting on net *network ID*

Description: When the router went to send a packet, the network output queue was too full, and the packet had to be discarded. The output overflow counter for that network was incremented, and the packet was returned to the protocol forwarder for attention.

Cause: This is caused by the offered load in the network being higher than the bandwidth available in the output network. Since the router itself is keeping up with the traffic, there is little it can do; the hosts generating the traffic are simply sending more data than the output network can accommodate.

Action: Increase the speed of the network in question (particularly if it is a slow speed leased line), or take measures to restrict the offered load.

GW.058

Level: U–INFO

Short Syntax: GW.058 Op err hst *next_hop_host* nt *network ID*

Long Syntax: GW.058 Output error transmitting to host *next_hop_host* net *network ID*

Description: A packet has not been successfully retransmitted. The output error counter for that network is incremented, and the packet is discarded. Usually, *next_hop_host* will be the MAC layer address of the next hop router or host that this packet is being sent to. The format of this MAC address depends on the network type. If *next_hop_host* starts with an @ (at–sign), then that network does not provide formatted display of MAC addresses, and the following number is the hex address in router memory that the next hop host address is stored at.

Cause: If this message occurs more than very rarely, it probably indicates hardware transmission problems on the network in question.

Action: Utilize appropriate level 2 network management tools such as Tokenview (for rings) or a Time Domain Reflectometer (for Ethernet) to isolate and fix the problem.

GW.059

Level: C–INFO

Short Syntax: GW.059 Alloc buff with min *global_buffers* global, *private_buffers* per net

Long Syntax: GW.059 Allocating buffers with minimum of *global_buffers* global buffers, and *private_buffers* buffers per fast input network

Description: The router is going to do the buffer allocation with the specified constraints.

Cause: This is normal on router startup.

GW.060

Level: C–INFO

Short Syntax: GW.060 Buffs alloc with reduction *reduction*

Long Syntax: GW.060 Buffers allocated with reduction by *reduction* of private buffers

Description: The router has completed the buffer allocations. If the input networks could not get all the buffers that were requested, the reduction will be non–zero.

Cause: This message always happens on startup of the router. However, a non–zero reduction indicates that the router is close to being short on buffer memory. The higher the reduction, the more severe the buffer memory shortage. However, the shortage is not so severe that the router will not operate, but performance may be impaired.

Action: Upgrade size of buffer memory. Choose smaller buffer size on those devices (Token–Ring, Serial Line) where that is configurable.

GW.061

Level:	UI–ERROR
Short Syntax:	GW.061 Priv buff alloc failed, nt *network ID*
Long Syntax:	GW.061 Private buffer allocation failed, network *network ID*
Description:	The buffer allocation for a private buffer for the specified network failed. This network will have one less buffer than was intended. This message is severe only of it happens many times.
Cause:	Shortage of buffer memory. (Particularly if preceded by ELS message GW.064.)
Action:	Upgrade size of buffer memory.
Action:	Choose smaller buffer size on those devices (Token–Ring, Serial Line) where that is configurable.
Cause:	Shortage of heap memory. (Particularly if preceded by ELS message GW.063.)
Action:	Reduce routing table sizes. Increase size of data memory.
Cause:	Buffer allocation routine did not accurately predict how many buffers could be allocated.
Action:	On some configurations, some portions of the buffer memory are unusable. The preallocator does not take this into account, so a few buffer allocations may fail.

GW.062

Level: UI–ERROR

Short Syntax: GW.062 Global buff alloc failed after *count*

Long Syntax: GW.062 Global buffer allocation failed after *count* allocated

Description: The buffer allocation for a global buffer failed. The router will have one less global buffer than was intended. This message is severe only of it happens many times, starting at low values of count.

Cause: Shortage of buffer memory. (Particularly if preceded by ELS message GW.064.)

Action: Upgrade size of buffer memory.

Action: Choose smaller buffer size on those devices (Token–Ring, Serial Line) where that is configurable.

Cause: Shortage of heap memory. (Particularly if preceded by ELS message GW.063.)

Action: Reduce routing table sizes. Increase size of data memory.

Cause: Buffer allocation routine did not accurately predict how many buffers could be allocated.

Action: On some configurations, some portions of the buffer memory are unusable. The preallocator does not take this into account, so a few buffer allocations may fail.

GW.063

Level: UI–ERROR

Short Syntax: GW.063 Alloc of iorb failed

Long Syntax: GW.063 Allocation of I/O request block failed

Description: Some code in the router was allocating an I/O request block and buffer. The allocation of the I/O request block failed.

Cause: Shortage of heap memory.

Action: Reduce routing table sizes. Increase size of data memory.

GW.064

Level: UI–ERROR

Short Syntax: GW.064 Alloc of buffer failed

Long Syntax: GW.064 Allocation of buffer failed

Description: Some code in the router was allocating an I/O request block and buffer. The allocation of the buffer failed.

Cause: Shortage of buffer memory.

Action: Upgrade size of buffer memory.

Action: Choose smaller buffer size on those devices (Token–Ring, Serial Line) where that is configurable.

GW.065

Level: U–INFO

Short Syntax: GW.065 *heap_bytes* bytes buff reserved by *subsystem*

Long Syntax: GW.065 *heap_bytes* bytes of buffer memory reserved by *subsystem* subsystem

Description: At start–up time, one of the router's subsystems has reserved so many bytes of buffer memory. This will be subtracted from the router's free buffer memory before the remainder is carved into packet buffers.

GW.066

Level: UI_ERROR

Short Syntax: GW.066 LID no bf, *message_type*, not snt nt *network ID*

Long Syntax: GW.066 LID no buffer, *message_type*, msg not sent on net *network ID*

Description: Line ID code couldn't allocate a buffer to send a message.

GW.067

Level: UE_ERROR

Short Syntax: GW.067 LID NAK rcv nt *network ID*

Long Syntax: GW.067 LID NAK received net *network ID*

Description: The other end of the switched circuit didn't like the LINE ID we sent, and returned a NAK.

Action: Check configuration on both sides. Remote side does not think we should be calling it.

GW.068

Level: C–INFO

Short Syntax: GW.068 LID ACK rcv nt *network ID*

Long Syntax: GW.068 LID ACK received net *network ID*

Description: The other end of the switched circuit liked our line ID.

GW.069

Level: UE_ERROR

Short Syntax: GW.069 LID tmo on mdm sgs nt *network ID*

Long Syntax: GW.069 LID timeout waiting for modem signals to come up on net *network ID*

Description: Either an inbound or outbound call, the V.25*bis* modem signals did not come up after the call was connected.

Action: Check line and modems. Line quality may be insufficient.

GW.070

Level: UE_ERROR

Short Syntax: GW.070 LID tmo on id nt *network ID*

Long Syntax: GW.070 LID timeout waiting for line ID from other side, net *network ID*

Description: Timed out waiting for line ID from remote side.

Action: Check configuration of whoever is calling into this router. They are not sending line ID message. Might be an unrecognized router.

GW.071

Level: UE_ERROR

Short Syntax: GW.071 LID unkn id [*bad_lineid_string,*]; nk snt, nt *network ID*

Long Syntax: GW.071 LID unknown line ID [*bad_lineid_string,*] received; NAK sent, net *network ID*

Description: Timed out waiting for line ID from remote side.

Action: Check configuration of whoever is calling into this router. They are not sending line ID message. Might be an unrecognized router.

GW.072

Level: UE_ERROR

Short Syntax: GW.072 LID no dflt circt; data ign nt *network ID*

Long Syntax: GW.072 LID no default circuit; received data was ignored, net *network ID*

Description: Received data from other side rather than line ID, but had no default circuit to assign the data to.

Action: Check configuration of whoever is calling into this router. They are not sending line ID message. Might be an unrecognized router.

GW.073

Level: UI_ERROR

Short Syntax: GW.073 Rcv buffs increased to *configured_buffers*, exceeds max of *maximum_buffers*, nt *network ID*

Long Syntax: GW.073 Receive buffers increased to *configured_buffers*, exceeds maximum of *maximum_buffers*, net *network ID*

Description: The user–configured number of receive buffers exceeds the maximum allowed for this interface type. The number of buffers will be reduced to the maximum.

Cause: Excessive number of buffers in Config>SET RECEIVE–BUFFERS command.

Action: Configure for acceptable number of buffers.

GW.074

Level: C_INFO

Short Syntax: GW.074 Rcv buffs increased from *configured_buffers* to *default_buffers*, nt *network ID*

Long Syntax: GW.074 Receive buffers increased from *configured_buffers* to *default_buffers*, net *network ID*

Description: The user configuration is increasing the number of receive buffers on this interface from the default to the configured value.

GW.075

Level: U_INFO

Short Syntax: GW.075 Rcv buffs decreased from *default_buffers* to *configured_buffers*, nt *network ID*

Long Syntax: GW.075 Receive buffers decreased from *default_buffers* to *configured_buffers*, net *network ID*

Description: The user configuration is decreasing the number of receive buffers on this interface from the default to the configured value. This may reduce performance on this interface.

GW.076

Level: UI–ERROR

Short Syntax: GW.076 Swcthd net (*switched network ID*) rjctd rgstrtn for nt *network ID*

Long Syntax: GW.076 The switched network (network *switched network ID*) rejected the registration request for this dial circuit: net *%3* interface *network ID*/

Description: The dial circuit is misconfigured.

GW.077

Level: C–INFO

Short Syntax: GW.077 No dl crct inc call on nt *switched network ID*

Long Syntax: GW.077 No dial circuit configured for inbound calls on switched network *switched network ID*

Description: An inbound call was received over the switched network, and there isn't a dial circuit configured to take it.

Cause: Misconfiguration.

Action: A dial circuit needs to be configured to accept inbound calls.

Cause: Wrong number.

Action: If this persists, you may want to pursue what avenues you can to identify a possible security break–in.

Panic "gwbadhd"

Short Syntax: GW: Bd cnf inf nt hdr lngths

Description: Bad configuration information in the load was detected.

Cause: Hand–configured maximum header and trailer sizes are smaller than the actual lengths of at least one network in the router.

Action: Contact Customer Service.

Panic “gwbdntv”

Short Syntax: GW: incompatible net table vers

Description: A load with incompatible versions of binary modules has been detected.

Cause: The version number on the network configuration table does not match that which the code was compiled with.

Action: Contact Customer Service.

Panic “gwbdpm”

Short Syntax: GW: incompatible P_MAX

Description: A load with incompatible versions of binary modules has been detected.

Cause: The maximum number of protocols in the configuration information does not match that which the code was compiled with.

Action: Contact Customer Service.

Panic “gwbdtm”

Short Syntax: GW: incompatible T_MAX

Description: A load with incompatible versions of binary modules has been detected.

Cause: The maximum number of network types in the configuration information does not match that which the code was compiled with.

Action: Contact Customer Service.

Panic “gwbdim”

Short Syntax: GW: incompatible I_MAX

Description: A load with incompatible versions of binary modules has been detected.

Cause: The maximum number of interface types in the configuration information does not match that which the code was compiled with.

Action: Contact Customer Service.

Panic “gwnmp”

Short Syntax: GW: no mem for prot tbl

Description: No memory was available for a critical system table.

Cause: Insufficient memory was available to allocate either the installed or complete protocol table, or the per network protocol upcalls, early in initialization.

Action: Contact Customer Service.

Panic “gwfrfr”

Short Syntax: GW: freeing free buffer

Description: The buffer free routine detected software in the system attempting to free a buffer that has already been freed.

Cause: Software bug that frees the same buffer twice, which is a grave error.

Action: Take dump of this failure, and send it to Customer Service.

Panic “gwgtgt”

Short Syntax: GW: alloc busy buffer

Description: The buffer free routine detected software in the system attempting to allocate a buffer that is already busy.

Cause: Software bug.

Action: Take dump of this failure, and send it to Customer Service.

Panic “gwifdrv”

Short Syntax: GW: net with multiple i_fdrv requests

Description: The buffer allocation routine encountered a network that wanted more than one type of memory per buffer.

Cause: Software bug.

Action: Take dump of this failure, and send it to Customer Service.

Panic “gwlgwc”

Short Syntax: GW: leading buffer guard word corrupted

Description: The code that monitors the packet buffers detected that the guard word in front of a buffer has been corrupted.

Cause: Software bug.

Cause: Hardware failure.

Action: Take dump of this failure, and send it to Customer Service.

Panic “gwtgwc”

Short Syntax: GW: trailing buffer guard word corrupted

Description: The code that monitors the packet buffers detected that the guard word after the end of a buffer has been corrupted.

Cause: Software bug.

Cause: Hardware failure.

Action: Take dump of this failure, and send it to Customer Service.

Panic “gwnhifdrv”

Short Syntax: GW: no heap mem for i_fdrv

Description: No heap memory available for buffer cache data block.

Cause: Shortage of heap memory.

Action: Reduce routing table sizes. Increase size of data memory.

Action: Take dump of this failure, and send it to Customer Service.

Panic “gwnbifdrv”

Short Syntax: GW: no buff mem for i_fdrv

Description: No buffer memory available for buffer cache data block.

Cause: Shortage of buffer memory.

Action: Upgrade size of buffer memory.

Action: Choose smaller buffer size on those devices (Token–Ring, Serial Line) where that is configureable.

Action: Take dump of this failure, and send it to Customer Service.

Fatal “gwtfb”

Short Syntax: GW: too little buffer memory

Description: The buffer allocation code simply cannot allocate enough input buffers to each network while still leaving a reasonable number of free buffers for the routing protocols. (These constraints are given by ELS message GW.059.)

Cause: Too many devices, or too large a buffer size, for the available amount of buffer memory.

Action: Deconfigure (or remove) some devices. Upgrade size of buffer memory. Choose smaller buffer size on those devices (Token–Ring, Serial Line) where that is configureable.

ICMP

Internet Control Message Protocol

This chapter describes Internet Control Message Protocol (ICMP) messages. ICMP is part of the IP family. For information about message content and how to use the message, refer to the Introduction. The following table lists the message codes that are unique to ICMP messages. The **type** is the *ICMP_type* in the ELS messages, the **Subtype** is the *ICMP_code* in the ELS messages.

Type (Subtype)	Meaning
0	Echo reply
3	Destination unreachable
3 (0)	Network unreachable
3 (1)	Host unreachable
3 (2)	Protocol unreachable
3 (3)	Port unreachable
3 (4)	Fragmentation needed
3 (5)	Source route failed
4	Packets sent too often
5	Redirect packet
5 (0)	Redirect to network
5 (1)	Redirect to host
5 (2)	Redirect to TOS and network
5 (3)	Redirect to TOS and host
8	Echo
11	Time–to–live exceeded
11 (0)	Exceeded in transit
11 (1)	Exceeded in fragment reassembly
12	Parameter problem
13	Timestamp
14	Timestamp reply
15	Information request
16	Information request reply

ICMP.001

Level: UE–ERROR

Short Syntax: ICMP.001 bd cks 0x*received_checksum* (exp 0x*good_checksum*) *source_IP_address –> destination_IP_address*

Long Syntax: ICMP.001 bad ICMP checksum 0x*received_checksum* received (expected 0x*good_checksum*) in packet from *source_IP_address* to *destination_IP_address*

Description: A bad ICMP checksum was detected in an incoming ICMP message. The received checksum is displayed, together with the value that the checksum should have had. The received packet is discarded.

Cause: This is probably caused by an error in the source host.

Action: Contact the manufacturer of the source host and report the problem.

ICMP.002

Level: C–INFO

Short Syntax: ICMP.002 ech *source_IP_address –> destination_IP_address*

Long Syntax: ICMP.002 echo request packet received from *source_IP_address* to *destination_IP_address*

Description: An ICMP Echo Request was received from the source host by the router. Note that an ICMP Echo Request is only recognized when it comes in over the interface with the address the packet is addressed to.

ICMP.003

Level: U–INFO

Short Syntax: ICMP.003 ech rp *source_IP_address –> destination_IP_address*

Long Syntax: ICMP.003 echo reply packet received from *source_IP_address* to *destination_IP_address*

Description: An ICMP Echo Reply was received from the source host by the router. This is a slightly suspicious event, since the router does not normally send ICMP Echo Requests.

ICMP.004

Level: CI–ERROR

Short Syntax: ICMP.004 unhnd typ *ICMP_type ICMP_code source_IP_address –> destination_IP_address*

Long Syntax: ICMP.004 unhandled message type *ICMP_type ICMP_code* from *source_IP_address* to *destination_IP_address*

Description: An ICMP message came in with a type that the router software does not handle.

ICMP.005

Level: U–TRACE

Short Syntax: ICMP.005 unhnd brd typ *ICMP_type ICMP_code source_IP_address –> destination_IP_address*

Long Syntax: ICMP.005 unhandled broadcast message type *ICMP_type ICMP_code* from *source_IP_address* to *destination_IP_address*

Description: A broadcast ICMP message came in with a type that the router software does not handle.

ICMP.006

Level: UE–ERROR

Short Syntax: ICMP.006 bd typ *ICMP_type ICMP_code source_IP_address –> destination_IP_address*

Long Syntax: ICMP.006 bad message type *ICMP_type ICMP_code* from *source_IP_address* to *destination_IP_address*

Description: An ICMP message came in with a type that is not legal.

ICMP.007

Level: C–INFO

Short Syntax: ICMP.007 addr msk *source_IP_address –> destination_IP_address*

Long Syntax: ICMP.007 address mask request received from *source_IP_address* to *destination_IP_address*

Description: An ICMP Address Mask Request was received from the source host by the router.

ICMP.008

Level: C–TRACE

Short Syntax: ICMP.008 addr msk rep *source_IP_address –> destination_IP_address*

Long Syntax: ICMP.008 addres mask reply received from *source_IP_address* to *destination_IP_address*

Description: An ICMP Address Mask Reply was received from the source host by the router.

ICMP.009

Level: UI–ERROR

Short Syntax: ICMP.009 no pkt or mem

Long Syntax: ICMP.009 heap memory or packet buffer not available

Description: Internal resources in the router necessary to reply to the incoming message were unavailable.

ICMP.010

Level: UE–ERROR

Short Syntax: ICMP.010 amb addr msk *source_IP_address –> destination_IP_address*

Long Syntax: ICMP.010 ambiguous address mask request received from *source_IP_address* to *destination_IP_address*

Description: An incoming address mask request on an interface which contained more than one IP source address contained a destination address which could not be localized to one of the addresses, so no reply could be generated.

ICMP.011

Level: UI–ERROR

Short Syntax: ICMP.011 err *code* sndng pkt to nt *network ID*

Long Syntax: ICMP.011 error *code* sending packet to net *network ID*

Description: An outgoing reply packet was dropped as the result of some problem in the router.

Cause: There are many potential causes of this problem; an overloaded output queue, a down network, etc.

Action: Consult logging output from the relevant *network* subsystem for more information.

ICMP.012

Level: C–INFO

Short Syntax: ICMP.012 rdr *source_IP_address –> destination_IP_address* to *new_next_hop_IP_address*

Long Syntax: ICMP.012 sending redirect for packet from *source_IP_address* to *destination_IP_address* to use router *new_next_hop_IP_address*

Description: The router is sending an ICMP Redirect, advising a source host on a directly connected network that there is a better f

ICMP.013

Level: U–INFO

Short Syntax: ICMP.013 bd prm off *problem_offset source_IP_address –> destination_IP_address*

Long Syntax: ICMP.013 sending parameter problem message problem offset *problem_offset* for packet from *source_IP_address* to *destination_IP_address*

Description: The router is sending an ICMP Parameter Problem message, for an unspecified problem at the given offset.

ICMP.014

Level: U–TRACE

Short Syntax: ICMP.014 snd *ICMP_type ICMP_code* pkt *source_IP_address –> destination_IP_address*

Long Syntax: ICMP.014 sending packet type *ICMP_type code ICMP_code* for packet from *source_IP_address* to *destination_IP_address*

Description: The router is sending an ICMP packet of the specified type about a packet from the source host to the destination.

ICMP.015

Level: UE–ERROR

Short Syntax: ICMP.015 shrt ICMP hdr *header_length* src *source_ip_address*

Long Syntax: ICMP.015 short ICMP packet *header_length* received in packet from *source_ip_address*

Description: This message is generated when an ICMP packet's indicated header length is below the minimum possible length for an ICMP packet.

Cause: Most likely, this is a damaged packet. It may be that another node is building an incorrect header.

Action: If the problem persists, examine a line trace to determine where the packet is being damaged.

ICMP.016

Level: U–TRACE

Short Syntax: ICMP.016 *current_next_hop* rdr dest *IP_destination* to *better_next_hop*

Long Syntax: ICMP.016 *current_next_hop* has redirected traffic for *IP_destination* to *better_next_hop*

Description: A redirect has been received, changing the next hop for the given destination. Redirects are only processed when running in host mode.

ICMP.017

Level: UE–ERROR

Short Syntax: ICMP.017 Bad rdr from *gateway_address*, rsn: *reason*

Long Syntax: ICMP.017 Redirect received from *gateway_address* was bad for the reason: *reason*

Description: A redirect was received from a router, but rejected for the specified reason.

ICMP.018

Level: U–TRACE

Short Syntax: ICMP.018 Router advertisement received from *router_address*

Long Syntax: ICMP.018 Router advertisement received from *router_address*

Description: An ICMP Router Advertisement (Gateway Discovery) message has been received from the specified router.

ICMP.019

Level: UE–ERROR

Short Syntax: ICMP.019 Bad router adv from *gateway_address*, rsn: **reason**

Long Syntax: ICMP.019 Router advertisement received from *gateway_address* was bad for the reason: **reason**

Description: An ICMP Router Advertisement (Gateway Discovery) message has been received from the specified router, but was rejected for the specified reason.

ICMP.020

Level: U–INFO

Short Syntax: ICMP.020 rcvd typ *ICMP_type ICMP_code source_IP_address –> destination_IP_address*

Long Syntax: ICMP.020 received message type *ICMP_type ICMP_code* from *source_IP_address* to *destination_IP_address*

Description: The router has received an ICMP message of the specified type from the source host.

IP

Internet Protocol

This chapter describes Internet Protocol (IP) messages. For information about message content and how to use the message, refer to the Introduction.

IP.001

Level: U–INFO

Short Syntax: IP.001 q ovrf *source_ip_address –> destination_ip_address* nt *network ID*

Long Syntax: IP.001 Queue overflow on packet from *source_ip_address* for *destination_ip_address* from net *network ID*

Description: This message is generated when the forwarder must discard a packet that was not forwarded via the IP cache because of an input queue overflow. Note that this event does not get counted in ELS, it is instead counted in the IP console. The counters (kept per input network) can be read using the IP>COUNTERS command.

Cause: Input queue overflows happen when a packet is received from an interface that is short on buffers, the destination is not in the IP cache, and the length of the IP queue is greater than the fair share. This may be caused by either a burst or steady state of traffic arriving faster than the IP forwarder can forward it.

Action: Reduce traffic bursts. Upgrade to a faster router.

Cause: Excessive IP routing cache misses, causing most IP packets to go through the cache miss forwarder.

Action: Increase the size of the IP cache.

IP.002

Level: UE–ERROR

Short Syntax: IP.002 not V4 hdr *version_number* nt *network ID*

Long Syntax: IP.002 Not version 4 header (*version_number*) in packet from net *network ID*

Description: This message is generated when a packet has an incorrect version number.

Cause: Most likely, this packet was damaged since there should be no other versions of IP running.

Action: If the problem persists, examine a line trace to determine where the packet is being damaged.

IP.003

Level: UE–ERROR

Short Syntax: IP.003 shrt hdr *header_length* pkt ln *packet_length* nt *network ID*

Long Syntax: IP.003 Header too short (*header_length* bytes) in *packet_length* byte packet from net *network ID*

Description: This message is generated when a packet's indicated header length is below the minimum possible length.

Cause: Most likely, this is a damaged packet. It may be that another node is building an incorrect header.

Action: If the problem persists, examine a line trace to determine where the packet is being damaged.

IP.004

Level: UE–ERROR

Short Syntax: IP.004 bd hdr cks 0x*checksum* (exp 0x*expected_checksum*) *source_ip_address* –> *destination_ip_address*

Long Syntax: IP.004 Bad header checksum 0xchecksum (expected 0x*expected_checksum*) in packet from *source_ip_address* for *destination_ip_address*

Description: This message is generated when a packet has an invalid checksum. The received checksum, together with the correct checksum, are displayed.

Cause: Most likely, this is a damaged packet. It may be that another node is building an incorrect header.

Action: If the problem persists, examine a line trace to determine where the packet is being damaged.

IP.005

Level: UE–ERROR

Short Syntax: IP.005 pkt trunc *specified_length* pkt ln *true_length source_ip_address –> destination_ip_address*

Long Syntax: IP.005 Packet truncated from *specified_length* to *true_length* bytes from *source_ip_address* for *destination_ip_address*

Description: This message is generated when the packet length specified in the header is greater than the packet buffer length.

Cause: Packet corruption in transit.

Action: If problem persists, check networks and routers.

Cause: Programming error in remote note.

IP.006

Level: CI–ERROR

Short Syntax: IP.006 pkt *source_ip_address –> destination_ip_address* dsc rsn *reason_code*, nt *network ID*

Long Syntax: IP.006 Packet from *source_ip_address* for *destination_ip_address* discarded for reason *reason_code*, network *network ID*

Description: An attempt was made to send the packet on the specified network, but it was not accepted for transmission on that network. The *reason_code* indicates why the packet was not accepted. If the reason was flow–control, an ICMP source quench will be sent to the sender, otherwise an ICMP destination unreachable will be sent.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network_name.*

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

IP.007

Level: P–TRACE

Short Syntax: IP.007 *source_ip_address –> destination_ip_address*

Long Syntax: IP.007 Accepting packet from *source_ip_address* for *destination_ip_address*

Description: This message is generated for each packet which has passed first–level reasonableness checks.

IP.008

Level: U–INFO

Short Syntax: IP.008 no rte *source_ip_address –> destination_ip_address* dsc

Long Syntax: IP.008 No route for packet from *source_ip_address* for *destination_ip_address*; packet discarded

Description: This message is generated when a packet is discarded because there is no route to the destination.

IP.009

Level: CE–ERROR

Short Syntax: IP.009 TTL zero *source_ip_address –> destination_ip_address*

Long Syntax: IP.009 Time–to–live expired on packet from *source_ip_address* for *destination_ip_address*

Description: This message is generated when a packet is discarded because the time–to–live expired.

Cause: The packet has been through more routers than the initial value placed in the time–to–live field of the IP header by the originator. Many older systems use values of 15 or 30, which are not standard–conformant, and are often too small for current *network*s.

Action: Increase initial time–to–live value.

Cause: The packet was in a routing loop, going through a sequence of routers over and over until the time–to–live expired.

Action: Check the routing from the source of the packet to the destination, and see that there are no loops. However, temporary loops are an inevitable result of the timing out of routes in some routing protocols.

IP.010

Level: P–TRACE

Short Syntax: IP.010 dsc pkt *source_ip_address –> destination_ip_address* nt *network ID* no IP

Long Syntax: IP.010 Discarded packet from *source_ip_address* for *destination_ip_address* net *network ID*, no IP forwarder

Description: This message is generated by the fake IP forwarder for each packet which is received.

Cause: Received IP packet, but no IP forwarder.

IP.011

Level: C–INFO

Short Syntax: IP.011 unsup mcst *source_ip_address –> destination_ip_address*

Long Syntax: IP.011 Unsupported multicast from *source_ip_address* for *destination_ip_address*

Description: This message is generated when an unsupported multicast packet is received.

IP.012

Level: UE–ERROR

Short Syntax: IP.012 bd nt cl *source_ip_address –> destination_ip_address*

Long Syntax: IP.012 Bad network class from *source_ip_address* for *destination_ip_address*

Description: This message is generated when a packet is destined for a network which is not class A, B, C or D.

Cause: The indicated source node has sent a packet which the router cannot forward because the network class is unsupported.

IP.013

Level: C–INFO

Short Syntax: IP.013 unsup bcst *source_ip_address –> destination_ip_address*

Long Syntax: IP.013 Unsupported broadcast from *source_ip_address* for *destination_ip_address*

Description: This message is generated when an unsupported broadcast packet is received.

IP.014

Deleted: Message deleted.

IP.015

Level: UE–ERROR

Short Syntax: IP.015 bad subnet *source_ip_address –> destination_ip_address*

Long Syntax: IP.015 Bad subnet in packet from *source_ip_address* for *destination_ip_address*

Description: This message is generated when a packet cannot be routed because of an invalid subnet specification.

IP.016

Level: UI–ERROR

Short Syntax: IP.016 nt *subnet_address* add fail, bd sbnt

Long Syntax: IP.016 Add failed for net *subnet_address*; bad subnet number

Description: This message is generated when a network cannot be added to the routing table because of a bad subnet number.

IP.017

Level: UI–ERROR

Short Syntax: IP.017 nt *network_address* add fail, tbl ovrfl

Long Syntax: IP.017 Add failed for net *network_address*; routing table overflow

Description: This message is generated when a network cannot be added to the routing table because the table is full.

IP.018

Level: UI–ERROR

Short Syntax: IP.018 nt *network_address* add fail, bd nt

Long Syntax: IP.018 Add failed for net *network_address*; bad network number

Description: This message is generated when a network cannot be added to the routing table because of a bad network number.

IP.019

Level: U–INFO

Short Syntax: IP.019 re–add stat rt to **network**

Long Syntax: IP.019 Re–adding static route to net *network*

Description: This message is generated when a static route to a network is brought back into use.

IP.020

Level: UI–ERROR

*Short Syntax***:** IP.020 int for *network* add fail, dup addr

*Long Syntax***:** IP.020 Add of interface for net *network* failed; duplicate address

Description: This message is generated when a network cannot be added to the routing table because the access was denied.

IP.021

Deleted: Message deleted.

IP.022

Level: U–INFO

*Short Syntax***:** IP.022 add nt net_*ip_address* int *int_ip_address* nt *network ID*

*Long Syntax***:** IP.022 Added network net_*ip_address* to interface *int_ip_address* on net *network ID*

Description: This message is generated when a new directly–connected network is added to the routing table.

IP.023

Deleted: Message deleted.

IP.024

Level: CE–ERROR

*Short Syntax***:** IP.024 ign stat rt to *network*, mask *mask*

*Long Syntax***:** IP.024 Ignoring bad static route/filter to *network*, mask *mask*

Description: This message is generated when a bad static route or IP filter is encountered.

IP.025

Level: U–INFO

*Short Syntax***:** IP.025 add nt *network* rt via *network* nt *network ID*

*Long Syntax***:** IP.025 Added network *network* with route via *network* on net *network ID*

Description: This message is generated when a new indirectly–connected network is added to the routing table.

IP.026

Deleted: Message deleted.

IP.027

Deleted: Message deleted.

IP.028

Level: ALWAYS

*Short Syntax***:** IP.028 unnum stat rt on non–SL, *network* thru *network*

*Long Syntax***:** IP.028 Unnumbered static route on non–serial line, *network* thru *network*

Description: A static route has been configured with next hop of 0.0.0.x, yet x is not the interface number of an unnumbered serial line. The static route is ignored.

IP.029

Deleted: Message deleted.

IP.030

Deleted: Message deleted.

IP.031

Level: ALWAYS

*Short Syntax***:** IP.031 Unnum addr rej, nt *network ID*

*Long Syntax***:** IP.031 Unnumbered address rejected, net *network ID*

Description: An attempt has been made to configure an interface as unnumbered, yet either the interface is not a serial line or the interface already has been assigned an IP address. The unnumbered configuration request is ignored.

IP.032

Level: CI–ERROR

*Short Syntax***:** IP.032 fq ovf *source_ip_address –> destination_ip_address* nt *network ID*

*Long Syntax***:** IP.032 Fragment queue overflow from *source_ip_address* for *destination_ip_address* on net *network ID*

Description: This message is generated when an incoming fragment is discarded because the fragment queue overflowed.

IP.033

Level: CE–ERROR

***Short Syntax*:** IP.033 cant frg *source_ip_address –> destination_ip_address* nt *network ID*

***Long Syntax*:** IP.033 Cannot fragment packet from *source_ip_address* for *destination_ip_address* net *network ID*

Description: This message is generated when an outgoing packet needs to be fragmented but has the “don’t fragment” bit set.

IP.034

Level: CE–ERROR

***Short Syntax*:** IP.034 bd frg *source_ip_address –> destination_ip_address* foff *offset*

***Long Syntax*:** IP.034 Bad fragment from *source_ip_address* for *destination_ip_address* with fragment offset *offset*

Description: This message is generated when an outgoing packet has an invalid length of fragment offset.

IP.035

Level: CI–ERROR

***Short Syntax*:** IP.035 cant alloc for frg nt *network ID*

***Long Syntax*:** IP.035 Cannot allocate buffer for fragment for net *network ID*

Description: This message is generated when no buffer is available to fragment a packet.

IP.036

Level: P–TRACE

***Short Syntax*:** IP.036 rcv pkt prt *protocol* frm *source_ip_address*

***Long Syntax*:** IP.036 Received packet for protocol *protocol* from *source_ip_address*

Description: This message is generated for each packet destined for the router.

IP.037

Level: C–TRACE

***Short Syntax*:** IP.037 brd pkt *source_ip_address –> destination_ip_address* prot *protocol* no srvr

***Long Syntax*:** IP.037 Broadcast packet from *source_ip_address*, for *destination_ip_address*, protocol *protocol*; no server

Description: This message is generated when a broadcast packet arrives for an unknown protocol.

IP.038

Level: U–INFO

***Short Syntax*:** IP.038 pkt *source_ip_address –> destination_ip_address* prt *protocol* no srvr

***Long Syntax*:** IP.038 Packet from *source_ip_address*, for *destination_ip_address*, protocol *protocol*; no server

Description: This message is generated when a packet arrives for an unknown protocol. The packet was destined for the router.

IP.039

Level: C–INFO

***Short Syntax*:** IP.039 GGP echo frm *source_ip_address –> destination_ip_address*

***Long Syntax*:** IP.039 GGP echo from *source_ip_address* for *destination_ip_address*

Description: This message is generated for each GGP echo packet.

IP.040

Level: U–INFO

***Short Syntax*:** IP.040 GGP unhnd opc *opcode, source_ip_address –> destination_ip_address*

***Long Syntax*:** IP.040 GGP unhandled opcode *opcode* from *source_ip_address* for *destination_ip_address*

Description: This message is generated when GGP packet arrives with an unhandled opcode.

IP.041

Level: UE–ERROR

***Short Syntax*:** IP.041 GGP bd opc *opcode source_ip_address –> destination_ip_address*

***Long Syntax*:** IP.041 GGP bad opcode *opcode* from *source_ip_address* for *destination_ip_address*

Description: This message is generated when GGP packet arrives with an invalid opcode.

IP.042

Level: CE–ERROR

***Short Syntax*:** IP.042 illgl ARP sbnt req *source_ip_address –> destination_ip_address*

***Long Syntax*:** IP.042 Illegal ARP subnet request in packet from *source_ip_address* for *destination_ip_address*

Description: This message is generated when an ARP subnet request is not honored due to illegal source or destination IP addresses in the ARP packet.

Cause: No route to requested subnet.

Action: Determine why subnet is not reachable.

Cause: Request is for different IP network than source address.

Action: ARP subnet routing is only for subnets of the host's network. Correct routing code in host.

Cause: IP network is not subnetted.

Action: ARP subnet routing is only supported on subnets.

IP.043

Level: P–TRACE

***Short Syntax*:** IP.043 rcvd ARP sbnt rqst *source_ip_address –> destination_ip_address*

***Long Syntax*:** IP.043 Received ARP subnet route request from *source_ip_address* for *destination_ip_address*

Description: This message is generated when an ARP subnet request is received.

IP.044

Level: C–TRACE

***Short Syntax*:** IP.044 ARP sbnt rqst ign *source_ip_address –> destination_ip_address*

***Long Syntax*:** IP.044 ARP subnet request ignored from *source_ip_address* for *destination_ip_address*

Description: This message is generated when an ARP subnet request is not answered because the route to the target subnet is via another router on the same physical network as the originator.

Cause: ARP subnet routing code will only respond when this router is the best route to the target subnet.

Action: The best router should respond to the ARP subnet request.

IP.045

Level: C–INFO

***Short Syntax*:** IP.045 snt ARP rte *source_ip_address –> destination_ip_address*

***Long Syntax*:** IP.045 Sent ARP route from *source_ip_address* for *destination_ip_address*

Description: This message is generated when an ARP subnet request is answered.

IP.046

Level: C–INFO

***Short Syntax*:** IP.046 unkn opt option frm *source_ip_address*

***Long Syntax*:** IP.046 Unknown option option from *source_ip_address*

Description: This message is generated when an unknown option is specified in the IP header of a packet.

IP.047

Level: UE–ERROR

***Short Syntax*:** IP.047 opt *option* bd fmt frm *source_ip_address*

***Long Syntax*:** IP.047 Bad format for option *option* from *source_ip_address*

Description: This message is generated when an option is incorrectly formatted in the IP header.

IP.048

Level: UE–ERROR

***Short Syntax*:** IP.048 strict src rt bd nxt hop *source_ip_address –> destination_ip_address*

***Long Syntax*:** IP.048 Bad next hop in strict source route from *source_ip_address* for *destination_ip_address*

Description: This message is generated when the next hop specified in the strict source route is invalid.

IP.049

Level: UE–ERROR

***Short Syntax*:** IP.049 bd tmstmp fmt *timestamp* frm *source_ip_address*

***Long Syntax*:** IP.049 Bad timestamp format *timestamp* from *source_ip_address*

Description: This message is generated when the format of the timestamp option is invalid.

IP.050

Level: CE–ERROR

Short Syntax: IP.050 tmstmp ovrf, *source_ip_address –> destination_ip_address*

Long Syntax: IP.050 Timestamp list overflow in packet from *source_ip_address* for *destination_ip_address*

Description: This message is generated when the timestamp list is full and the new timestamp cannot be added.

IP.051

Level: UI–ERROR

Short Syntax: IP.051 rs ovfl, port *port_number* frm *source_ip_address*

Long Syntax: IP.051 Too many re–assembly buffers active; port *port_number* from *source_ip_address*

Description: This message is generated when a new packet needs reassembly but the maximum number of re–assembly buffers has already been assigned.

IP.052

Level: UI–ERROR

Short Syntax: IP.052 no stor for rs, port *port_number* frm *source_ip_address*

Long Syntax: IP.052 Insufficient storage for packet re–assembly; port *port_number* from *source_ip_address*

Description: This message is generated when a new packet needs reassembly but there is not enough storage to allocate a reassembly buffer.

IP.053

Level: UE–ERROR

Short Syntax: IP.053 reas pkt too big (*packet_size* byt), port *port_number* frm *source_ip_address*

Long Syntax: IP.053 Re–assembled packet too large (*packet_size* bytes); port *port_number* from *source_ip_address*

Description: This message is generated when a new packet needs reassembly but it is larger than the maximum size re–assembly buffer. The packet_size is how large the packet would be after adding this fragment, which may not be the last.

IP.054

Level: U–INFO

Short Syntax: IP.054 rs TTL exp, port *port_number* frm *source_ip_address*

Long Syntax: IP.054 Re–assembly TTL expired; port *port_number* from *source_ip_address*

Description: This message is generated when a packet being re–assembled has its time–to–live expire.

IP.055

Level: P–TRACE

Short Syntax: IP.055 rs free, port *port_number* frm *source_ip_address*

Long Syntax: IP.055 Re–assembly buffer free; port *port_number* from *source_ip_address*

Description: This message is generated when a re–assembly buffer is deallocated.

IP.056

Level: U–INFO

Short Syntax: IP.056 add dflt nt gw *ip_address* nt *network ID*

Long Syntax: IP.056 Added default gateway *ip_address* net *network ID*

Description: This message is generated when an interface using a default gateway comes up.

IP.057

Level: U–INFO

Short Syntax: IP.057 del dflt nt gw *ip_address*

Long Syntax: IP.057 Deleted default gateway *ip_address*

Description: This message is generated when a default gateway is deleted.

IP.058

Level: U–INFO

Short Syntax: IP.058 del nt *network* rt via *gateway* nt *network ID*

Long Syntax: IP.058 Deleted net *network* route via *gateway* net *network ID*

Description: This message is generated when a network goes down.

IP.059

Level: U–INFO

Short Syntax: IP.059 sbnt *network* dfnd

Long Syntax: IP.059 Subnet *network* defined

Description: This message is generated when a new subnetted network is defined.

IP.060

Level: U–INFO

Short Syntax: IP.060 del sbntd nt *network*

Long Syntax: IP.060 Deleting subnetted network *network*

Description: This message is generated when a subnetted network is deleted. This happens when there are no longer any interfaces to that network.

IP.061

Level: C–TRACE

Short Syntax: IP.061 add lcl pkt to ip op q

Long Syntax: IP.061 Added locally generated packet to IP output queue

Description: This message is generated whenever a locally generated packet is put on the IP output queue.

IP.062

Level: C–TRACE

Short Syntax: IP.062 rcvd ip frg frm *source_ip_address*

Long Syntax: IP.062 Received IP fragment from *source_ip_address*

Description: This message is generated when an IP fragment, requiring reassembly is received.

IP.063

Level: C–TRACE

Short Syntax: IP.063 rasmd pkt frm *source_ip_address*

Long Syntax: IP.063 Successfully re–assembled packet from *source_ip_address*

Description: This message is generated when an IP packet has been successfully re–assembled.

IP.064

Level: C–TRACE

Short Syntax: IP.064 frg pkt *source_ip_address –> destination_ip_address*

Long Syntax: IP.064 Packet from *source_ip_address* for *destination_ip_address* requires fragmentation

Description: This message is generated when an IP packet needs to be fragmented for transmission.

IP.065

Level: C–TRACE

Short Syntax: IP.065 add frg to op frg q *source_ip_address –> destination_ip_address*

Long Syntax: IP.065 Added fragment to output fragment queue from *source_ip_address* for *destination_ip_address*

Description: This message is generated when an IP packet fragment is put on the output fragment queue.

IP.066

Level: P–TRACE

Short Syntax: IP.066 dsc pkt *source_ip_address –> destination_ip_address* nt *network ID* no IP on int

Long Syntax: IP.066 Discarded packet from *source_ip_address* for *destination_ip_address* net *network ID*, no IP on interface

Description: This message is generated by the stub IP forwarder for each packet which is received on an interface for which IP is not enabled.

IP.067

Level: UE–ERROR

Short Syntax: IP.067 RIP disabld on int *interface_ip_address* var len sbnt msks

Long Syntax: IP.067 RIP disabled on *interface_ip_address* variable length subnet masks

Description: The router is configured with variable length subnet masks on the same network, which RIP can't handle. Thus RIP is disabled on the interface.

IP.068

Level: U–INFO

Short Syntax: IP.068 routing cache cleared

Long Syntax: IP.068 routing cache cleared

Description: The IP routing cache has been cleared, probably as the result of a routing table change.

IP.069

Level: U–INFO

Short Syntax: IP.069 routing cache garbage collecting...

Long Syntax: IP.069 routing cache garbage collecting...

Description: The IP routing cache is collecting nonsense data. This takes several passes, and is only done when the cache starts overflowing.

IP.070

Level: U–INFO

Short Syntax: IP.070 cache entry *ip_destination* cleared

Long Syntax: IP.070 routing cache entry for destination *ip_destination* cleared

Description: The IP routing cache entry for the listed destination has been cleared.

IP.071

Level: C–TRACE

Short Syntax: IP.071 ARP sbnt rqst *source_ip_address –> destination_ip_address*, same sbnt, ign

Long Syntax: IP.071 Received ARP subnet route request from *source_ip_address* for *destination_ip_address*, same subnet, ignored

Description: This message is generated when an ARP subnet request is received for a host on the same subnet as it was received. The router ignores this, since that node is on this subnet, and should respond on its own. This message also happens when the router sends an ARP request on a network where the hardware receives its own broadcasts.

IP.072

Level: UE–ERROR

Short Syntax: IP.072 LL broadcast *source_ip_address –> destination_ip_address*, discarded

Long Syntax: IP.072 Received link level broadcast from *source_ip_address* for *destination_ip_address*, discarded

Description: This message is generated when an attempt is made to forward an IP packet that was received as a link level broadcast/multicast. Such packets are not forwarded, and are discarded without even sending back an ICMP message to the source.

IP.073

Level: UI–ERROR

Short Syntax: IP.073 can't copy *source_ip_address –> destination_ip_address*, discarded

Long Syntax: IP.073 Can't copy packet from *source_ip_address* for *destination_ip_address*, discarded

Description: This message is generated when an attempt is made to copy a packet for one of the router's internal applications (e.g., during multicast forwarding), and the router is unable to get a buffer. The requested service then fails.

IP.074

Deleted: Message deleted.

IP.075

Deleted: Message deleted.

IP.076

Deleted: Message deleted.

IP.077

Deleted: Message deleted.

IP.078

Level: C–TRACE

Short Syntax: IP.078 Acc cont miss dropped, *source_ip_address –> destination_ip_address*, prot protocol

Long Syntax: IP.078 Access control miss dropped, packet from *source_ip_address* to *destination_ip_address*, IP protocol number protocol

Description: This message is generated when a IP packet matches none of the access control records. The packet will be dropped.

IP.079

Level: C–TRACE

Short Syntax: IP.079 Acc cont miss dropped, *source_ip_address –> destination_ip_address*, protocol port *source_port –> destination_port*

Long Syntax: IP.079 Access control miss dropped, packet from *source_ip_address* to *destination_ip_address*, protocol port number *source_port* to *destination_port*

Description: This message is generated when a IP packet matches none of the access control records. The packet will be dropped.

IP.080

Level: U–TRACE

Short Syntax: IP.080 new router *router_address*

Long Syntax: IP.080 new router *router_address* has been discovered

Description: A new router has been discovered, either through static configuration, an ICMP redirect, RIP or ICMP router discovery. This message is produced only when running as an IP host (i.e., when IP routing disabled).

IP.081

Level: UE–ERROR

Short Syntax: IP.081 IP ds nt rn on *nettype*/*n_net*

Long Syntax: IP.081 IP protocol does not run over *nettype*/*n_net*

Description: An IP address was configured for a type of network which currently doesn't support IP.

IP.082

Level: UE–ERROR

Short Syntax: IP.082 shrt pkt ln *packet_length*, *source_ip_address –> destination_ip_address*

Long Syntax: IP.082 IP length of *packet_length* in packet from *source_ip_address –> destination_ip_address* is too short

Description: This message is generated when a packet's indicated length is below the minimum possible length. The packet is discarded.

Cause: Most likely, this packet has been incorrectly formatted by the source.

IP.083

Level: C–TRACE

Short Syntax: IP.083 Acc cont # *record_number* dropped, *cache*_status, *source_ip_address –> destination_ip_address*, prot *protcol_number*

Long Syntax: IP.083 Access control number *record_number* dropped, *cache*_status, packet from *source_ip_address* to *destination_ip_address*, IP protocol number *protcol_number*

Description: This message is generated when a IP packet matches one of the exclusive access control entries. The packet will be dropped. The *record_number* is the number of the access control record matched, or zero for no record (end–of–list). The *cache*_status will be ''cache–hit'' or ''cache–miss''.

IP.084

Level: C–TRACE

Short Syntax: IP.084 Acc cont # *record_number* passed, *cache*_status, *source_ip_address* –> *destination_ip_address*, prot *protcol_number*

Long Syntax: IP.084 Access control number *record_number* passed, *cache*_status, packet from *source_ip_address* to *destination_ip_address*, IP protocol number *protcol_number*

Description: This message is generated when a IP packet matches one of the inclusive access control entries. The packet may be forwarded. The *record_number* is the number of the access control record matched. The *cache*_status will be ''cache–hit'' or ''cache–miss''.

IP.085

Level: C–TRACE

Short Syntax: IP.085 Acc cont # *record_number* dropped, *cache*_status, *source_ip_address* –> *destination_ip_address*, protocol port *source_port* –> *destination_port*

Long Syntax: IP.085 Access control number *record_number* dropped, *cache*_status, packet from *source_ip_address* to *destination_ip_address*, protocol port number *source_port* to *destination_port*

Description: This message is generated when a IP packet matches one of the TCP or UDP exclusive access control entries. The packet will be dropped. The *record_number* is the number of the access control record matched, or zero for no record (end–of–list). The *cache*_status will be ''cache–hit'' or ''cache–miss''.

IP.086

Level: C–TRACE

Short Syntax: IP.086 Acc cont # *record_number* passed, *cache*_status, *source_ip_address* –> *destination_ip_address*, protocol port *source_port* –> *destination_port*

Long Syntax: IP.086 Access control number *record_number* passed, *cache*_status, packet from *source_ip_address* to *destination_ip_address*, protocol port number *source_port* to *destination_port*

Description: This message is generated when a IP packet matches one of the TCP or UDP inclusive access control entries. The packet may be forwarded. The *record_number* is the number of the access control record matched. The *cache*_status will be ''cache–hit'' or ''cache–miss''.

IPPN

IP Protocol Network

This chapter describes the IP Protocol Network messages. For information about message content and how to use the message, refer to the Introduction.

IPPN.001

Deleted: Message deleted.

IPPN.002

Level: U–INFO

*Short Syntax***:** IPPN.002 SRT *src_IP–>dst_IP* (UDP *src_port–>dst_port*) ign, no bdg on tunnel

*Long Syntax***:** IPPN.002 SRT packet from *src_IP* to *dst_IP* (from UDP socket *src_port* to *dst_port*) ignored, no bridging on tunnel

Description: A IP packet was received for one of the SRT tunnel's UDP ports, but bridging is not enabled on the tunnel. The packet will be discarded.

Cause: Another bridge is configured to have this bridge as a participant in a SRT bridging tunnel, but this bridge is not so configured.

Action: Makeconfiguration consistent.

Cause: Some other application on the IP network is sending packets to one of the SRT tunnel UDP ports on this router.

Action: Either change application, or ignore this message.

IPPN.003

Level: P–TRACE

Short Syntax: IPPN.003 SRT *src_IP–>dst_IP* (UDP *src_port–>dst_port*) ign, port blocked

Long Syntax: IPPN.003 SRT packet from *src_IP* to *dst_IP* (from UDP socket *src_port* to *dst_port*) ignored, port blocked

Description: A IP packet (which was not a BPDU) was received for one of the SRT tunnel's UDP ports, but that port is in "blocking" or "listening" state. The packet will be discarded.

Cause: Perfectly normal when one port into the tunnel blocks. However, ports to the tunnel will not ordinarily block unless there is an alternative bridging path in parallel with the tunnel.

IPPN.004

Level: P–TRACE

Short Syntax: IPPN.004 Old SRB *src_IP–>dst_IP* (UDP *src_port–>dst_port*), compat mode

Long Syntax: IPPN.004 Old SRB packet from *src_IP* to *dst_IP* (from UDP socket *src_port* to *dst_port*), in compatibility mode

Description: A packet has been received from a node participating in the IP tunnel which is using the SRB tunnel encapsulation used prior to Release 12.0. This packet will be processed normally, but the tunnel will now remain in the mode compatible with the old encapsulation. This means that FCS will never be preserved for 802.5 frames across the tunnel.

Cause: Node running SRB tunnel software from before Release 12.0.

Action: Update all participants in tunnel to Release 12.0, and you will not get this message, and will be able to preserve 802.5 FCS across the tunnel.

Panic "ippnudpregfail"

Short Syntax: IPPN UDP socket registration failure

Description: The IPPN protocol net was unable to register one of the UDP sockets it requires with the UDP protocol.

Cause: Bug in software.

Action: Contact customer service.

IPX

Internet Packet Exchange

This chapter describes Internet Packet Exchange (IPX) protocol messages. This is the network layer for Novell Advanced NetWare. For information about message content and how to use the message, refer to the Introduction.

IPX.001

Level: P–TRACE

Short Syntax: IPX.001 *source_net/source_node –> dest_net/dest_node* ign

Long Syntax: IPX.001 Packet from *source_net/source_node* for *dest_net/dest_node* ignored

Description: An IPX packet arrived on a network and the IPX forwarder is not installed.

IPX.002

Level: UI–ERROR

Short Syntax: IPX.002 q ovf *source_net/source_node –> dest_net/dest_node* nt *network ID*

Long Syntax: IPX.002 Queue overflow, *source_net/source_node –> dest_net/dest_node*, net *network ID*

Description: IPX forwarder input queue has overflowed.

Cause: More packets are being received than the forwarder can forward.

IPX.003

Level: UE–ERROR

Short Syntax: IPX.003 bad hst chksm frm *source_net/source_node*

Long Syntax: IPX.003 Bad host checksum from *source_net/source_node*

Description: This message is generated when a packet arrives for this host with an incorrect checksum.

IPX.004

Level: U–INFO

***Short Syntax*:** IPX.004 err pkt *error_type* frm *source_net/source_node*

***Long Syntax*:** IPX.004 Error packet, errno *error_type*, received from *source_net/source_node*

Description: This message is generated when an error packet is received.

IPX.005

Level: U–TRACE

***Short Syntax*:** IPX.005 no hndlr for skt *destination_socket* typ *packet_type* frm *source_net/source_node*

***Long Syntax*:** IPX.005 No handler for socket *destination_socket* type *packet_type* from *source_net/source_node*

Description: A packet arrived for an unknown or unsupported socket or type. The packet was a broadcast packet.

IPX.006

Level: UE–ERROR

***Short Syntax*:** IPX.006 no hndlr for skt *destination_socket* typ *packet_type* frm *source_net/source_node*

***Long Syntax*:** IPX.006 No handler for socket *destination_socket* type *packet_type* from *source_net/source_node*

Description: A packet arrived for an unknown or unsupported socket or type. The packet was addressed to the router.

IPX.007

Level: UI–ERROR

***Short Syntax*:** IPX.007 no hst addr set for nt *network ID*, not enabled

***Long Syntax*:** IPX.007 no host address set for net *network ID*, not enabled

Description: The forwarder was bringing up IPX on the specified serial line interface, but no host address was set so the interface was not enabled.

Cause: Serial line enabled without setting host address.

Action: Set IPX host address.

IPX.008

Level: UE–ERROR

Short Syntax: IPX.008 SAP bad typ *packet_type* frm *source_net/source_node*

Long Syntax: IPX.008 SAP bad type *packet_type* from *source_net/source_node*

Description: This message is generated when a packet is received with a bad SAP type.

IPX.009

Level: C–TRACE

Short Syntax: IPX.009 SAP gen rply frm *source_net/source_node*

Long Syntax: IPX.009 SAP general reply from *source_net/source_node*

Description: SAP has received a General Reply packet from the specified host. The data in the packet will be used to update the SAP database.

IPX.010

Level: UI–ERROR

Short Syntax: IPX.010 SAP tbl ovrfl, dsc type *service_type* nm [*service_name*]

Long Syntax: IPX.010 SAP table overflow, discarded type *service_type* name [*service_name*]

Description: A new entry cannot be added to the SAP table because it is full. The new entry is discarded.

Cause: SAP table is smaller than number of services on IPX internet.

Action: Increase the size of the SAP table.

IPX.011

Level: UE–ERROR

Short Syntax: IPX.011 SAP srvc typ *service_type* nm [*service_name*] mvd to *new_net*/*new_node*

Long Syntax: IPX.011 SAP service type *service_type* name [*service_name*] moved to *new_net*/*new_node*

Description: A SAP General Reply was received with a different *network*/address pair than is presently in the SAP database.

Cause: Duplicate name assigned for service.

Action: Eliminate duplicated name.

Cause: Service physically moved faster than SAP timeout.

Action: Do not move services so fast.

IPX.012

Level: U–INFO

Short Syntax: IPX.012 SAP del typ *service_type* nm [*service_name*]

Long Syntax: IPX.012 SAP deleted type *service_type* name [*service_name*]

Description: A SAP table entry has been declared dead. It will be advertised as unreachable for another 60 seconds, and then removed from the SAP table.

Cause: No SAP General Reply has been heard containing data on this service type/name pair in 240 seconds.

Action: None, unless service should be up.

IPX.013

Level: UE–ERROR

Short Syntax: IPX.013 SAP bd nearest qry frm *source_net/source_node* ln length

Long Syntax: IPX.013 SAP bad length Nearest Service Query from *source_net/source_node*, len length

Description: A SAP Nearest Service Query was received with an illegal length.

Cause: Programming error in remote node.

IPX.014

Level: C–TRACE

Short Syntax: IPX.014 SAP nearest qry frm *source_net/source_node*

Long Syntax: IPX.014 SAP nearest query from *source_net/source_node*

Description: A SAP Nearest Service Query was received from the specified node. It will be answered as appropriate.

IPX.015

Level: C–TRACE

Short Syntax: IPX.015 SAP gen qry frm *source_net/source_node*

Long Syntax: IPX.015 SAP general query from *source_net/source_node*

Description: A SAP General Service Query was received from the specified node. It will be answered as appropriate.

IPX.016

Level: U–TRACE

Short Syntax: IPX.016 SAP qry sent nt *network ID*

Long Syntax: IPX.016 SAP General Service Query sent, net *network ID*

Description: A SAP General Service Query was sent on the specified network. One is sent on a network when it comes up.

IPX.017

Level: UI–ERROR

Short Syntax: IPX.017 No mem fr SAP bcst nt *network ID*, *count* pkts snt

Long Syntax: IPX.017 No memory for SAP General Service Query or Reply, net *network ID*, *count* packets sent

Description: This message is generated when no buffer is available to send a SAP General Service Query or Reply packet. Since a General Service Reply can require multiple packets, the message notes how many packets were sent on this network before they ran out.

IPX.018

Level: C–TRACE

Short Syntax: IPX.018 SAP gen rply sent nt *network ID*, *count* pkts

Long Syntax: IPX.018 SAP General Service Reply sent, net *network ID*, *count* packets

Description: A SAP General Service Reply has just been sent on the specified network. It took the specified number of packets to send the complete SAP database.

IPX.019

Level: P–TRACE

Short Syntax: IPX.019 NB brd *source_net/source_node –> dest_net/dest_node*, nt *network ID*, *hop_count* hops

Long Syntax: IPX.019 NETBIOS broadcast *source_net/source_node –> dest_net/dest_node*, net *network ID,hop_count* hops

Description: A NETBIOS emulation multi–network broadcast packet has been received for forwarding to other IPX networks. The IPX hop count indicates how many routers it has been through.

IPX.020

Level: U–TRACE

Short Syntax: IPX.020 NB too many hops frm *source_net/source_node* nt *network ID*, ign

Long Syntax: IPX.020 NETBIOS too many hops from *source_net/source_node* net *network ID*, ignored

Description: A NETBIOS emulation broadcast packet has been through more than 8 routers. It will be dropped.

Cause: Normal looping due to multiple paths from source of broadcast packet.

Action: None. This is a normal consequence of the protocol used.

Cause: IPX NETBIOS traffic trying to go across more than 8 hops (networks) between source and destination.

Action: Reconfigure network.

IPX.021

Level: C–TRACE

Short Syntax: IPX.021 NB frm *source_net/source_node* nt *network ID*, already on *connected_network*, ign

Long Syntax: IPX.021 NETBIOS from *source_net/source_node* net *network ID*, already on*connected_network*, ignored

Description: This IPX NETBIOS emulation broadcast packet has already been on one of the directly attached networks. It will not be forwarded, as that would generate a duplicate.

Cause: Normal side–effect of the protocol used.

IPX.022

Level: UI–ERROR

Short Syntax: IPX.022 NB frm *source_net/source_node*, no mem to cpy

Long Syntax: IPX.022 NETBIOS from *source_net/source_node*, no memory to copy

Description: No memory available to make working copy of this NETBIOS emulation packet to send it out multiple interfaces.

IPX.023

Level: CI–ERROR

Short Syntax: IPX.023 NB frm *source_net/source_node*, non–brd nt *network ID* unsupp

Long Syntax: IPX.023 NETBIOS from *source_net/source_node*, non–broadcast net *network ID* unsupported

Description: Attempting to send NETBIOS emulation packet on network that does not support broadcast. The packet will not be sent on that network.

IPX.024

Level: UI–ERROR

Short Syntax: IPX.024 NB frm *source_net/source_node*, unnumbrd nt *network ID* unsupp

Long Syntax: IPX.024 NETBIOS from *source_net/source_node*, un–numbered net *network ID* unsupported

Description: Attempting to send NETBIOS emulation packet on a network with no network number. The packet will not be sent on that network.

Cause: Serial line network operating without a network number.

Action: If you want to run NETBIOS emulation across a serial line network, it must have a network number.

IPX.025

Level: UI–ERROR

Short Syntax: IPX.025 NB frm *source_net/source_node*, no buf to cpy

Long Syntax: IPX.025 NETBIOS from *source_net/source_node*, no buffer to copy

Description: No packet buffer available to copy this NETBIOS emulation broadcast packet into in order to send it on a network.

IPX.026

Level: UI–ERROR

Short Syntax: IPX.026 NB snd dsc, nt *network ID*, rsn *reason_code*

Long Syntax: IPX.026 NETBIOS send discarded, net *network ID*, reason *reason_code*

Description: An outgoing NETBIOS emulation broadcast packet was not successfully transmitted for the reason indicated by the error code.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network ID*.

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

IPX.027

Level: UE–ERROR

Short Syntax: IPX.027 bad RIP typ *RIP_opcode* frm *source_net/source_node*

Long Syntax: IPX.027 Bad RIP type *RIP_opcode* from *source_net/source_node*

Description: RIP packet received which was not a request or response.

Cause: Programming error on remote node.

IPX.028

Level: C–TRACE

Short Syntax: IPX.028 RIP resp frm *source_net/source_node*

Long Syntax: IPX.028 RIP response from *source_net/source_node*

Description: This message is generated when a RIP response packet is received. It will be parsed, and the data incorporated into the routing table.

IPX.029

Level: UE–ERROR

Short Syntax: IPX.029 bad net *network* in RIP frm *source_net/source_node*

Long Syntax: IPX.029 Bad network *network* in RIP from *source_net/source_node*

Description: A RIP response was received with an entry having a network number of 00000000 or FFFFFFFF. That entry will be ignored.

Cause: Programming error on remote node.

IPX.030

Level: UI–ERROR

Short Syntax: IPX.030 net route table ovrfl, dscrd network

Long Syntax: IPX.030 Network routing table overflow, discarding network

Description: This message is generated when a new network cannot be added to the routing table because it is full. The entry is discarded.

Cause: Routing table too small.

Action: Reconfigure IPX protocol to make routing table larger.

IPX.031

Level: C–INFO

Short Syntax: IPX.031 RIP route to *network* now via *router_net*/*router_node*, *hop_count* hops

Long Syntax: IPX.031 RIP route to network *network* now via *router_net*/*router_node*, *hop_count* hops

Description: This message is generated when the route to a network changes. The specified router_net/router_node is now the best route to this network, with the noted number of hops.

Cause: Newly reachable network (if preceded by message IPX.055).

Cause: Change in network topology causes best route to a network to change. This can happen when networks come up, or go down.

Action: Determine what changes in network topology occurred.

IPX.032

Level: U–INFO

Short Syntax: IPX.032 RIP route to *network* aged away

Long Syntax: IPX.032 RIP route to *network* aged away

Description: This message is generated when a network is declared unreachable because no routing updates have been heard for it in 240 seconds. It will be advertised as unreachable for another 60 seconds, and then deleted from the routing table.

Cause: Intervening router that was advertising this network crashed.

IPX.033

Level: C–TRACE

Short Syntax: IPX.033 Rspnd to RIP rqst frm *source_net/source_node*

Long Syntax: IPX.033 Responding to RIP Request from *source_net/source_node*

Description: This message is generated when a RIP Request packet is being parsed for a Reply.

IPX.034

Level: UE–ERROR

Short Syntax: IPX.034 RIP rqst frm *source_net/source_node* shrt, ln *packet_length*

Long Syntax: IPX.034 RIP Request from *source_net/source_node* too short, len *packet_length*

Description: A RIP request packet was received which is too short to contain one RIP entry. It will be discarded.

Cause: Programming error on remote node.

IPX.035

Level: U–TRACE

Short Syntax: IPX.035 RIP qry sent nt *network ID*

Long Syntax: IPX.035 RIP Query sent, net *network ID*

Description: A RIP Query has been sent on the specified interface. A Query is sent on each interface when it comes up.

IPX.036

Level: UI–ERROR

Short Syntax: IPX.036 No mem for RIP pkt nt *network ID, packet_count* pkts snt

Long Syntax: IPX.036 No memory for RIP packet, net *network ID, packet_count* packets sent

Description: This message is generated when no buffer is available to send a RIP Query or Response packet.

IPX.037

Level: C–TRACE

Short Syntax: IPX.037 RIP resp sent nt *network ID, packet_count* pkts

Long Syntax: IPX.037 RIP Response sent net *network ID,packet_count* packets

Description: This message is generated when a RIP Response is sent. The response was sent in the specified number of packets.

IPX.038

Level: U–TRACE

Short Syntax: IPX.038 *source_net/source_node –> dest_net/dest_node* ign

Long Syntax: IPX.038 Packet from *source_net/source_node* for *dest_net/dest_node* ignored

Description: This message is generated when an IPX packet arrives on a network and the IPX forwarder is not active on that network.

IPX.039

Level: C–TRACE

Short Syntax: IPX.039 RIP delta resp sent nt *network ID, packet_count* pkts

Long Syntax: IPX.039 RIP delta Response sent net *network ID, packet_count* packets

Description: This message is generated when a RIP delta Response is sent. This response only includes those networks whose data changed in the last update period. The response was sent in the specified number of packets.

IPX.040

Level: UI–ERROR

Short Syntax: IPX.040 RIP resp snd dsc, nt *network ID*, rsn *reason_code*

Long Syntax: IPX.040 RIP Response send discarded, net *network ID*, reason *reason_code*

Description: An outgoing RIP response packet was not successfully transmitted for the reason indicated by the error code.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network ID*.

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

IPX.041

Level: UI–ERROR

Short Syntax: IPX.041 RIP query snd dsc, nt *network ID*, rsn *reason_code*

Long Syntax: IPX.041 RIP Query send discarded, net *network ID*, reason *reason_code*

Description: An outgoing RIP query packet was not successfully transmitted for the reason indicated by the error code.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network ID*.

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

IPX.042

Level: C–TRACE

Short Syntax: IPX.042 SAP gives near reply typ *service_type* nm [*service_name*] to *source_net/source_node*, nt *network ID*

Long Syntax: IPX.042 SAP giving Nearest Server Reply of type *service_type* name [*service_name*] to *source_net/source_node*, net *network ID*

Description: A SAP Nearest Service Reply will be sent to the specified node. The *service_type* is the type of service, and the *service_name* is the name of the service.

Cause: Node sent Nearest Server Request, and the nearest server of that type is reachable through this router.

IPX.043

Level: C–TRACE

Short Syntax: IPX.043 SAP nearest qry for typ *service_type* frm *source_net/source_node* nt *network ID*

Long Syntax: IPX.043 SAP Nearest Query for service type *service_type* from *source_net/source_node*, net *network ID*

Description: A SAP Nearest Service Query was received from the specified node via the specified interface. If this router is the best route to the closest server of the specified service_type, this router will answer.

Cause: New IPX node booting on an attached LAN, looking for a first file server (*service_type* of 4).

Cause: Node attempting to locate a particular server by service type, such as a communications server or database server.

IPX.044

Level: C–TRACE

Short Syntax: IPX.044 SAP delta gen rply nt *network ID*, *count* pkts

Long Syntax: IPX.044 SAP delta General Service Reply sent, net *network ID*, *count* packets

Description: A SAP delta General Service Reply has just been sent on the specified network. This Reply only includes those services whose data changed in the last update period. It took the specified number of packets to send the changes in SAP database.

IPX.045

Level: U–INFO

Short Syntax: IPX.045 SAP new serv typ *service_type* nm [*service_name*] via *via_net*/*via_node*, *hop_count* hops, nt *network ID*

Long Syntax: IPX.045 SAP new service route to service type *service_type* name [*service_name*] via *via_net*/*via_node*, *hop_count* hops, net *network ID*

Description: This message is generated when a new service is added to the SAP tableThe specified via_net/via_node is the route to this service, with the noted number of hops.

Cause: New service started on IPX internetwork.

Cause: Existing service becomes reachable, due to change in network connectivity.

IPX.046

Level: U–TRACE

Short Syntax: IPX.046 SAP nearest qry frm *source_net/source_node* ignored, nt *network ID*

Long Syntax: IPX.046 SAP Nearest Query from *source_net/source_node* ignored, net *network ID*

Description: A SAP Nearest Service Query was received from the specified node via the specified interface, but processing of these packets has been administratively disabled on this interface. The query will be ignored.

Cause: User has used IPX Config command DISABLE REPLY–TO–GET–NEAREST–SERVER.

Action: If this is the desired action, none. To enable response (the default), use the IPX Config command ENABLE REPLY–TO–GET–NEAREST–SERVER.

IPX.047

Level: UI–ERROR

Short Syntax: IPX.047 SAP query snd dsc, nt *network ID*, rsn *reason_code*

Long Syntax: IPX.047 SAP Query send discarded, net *network ID*, reason *reason_code*

Description: An outgoing SAP query packet was not successfully transmitted for the reason indicated by the error code.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network ID*.

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

IPX.048

Level: UI–ERROR

Short Syntax: IPX.048 SAP resp snd dsc, nt *network ID*, rsn *reason_code*

Long Syntax: IPX.048 SAP Response send discarded, net *network ID*, reason *reason_code*

Description: An outgoing SAP response packet was not successfully transmitted for the reason indicated by the error code.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network ID*.

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

IPX.049

Level: U–TRACE

Short Syntax: IPX.049 SAP no serv typ *service_type* for *source_net/source_node*, nt *network ID*

Long Syntax: IPX.049 SAP no server of type *service_type* for Query from *source_net/source_node*, net *network ID*

Description: A SAP Nearest Service Query was received from *source_net/source_node*, but the SAP database has no service registered of the desired *service_type*. No response will be sent.

Cause: Service of desired *service_type* is down or unreachable.

Action: Find out why serveice is down or unreachable.

Cause: Workstation looking for non–existent *service_type*.

Action: Correct query on workstation.

IPX.050

Level: U–INFO

Short Syntax: IPX.050 SAP dead serv typ *service_type* nm [*service_name*] from *via_net*/*via_node*, nt *network ID*

Long Syntax: IPX.050 SAP dead service route to service type *service_type* name [*service_name*] reported by *via_net*/*via_node* has become unreachable, net *network ID*

Description: This message is generated when a previously reachable service becomes unreachable, and is marked as Dead in the SAP table. The specified *via_net*/*via_node* is the server or router that announced the service as being unreachable.

Cause: Server administratively disabled, as with :DOWN command.

Action: None.

Cause: Server crashed.

Action: Find out why server crashed.

Cause: Network on which service is provided has become unreachable.

Action: Use IPX console DUMP command to see if network is reachable.

IPX.051

Level: U–INFO

Short Syntax: IPX.051 RIP route died to *network* from *router_net*/*router_node*

Long Syntax: IPX.051 RIP route died to network *network* from router *router_net*/*router_node*

Description: This message is generated when a previously reachable network becomes unreachable, and is marked as Dead in the RIP table. The specified *router_net*/*router_node* is the router that announced the network as being unreachable.

Cause: Remote network, or intervening network, went down.

Action: Find out why remote network went down.

Cause: Network is on router that went down.

Action: Find out why remote router went down.

Cause: Network is connected via File Server that was administratively taken down.

IPX.052

Level: UE–ERROR

Short Syntax: IPX.052 RIP resp frm wrong net *source_net/source_node* not *local_net*, nt *network ID*

Long Syntax: IPX.052 RIP response from wrong network *source_net/source_node* not local network *local_net*, nt *network ID*

Description: This message is generated when a RIP response packet is received with a source network number that is not the same as the network number of this interface. The packet will be ignored.

Cause: Misconfiguration of router on this network.

Action: Correct configuration.

IPX.053

Level: UE–ERROR

Short Syntax: IPX.053 SAP resp frm wrong net *source_net/source_node* not *local_net*, nt *network ID*

Long Syntax: IPX.053 SAP response from wrong network *source_net/source_node* not local network *local_net*, nt *network ID*

Description: This message is generated when a SAP response packet is received with a source network number that is not the same as the network number of this interface. The packet will be ignored.

Cause: Misconfiguration of router on this network.

Action: Correct configuration.

IPX.054

Level: C–INFO

Short Syntax: IPX.054 SAP serv typ *service_type* nm [*service_name*] now via *via_net*/*via_node*, *hop_count* hops, nt *network ID*

Long Syntax: IPX.054 SAP service route to service type *service_type* name [*service_name*] is now via *via_net*/*via_node*, *hop_count* hops, net *network ID*

Description: This message is generated when the route to a service in the SAP table changes. The specified *via_net*/*via_node* is the new route to this service, with the noted number of hops.

Cause: Newly reachable service (if proceeded by IPX.045).

Cause: Change in network topology causes best route to a service to change. This can happen when new networks come up, or go down.

Action: Determine what changes in network topology occurred.

IPX.055

Level: U–INFO

Short Syntax: IPX.055 new RIP net *network* via *router_net*/*router_node*, *hop_count* hops

Long Syntax: IPX.055 New RIP network number *network* via *router_net*/*router_node*, *hop_count* hops

Description: This message is generated when a new network is added to the RIP routing table. The new network was advertised by *router_net*/*router_node*, which is now the route to this network, with the noted number of hops.

IPX.056

Level: U–TRACE

Short Syntax: IPX.056 RIP route to *network* garbage coll

Long Syntax: IPX.056 RIP route to *network* garbage collected

Description: This message is generated when a network is removed from the RIP routing table because no routing updates have been heard for it in 300 seconds. This normally happens 60 seconds after an IPX.012 message on the same service.

Cause: Intervening router that was advertising this network went down.

IPX.057

Level: U–INFO

***Short Syntax*:** IPX.057 SAP del typ *service_type* nm [*service_name*], nt *network ID* down

***Long Syntax*:** IPX.057 SAP deleted type *service_type* name [*service_name*], network *network ID* down

Description: The specified network interface has gone down, and this SAP service having a first hop on that network will be placed in the dead state. It will be advertised as unreachable for another 60 seconds, and then removed from the SAP table. However, if there are alternate routes to the same service, they will be learned about within 60 seconds.

Cause: The network via which we reached this service went down.

Action: Bring up network.

IPX.058

Level: U–TRACE

***Short Syntax*:** IPX.058 SAP typ *service_type* nm [*service_name*] garbage coll

***Long Syntax*:** IPX.058 SAP type *service_type* name [*service_name*] garbage collected

Description: This message is generated when a network is removed from the SAP routing table because no SAP responses have been heard for it in 300 seconds.

Cause: Intervening router that was advertising this service went down.

IPX.059

Level: CE–ERROR

***Short Syntax*:** IPX.059 SAP unreach serv typ *service_type* nm [*service_name*] at *service_net*/*service_node* from *via_net*/*via_node*, nt *network ID*

***Long Syntax*:** IPX.059 SAP unreachable service type *service_type* name [*service_name*] at *service_net*/*service_node* from *via_net*/*via_node*, net *network ID*

Description: This message is generated when an advertisement for a service is received, but that service in on an IPX network (service_net) that this router has no route to. This advertisement will be ignored.

Cause: Configuration error on node service_net/service_node.

Action: Correct configuration error.

Cause: Service information for some new service has propagated faster than the associated routing information.

Action: None needed if service_net becomes reachable shortly, and this message does not repeat.

IPX.060

Level: C–TRACE

Short Syntax: IPX.060 SAP periodic GSR starting

Long Syntax: IPX.060 SAP periodic General Service Response starting

Description: The SAP protocol is beginning the transmission of SAP General Service Response packets on all IPX interfaces. This happens approximately every 60 seconds. It will generally be followed by IPX.064 messages (if there are any services), followed by an IPX.061 message.

IPX.061

Level: C–TRACE

Short Syntax: IPX.061 SAP periodic GSR completed, *packet_count* pkts

Long Syntax: IPX.061 SAP periodic General Service Response completed, *packet_count* packet

Description: The SAP protocol has completed the transmission of the periodic SAP General Service Response packets on all IPX interfaces. In doing this, it attempted to send *packet_count* packets in total.

IPX.062

Level: UI–ERROR

Short Syntax: IPX.062 No mem for SAP periodic GSR pkt *packet_number*, delaying, nt *network ID*

Long Syntax: IPX.062 No memory for SAP periodic General Service Response packet *packet_number*, delaying, network *network ID*

Description: There was no packet buffer available to send one packet of the periodic SAP General Service Response. The response will stall for half a second, waiting for a buffer to become available. The packet_number is the count of this packet within the complete response, starting at 0.

Cause: Temporary packet buffer shortage due to traffic peak.

Action: See if message recurs after half a second.

Cause: Parmanent buffer shortage.

Action: Take dump of router and contact Customer Service.

IPX.063

Level: UI–ERROR

Short Syntax: IPX.063 SAP periodic GSR snd dsc, nt *network ID*, rsn *reason_code*

Long Syntax: IPX.063 SAP periodic General Service Response send discarded, net *network ID*, reason*reason_code*

Description: An outgoing SAP periodic General Service Response packet was not successfully transmitted for the reason indicated by the error code.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network ID*.

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

IPX.064

Level: P–TRACE

Short Syntax: IPX.064 SAP periodic GSR sent, *entry_count* entries, nt *network ID*

Long Syntax: IPX.064 SAP periodic General Service Reply sent, *entry_count* entries, net *network ID*

Description: A SAP periodic General Service Reply packet has just been sent on the specified network. There were entry_count services in this packet.

IPX.065

Level: U–INFO

Short Syntax: IPX.065 routing cache cleared

Long Syntax: IPX.065 routing cache cleared

Description: The IPX routing cache has been cleared, probably as the result of a routing table change.

IPX.066

Level: U–INFO

Short Syntax: IPX.066 routing cache garbage collecting...

Long Syntax: IPX.066 routing cache garbage collecting...

Description: The IPX routing cache is collecting nonsense data. This takes several passes, and is only done when the cache starts overflowing.

IPX.067

Level: U–INFO

Short Syntax: IPX.067 cache entry *dest_net/dest_node* cleared

Long Syntax: IPX.067 routing cache entry for destination *dest_net/dest_node* cleared

Description: The IPX routing cache entry for the listed destination has been cleared.

IPX.068

Level: UI–ERROR

Short Syntax: IPX.068 no memory left for IPX local network/node cache entries

Long Syntax: IPX.068 no memory left for IPX local network/node cache entries

Description: The IPX routing local network/node cache needs memory before it can add a new local network and its table into the IPX cache.

IPX.069

Level: UI–ERROR

Short Syntax: IPX.069 *protocol* tbl ovrfl, dst *destination_net*

Long Syntax: IPX.069 *protocol* Table overflow, destination *destination_net*

Description: This message is generated when a new alternate entry cannot be made to routing table because alternate entry space is already full.

Cause: Alternate entry routing table too small.

Action: Increase alternate routing entries for this protocol.

IPX.070

Level: UI–ERROR

Short Syntax: IPX.070 rte ovrfl, dst *destination_net*

Long Syntax: IPX.070 route overflow, destination *destination_net*

Description: This message is generated when a new alternate entry cannot be made to routing table because alternate entry space for a given route is already full.

Cause: Maximum routes per destination network is too small.

Action: Increase maximum routing entries per destination network for this protocol.

IPX.071

Level: U–INFO

Short Syntax: IPX.071 drop pkt w/*encap_seen* using *encap_used* intintnum

Long Syntax: IPX.071 dropped pkt with encaps *encap_seen* using *encap_used* on interface intnum

Description: This message is generated when an IPX packet is received with an encapsulation other than that which has been selected for this interface.

Cause: Normal for networks using multiple encapsulations on a single wire.

Action: None needed.

IPX.072

Level: UI–ERROR

Short Syntax: IPX.072 Error building IPXWAN *iw_pkttype* on net *network ID*

Long Syntax: IPX.072 Error building IPXWAN *iw_pkttype* on network *network ID*

Description: An IPXWAN Response is built from a Request. An attempt has been made to build the response without a request.

IPX.073

Level: UI–ERROR

Short Syntax: IPX.073 Name and Node ID must be config'd before IPXWAN can operate

Long Syntax: IPX.073 Router name and Node ID must be configured before IPXWAN can operate

Description: The IPX configuration parameters Name and Node ID must be configured before IPXWAN can operate on any network.

IPX.074

Level: UI–ERROR

Short Syntax: IPX.074 IPXWAN can't operate on net *network ID* – unsupported type

Long Syntax: IPX.074 IPXWAN can't operate on network *network ID* because it's an unsupported type

Description: IPXWAN has been configured to run on an unsupported interface type.

IPX.075

Level: U–INFO

Short Syntax: IPX.075 IPXWAN is configured but not enabled on net *network ID*

Long Syntax: IPX.075 IPXWAN is configured but not enabled to run on network *network ID*

Description: IPXWAN has been configured to run on the interface, but it has been disabled by the user.

IPX.076

Level: UE–ERROR

Short Syntax: IPX.076 IPXWAN *iw_pkttype* pkt dropped, rcv'd on net *network ID*, unsupported int type

Long Syntax: IPX.076 IPXWAN *iw_pkttype* packet dropeed because it was received on an unsupported interface type, network *network ID*

Description: An IPXWAN packet was dropped because it was received on an unsupported interface type.

IPX.077

Level: UI–ERROR

Short Syntax: IPX.077 IPXWAN *iw_pkttype* pkt dropped, rcv'd on disabled net *network ID*

Long Syntax: IPX.077 IPXWAN *iw_pkttype* packet dropped – it was received on network *network ID* which is disabled for IPXWAN traffic

Description: An IPXWAN packet was dropped because it was received on an interface which is configured to be disabled for IPXWAN traffic.

IPX.078

Level: UE–ERROR

Short Syntax: IPX.078 IPXWAN *iw_pkttype* pkt rejected on net *network ID*, confidence id check failed

Long Syntax: IPX.078 IPXWAN *iw_pkttype* packet received on network *network ID* was rejected due to the confidence ID check failing

Description: An IPXWAN packet was rejected because the confidence ID check failed.

IPX.079

Level: UI–ERROR

Short Syntax: IPX.079 IPXWAN *iw_pkttype* pkt rejected on net *network ID*, non–unique node id

Long Syntax: IPX.079 IPXWAN *iw_pkttype* packet received on network *network ID* was rejected because its node id is identical to the local node id

Description: An IPXWAN packet was rejected because the node id reported in it was identical to the local node id.

Action: Reconfigure the local IPX node id with a unique value.

IPX.080

Level: UI–ERROR

Short Syntax: IPX.080 No memory to build IPXWAN packet

Long Syntax: IPX.080 Not able to get a buffer to build an IPXWAN packet

Description: An attempt to get a buffer to build an IPXWAN packet failed.

IPX.081

Level: UI–ERROR

Short Syntax: IPX.081 Failed to send an IPXWAN *iw_pkttype* pkt on net *network ID*

Long Syntax: IPX.081 An attempt to send an IPXWAN *iw_pkttype* packet on network *network ID* failed

Description: An attempt to send an IPXWAN packet failed.

IPX.082

Level: UI–ERROR

Short Syntax: IPX.082 IPXWAN *iw_pkttype*, pkt rejected on net *network ID*, seq num mismatch

Long Syntax: IPX.082 IPXWAN *iw_pkttype*, packet received on network *network ID* was rejected due to a sequence number mismatch

Description: An IPXWAN packet was dropped due to a sequence number mismatch.

IPX.083

Level: UE–ERROR

Short Syntax: IPX.083 IPXWAN *iw_pkttype* rejected on net *network ID – opt_type* opt not accepted

Long Syntax: IPX.083 IPXWNA *iw_pkttype* rejected on network *network ID–opt_type* option not accepted

Description: An IPXWAN packet was rejected because an option was not accepted by the other side of the link.

IPX.084

Level: U–INFO

Short Syntax: IPX.084 IPXWAN connection to be retried on net *network ID*

Long Syntax: IPX.084 IPXWAN connection to be retried on network *network ID*

Description: A previously timed–out IPXWAN connection is to be retried.

IPX.085

Level: U–INFO

Short Syntax: IPX.085 IPXWAN connection on net *network ID* timed–out

Long Syntax: IPX.085 IPXWAN connection on network *network ID* timed–out

Description: An IPXWAN connection attempt timed–out.

IPX.086

Level: C–INFO

Short Syntax: IPX.086 IPXWAN *iw_pkttype* pkt rcvd on net *network ID*

Long Syntax: IPX.086 IPXWAN *iw_pkttype* packet received on network *network ID*

Description: An IPXWAN packet was successfully received, accepted, and processed.

IPX.087

Level: C–INFO

Short Syntax: IPX.087 IPXWAN *iw_pkttype* pkt sent on net *network ID*

Long Syntax: IPX.087 IPXWAN *iw_pkttype* packet sent on network *network ID*

Description: An IPXWAN packet was successfully sent on the network.

IPX.088

Level: C–INFO

Short Syntax: IPX.088 IPXWAN connection up on net *network ID*

Long Syntax: IPX.088 IPXWAN connection has come up on network *network ID*

Description: An IPXWAN connection is up on the given network.

IPX.089

Level: U–INFO

Short Syntax: IPX.089 IPXWAN connection down on net *network ID*

Long Syntax: IPX.089 IPXWAN connection has gone down on network *network ID*

Description: An IPXWAN connection has gone down the given network. This can happen if the link goes down, if the protocol goes down on the link (IPXCP goes down) or if a Timer Request packet is received.

ISIS

Intermediate System–Intermediate System Protocol

This chapter describes Intermediate System–Intermediate System (ISIS) protocol messages. ISIS is part of the OSI–CLNP forwarder. For information about message content and how to use the message, refer to the Introduction.

ISIS.001

Level: UE–ERROR

Short Syntax: ISIS.001 OSI protocol does not run over *nettype*/*n_net*

Long Syntax: ISIS.001 OSI protocol does not run over *nettype*/*n_net*

Description: OSI was configured to run over a type of network which currently doesn't support OSI.

ISIS.002

Level: UE–ERROR

Short Syntax: ISIS.002 received incomplete isis pdu

Long Syntax: ISIS.002 received incomplete isis packet

Description: A packet fragment recognized as an ISIS packet was received.

ISIS.003

Level: UE–ERROR

Short Syntax: ISIS.003 received isis pdu with a bad version # = *version_number*

Long Syntax: ISIS.003 received packet with a bad version number, vers = *version_number*

Description: An ESIS packet was received but had a bad or unsupported version number.

ISIS.004

Level: UE–ERROR

Short Syntax: ISIS.004 received isis pdu with a bad id length = *id_length*

Long Syntax: ISIS.004 received packet with a bad ID length = *id_length*

Description: An ISIS packet was dropped because it had a bad ID length.

ISIS.005

Level: P_TRACE

Short Syntax: ISIS.005 *pdu_type* rcvd on int interface source id *source_id*

Long Syntax: ISIS.005 *pdu_type* received on interface interface source id *source_id*

Description: An ISIS packet was received.

ISIS.006

Level: UE–ERROR

Short Syntax: ISIS.006 received isis pdu *pdu_type* with bad header length = *hdr_length*

Long Syntax: ISIS.006 received packet, type=*pdu_type*, with a bad header length = *hdr_length* bytes

Description: An ISIS packet with a bad header length has been dropped.

ISIS.007

Level: UE–ERROR

Short Syntax: ISIS.007 received pdu *pdu_type* with out of range area address, length = *add_length*

Long Syntax: ISIS.007 received packet, type=*pdu_type*, with an out of range area address length = *add_length*

Description: An IS–IS packet with an out of range area address has been dropped.

ISIS.008

Level: UE–ERROR

Short Syntax: ISIS.008 isis pdu *pdu_type* received with a bad option *opt_code* length = *opt_length*

Long Syntax: ISIS.008 received packet, type=*pdu_type*, with a bad option, code=*opt_code*, length = *opt_length*

Description: An ISIS packet with an unknown PDU type has been dropped.

ISIS.009

Level: UE–ERROR

Short Syntax: ISIS.009 received isis pdu *pdu_type* with invalid option *opt_code*

Long Syntax: ISIS.009 received packet *pdu_type* with an invalid option = *opt_code*

Description: An ISIS packet with an invalid option has been dropped.

ISIS.010

Level: UE–ERROR

Short Syntax: ISIS.010 received isis pdu *pdu_type* with multiple authentication fields

Long Syntax: ISIS.010 received packet, type=*pdu_type*, with multiple authentication fields

Description: An ISIS packet with multiple authentication fields has been dropped.

ISIS.011

Level: UE–ERROR

Short Syntax: ISIS.011 isis pdu *pdu_type* dropped – unsupported password type = *pwd_type*

Long Syntax: ISIS.011 received packet, type=*pdu_type*, with unsupported password type = *pwd_type*

Description: An ISIS packet with an unsupported password type has been dropped.

ISIS.012

Level: UE–ERROR

Short Syntax: ISIS.012 isis pdu *pdu_type* dropped – authentication failure

Long Syntax: ISIS.012 received packet, type=*pdu_type* – authentication failure

Description: An ISIS packet failed authentication, packet dropped.

ISIS.013

Level: UE–ERROR

Short Syntax: ISIS.013 isis pdu *pdu_type* dropped – bad pdu length = *pdu_length*

Long Syntax: ISIS.013 received packet, type=*pdu_type*, with a bad pdu length = *pdu_length* bytes

Description: An ISIS packet with a bad header length has been dropped.

ISIS.014

Level: UE–ERROR

***Short Syntax*:** ISIS.014 isis pdu *pdu_type* dropped – out of order options

***Long Syntax*:** ISIS.014 received packet, type=*pdu_type*, has out of order options

Description: An ISIS packet with out of order options has been dropped.

ISIS.015

Level: UE–ERROR

***Short Syntax*:** ISIS.015 isis pdu *pdu_type* dropped – out of range prefix address, length = *add_length*

***Long Syntax*:** ISIS.015 received packet, type=*pdu_type*, with an out of range prefix address length = *add_length*

Description: An IS–IS packet with an out of range prefix address has been dropped.

ISIS.016

Level: UE_ERROR

***Short Syntax*:** ISIS.016 mismatch between subnet type and net type on *nettype*/*netnum*

***Long Syntax*:** ISIS.016 mismatch between subnet type and net type on *nettype*/*netnum*

Description: While bringing up a network, an inconsistency between the ISIS subnet type and the network type was discovered.

ISIS.017

Level: UE_ERROR

***Short Syntax*:** ISIS.017 invalid subnet type on *nettype*/*netnum*

***Long Syntax*:** ISIS.017 invalid subnet type on *nettype*/*netnum*

Description: Couldn’t bring up the ISIS subnet due to an invalid subnet type.

ISIS.018

Level: UE_ERROR

***Short Syntax*:** ISIS.018 isis turned off on lan – not started on *nettype*/*netnum*

***Long Syntax*:** ISIS.018 ISIS turned off on lan, ISIS not started on *nettype*/*netnum*

Description: Couldn’t start ISIS on the LAN because ISIS is configured to be off.

ISIS.019

Level: UE_ERROR

Short Syntax: ISIS.019 adjacency not established – no common area

Long Syntax: ISIS.019 Adjacency rejected because it doesn't have a matching area address

Description: The adjacency is rejected because it doesn't have an area address that matches one in the router's set of area addresses.

ISIS.020

Level: UE_ERROR

Short Syntax: ISIS.020 no free IS adjacencies

Long Syntax: ISIS.020 No free IS adjacency structures

Description: Unable to get an IS adjacency structure from the free list.

ISIS.021

Level: UE_ERROR

Short Syntax: ISIS.021 adjacency not established – system type mismatch

Long Syntax: ISIS.021 Adjacency rejected due to a system type mismatch

Description: Adjacency rejected due to a mismatch between the remote system and the router IS type.

ISIS.022

Level: UE_ERROR

Short Syntax: ISIS.022 send of isis pkt failed on *nettype*/*netnum*

Long Syntax: ISIS.022 Send of an ISIS packet on *nettype*/*netnum* failed

Description: An attempt to send an ISIS packet on the specified interface failed.

ISIS.023

Level: P_TRACE

Short Syntax: ISIS.023 Not Used

Long Syntax: ISIS.023 Not Used

Description: Not Used

ISIS.024

Level: P_TRACE

***Short Syntax*:** ISIS.024 iipph pdu sent on *nettype*/*netnum*

***Long Syntax*:** ISIS.024 ISIS point–to–point hello packet sent on *nettype*/*netnum*

Description: An ISIS point–to–point packet was successfully transmitted on the specified interface.

ISIS.025

Level: UE_ERROR

***Short Syntax*:** ISIS.025 no memory for lsu

***Long Syntax*:** ISIS.025 No memory available for the link state update

Description: No memory available for the link state update – entering the wait state.

ISIS.026

Level: UE_ERROR

***Short Syntax*:** ISIS.026 isis pdu not prcssd – sbnt not cnfg'd on *nettype*/*netnum*

***Long Syntax*:** ISIS.026 ISIS pkt not processed – subnet not configured on *nettype*/*netnum*

Description: An ISIS packet was not processed because the subnet was nonexistent or inactive on the interface.

ISIS.027

Level: UE_ERROR

***Short Syntax*:** ISIS.027 isis pdu not processed – pvc not configured

***Long Syntax*:** ISIS.027 ISIS pkt not processed over X25 interface – PVC not configured

Description: ISIS pkt not processed over the specified X25 interface – couldn't find the PVC.

ISIS.028

Level: UE_ERROR

***Short Syntax*:** ISIS.028 isis pdu not processed – isis turned off on *nettype*/*netnum*

***Long Syntax*:** ISIS.028 ISIS packet not processed – ISIS turned off on *nettype*/*netnum*

Description: An ISIS packet was not processed because ISIS was configured to be off on the specified interface.

ISIS.029

Level: UE_ERROR

Short Syntax: ISIS.029 isis pdu not processed – external domain on *nettype*/*netnum*

Long Syntax: ISIS.029 ISIS packet not processed – external domain defined on *nettype*/*netnum*

Description: An ISIS packet was not processed because ISIS was configured to be an external domain.

ISIS.030

Level: UE_ERROR

Short Syntax: ISIS.030 L2 PDU dropped (type = *pdu_type*) – IS type is L1 only

Long Syntax: ISIS.030 Level 2 PDU dropped (type = *pdu_type*), IS type is level 1 only

Description: A level 2 ISIS PDU was dropped because this router is configured with an IS type of level 1 only.

ISIS.031

Level: P_TRACE

Short Syntax: ISIS.031 Not Used

Long Syntax: ISIS.031 Not Used

Description: Not used.

ISIS.032

Level: P_TRACE

Short Syntax: ISIS.032 *pdu_type* sent on int *interface* source id *source_id*

Long Syntax: ISIS.032 *pdu_type* sent on interface *interface* source id *source_id*

Description: An ISIS packet was sent.

ISIS.033

Level: UE–ERROR

Short Syntax: ISIS.033 no iob avail to send ISIS packet

Long Syntax: ISIS.033 no i/o buffer available to send isis packet

Description: An attempt to send an ISIS packet failed because of a lack of system i/o buffers.

ISIS.034

Level: P_TRACE
Short Syntax: ISIS.034 LSU queued oncircuit *circuit* type *type*
Long Syntax: ISIS.034 A link state update was queued on LAN circuit *circuit* type *type*
Description: A link state update was queued on a LAN circuit do to maximum number of transmission constraints.

ISIS.035

Level: UE_ERROR
Short Syntax: ISIS.035 Transmission failed
Long Syntax: ISIS.035 Transmission failed
Description: The handler returned an error on an attemted transmission.

ISIS.036

Level: UE_ERROR
Short Syntax: ISIS.036 Link State database *type* entering wait state
Long Syntax: ISIS.036 Link State database *type* entering wait state
Description: One of the link state databases entered the waiting state.

ISIS.037

Level: P_TRACE
Short Syntax: ISIS.037 Link State database *type* leaving wait state
Long Syntax: ISIS.037 Link State database *type* leaving wait state
Description: One of the link state databases left the waiting state.

ISIS.038

Level: P_TRACE
Short Syntax: ISIS.038 Dijkstra run on level *type*
Long Syntax: ISIS.038 The decision process (Dijkstra) is being run on level *type*
Description: The decision process (Dijkstra) is being run on one of the levels.

ISIS.039

Level: P_TRACE
Short Syntax: ISIS.039 Not used
Long Syntax: ISIS.039 Not used
Description: Not used.

ISIS.040

Level: UE_ERROR

***Short Syntax*:** ISIS.040 Verification of LSP checksum failed, checksum should be *checksum*

***Long Syntax*:** ISIS.040 Verification of LSP checksum failed, checksum should be *checksum*

Description: Verification of a received LSP checksum failed – the user is shown what the checksum should have been.

ISIS.041

Level: U_INFO

***Short Syntax*:** ISIS.041 Not Used

***Long Syntax*:** ISIS.041 Not Used

Description: Not used.

ISIS.042

Level: U_INFO

***Short Syntax*:** ISIS.042 Not Used

***Long Syntax*:** ISIS.042 Not Used

Description: Not used.

ISIS.043

Level: U_INFO

***Short Syntax*:** ISIS.043 Level *level* adj with IS *sysid* is now 2–way

***Long Syntax*:** ISIS.043 Level *level* adj with IS *sysid* is now 2–way.

Description: An IS adj has gone from one–way to two–way and is now in the UP state.

ISIS.044

Level: U_INFO

***Short Syntax*:** ISIS.044 Level *level* adj with IS *sysid* gone from two–way to one–way

***Long Syntax*:** ISIS.044 Level *level* adj with IS *sysid* has gone from being two–way to one–way.

Description: An IS adjacency has gone from being two–way to one–way. The router will rerun the designated router election process and rebuild the pseudonode LSU if it is the designated router.

ISIS.045

Level: U_INFO

Short Syntax: ISIS.045 A new level *level* adj with IS *sysid* has been created

Long Syntax: ISIS.045 A new level *level* adj with IS *sysid* has been created.

Description: A new IS adjacency has been established and placed in the initialization state.

ISIS.046

Level: U_INFO

Short Syntax: ISIS.046 This router has been elected as the level *level* DR on circuit *cct*

Long Syntax: ISIS.046 This router has been elected as the level *level* DR on circuit *cct*

Description: This router has been elected designated router on the specified circuit.

ISIS.047

Level: U_INFO

Short Syntax: ISIS.047 This router has resigned as the level *level* DR on circuit *cct*

Long Syntax: ISIS.047 This router has resigned as the level *level* DR on circuit *cct*

Description: This router has resigned as the designated router on the specified circuit.

ISIS.048

Level: U_INFO

Short Syntax: ISIS.048 System *lanid* has been elected as the level *level* DR on circuit *cct*

Long Syntax: ISIS.048 System *lanid* has been elected as the level *level* DR on circuit *cct*.

Description: The specified system has been elected as the designated router on the specified circuit.

ISIS.049

Level: U_INFO

Short Syntax: ISIS.049 Not Used

Long Syntax: ISIS.049 Not Used

Description: Not Used

ISIS.050

Level: UE_ERROR

***Short Syntax*:** ISIS.050 L1 IS–IS Hello dropped – circuit *cct_id* is L2 only

***Long Syntax*:** ISIS.050 Level 1 IS–IS Hello dropped, circuit *cct_id* is level 2 only

Description: A level 1 ISIS hello packet was dropped because the circuit is configured as level 2 only.

ISIS.051

Level: UE_ERROR

***Short Syntax*:** ISIS.051 LSP dropped – received from non–adjacent system

***Long Syntax*:** ISIS.051 LSP dropped, received from non–adjacent system

Description: A link state packet was dropped because it was received from a system to which no “up” adjacency currently exists or an adjacency exists but is the wrong level.

ISIS.052

Level: UE_ERROR

***Short Syntax*:** ISIS.052 SNP dropped – received from non–adjacent system

***Long Syntax*:** ISIS.052 SNP dropped, received from non–adjacent system

Description: A sequence number packet was dropped because it was received from a system to which no “up” adjacency currently exists or an adjacency exists but is the wrong level.

ISIS.053

Level: UE_ERROR

***Short Syntax*:** ISIS.053 LSP buffer size (*lspbufsz*) > datalink block size (*datalinkblksz*) on int *interface* net *nettype*/netinstance

***Long Syntax*:** ISIS.053 LSP buffer size (*lspbufsz*) is greater than the datalink block size (*datalinkblksz*) on cir *interface* net *nettype*/netinstance

Description: The datalink block size of the circuit is not large enough to accommodate sending ISIS LSPs.

ISIS.054

Level: C_INFO

Short Syntax: ISIS.054 Level *level* PSNP rcvd on ifc *network* dropped – not DR

Long Syntax: ISIS.054 Level *level* Partial Sequence Number PDU received on interface *network* was dropped because this IS is not the designated router.

Description: A partial sequence number PDU was dropped because this intermediate system is not the designated router. Only the designated router processes partial sequence number PDUs.

ISIS.055

Level: UE–ERROR

Short Syntax: ISIS.055 ISIS input que ovflw

Long Syntax: ISIS.055 ISIS input queue overflow

Description: The ISO ISIS input packet queue has overflowed. Packet is dropped.

ISIS.056

Level: UI–ERROR

Short Syntax: ISIS.056 Disabling Integrated ISIS because OSPF is enabled

Long Syntax: ISIS.056 Disabling Integrated ISIS because OSPF is enabled

Description: Integrated ISIS cannot be enabled if OSPF is enabled because these protocols do not currently coordinate access to the IP routing table.

Cause: Both OSPF and Integrated ISIS are enabled in the SRAM configuration.

Action: Disable either OSPF or Integrated ISIS.

ISO

ISO OSI Connectionless Network Layer Protocol

This chapter describes ISO OSI Connectionless Network Layer Protocol (CNLP) messages. For information about message content and how to use the message, refer to the Introduction.

ISO.001

Level: UE–ERROR

Short Syntax: ISO.001 rcvd incmplt pkt

Long Syntax: ISO.001 received incomplete packet

Description: A packet fragment recognized as an ISO CLNP data packet was received.

ISO.002

Level: UE–ERROR

Short Syntax: ISO.002 rcvd pkt bad NSAP len (= *length*)

Long Syntax: ISO.002 received packet with a bad NSAP length (= *length*)

Description: An ISO CLNP data packet was received with an illegal NSAP length.

ISO.003

Level: UE–ERROR

Short Syntax: ISO.003 rcvd pkt bad chksm = *pkt_chksum*

Long Syntax: ISO.003 received packet with a bad checksum = *pkt_chksum*

Description: An ISO CLNP data packet was received but had a bad checksum.

ISO.004

Level: UE–ERROR

Short Syntax: ISO.004 rcvd pkt bad vers # =*version_number*

Long Syntax: ISO.004 received packet with a bad version number (vers = *version_number*)

Description: An ISO CLNP data packet was received but had a bad or unsupported version number.

ISO.005

Level: UE–ERROR

Short Syntax: ISO.005 rcvd pkt bad typ # =*type_field*

Long Syntax: ISO.005 received packet with a bad type field (vers = *type_field*)

Description: An ISO CLNP data packet was received but had a bad or unsupported type field.

ISO.006

Level: UE–ERROR

Short Syntax: ISO.006 rcvd pkt life exp *source_NSAP –> destination_NSAP*

Long Syntax: ISO.006 received packet with an expired liftime *source_NSAP –> destination_NSAP*

Description: An ISO CLNP data packet was received but had a bad checksum.

ISO.007

Level: UE–ERROR

Short Syntax: ISO.007 rcvd pkt bad opt *source_NSAP –> destination_NSAP*

Long Syntax: ISO.007 received packet with a bad optional parameter *source_NSAP –> destination_NSAP*

Description: An ISO CLNP data packet was received with a bad optional parameter.

ISO.008

Level: UE–ERROR

Short Syntax: ISO.008 rcvd pkt dest unkwn *source_NSAP –> destination_NSAP*

Long Syntax: ISO.008 received packet – destination unknown *source_NSAP –> destination_NSAP*

Description: An ISO CLNP data packet is received but can not be routed since there is no routing table entry for destination.

ISO.009

Level: UE–ERROR

Short Syntax: ISO.009 rcvd pkt no seg prmit *source_NSAP –> destination_NSAP*

Long Syntax: ISO.009 received packet–no segmentation permitted *source_NSAP –> destination_NSAP*

Description: An ISO CLNP data packet was received which needed segmentation, but the segmentation permitted flag was not set.

ISO.010

Level: UE–ERROR

***Short Syntax*:** ISO.010 rcvd pkt cnnt fwd *source_NSAP –> destination_NSAP* hndlr err (= *error_code*)

***Long Syntax*:** ISO.010 received packet cannot forward, handler error *source_NSAP –> destination_NSAP* (err = *error_code*)

Description: An ISO CLNP data packet was received and routed but couldn't be forwarded because of a handler error.

ISO.011

Level: UE–ERROR

***Short Syntax*:** ISO.011 CLNP input que ovflw *source_NSAP –> destination_NSAP*

***Long Syntax*:** ISO.011 CLNP input queue overflow *source_NSAP –> destination_NSAP*

Description: The ISO CLNP input packet queue has overflowed. Packet is dropped.

ISO.012

Level: UE–ERROR

***Short Syntax*:** ISO.012 no iob avail to snd err pkt

***Long Syntax*:** ISO.012 no i/o buffer available to send error packet

Description: An attempt to send an ISO CLNP error packet failed because of a lack of system i/o buffers.

ISO.013

Level: UE–ERROR

***Short Syntax*:** ISO.013 no rte to snd err pkt *source_NSAP –> destination_NSAP*

***Long Syntax*:** ISO.013 no route available to send error packet *source_NSAP –> destination_NSAP*

Description: An attempt to send an ISO CLNP error packet failed because it could not be routed.

ISO.014

Level: P–TRACE

***Short Syntax*:** ISO.014 rcvd pkt *source_NSAP –> destination_NSAP*

***Long Syntax*:** ISO.014 received packet *source_NSAP –> destination_NSAP*

Description: An ISO CLNP data packet was received and passed error checking.

ISO.015

Level: UE–ERROR

Short Syntax: ISO.015 cnnt fwd err pkt hndlr err (=*error_code*) *source_NSAP –> destination_NSAP*

Long Syntax: ISO.015 cannot forward an error packet, handler error (err=*error_code*) *source_NSAP –> destination_NSAP*

Description: An ISO CLNP error packet coundn't be forwarded because of a handler error.

ISO.016

Level: UE–ERROR

Short Syntax: ISO.016 ISO ESIS input que ovflw

Long Syntax: ISO.016 ISO ESIS input queue overflow

Description: The ISO ESIS input packet queue has overflowed. Packet is dropped.

ISO.017

Level: UE–ERROR

Short Syntax: ISO.017 OSI unknwn init prot id

Long Syntax: ISO.017 OSI unknown initial protocol identifier

Description: An ISO CLNP packet has been received with an unknown or unsupported initial protocol identifier.

ISO.018

Level: P–TRACE

Short Syntax: ISO.018 rcvd ERR pkt *source_NSAP –> destination_NSAP* cd=*error_code*

Long Syntax: ISO.018 received Error packet *source_NSAP –> destination_NSAP* code = *error_code*

Description: An ISO CLNP Error packet was received for this router.

ISO.019

Level: UE–ERROR

Short Syntax: ISO.019 rcvd DT loc *source_NSAP –> destination_NSAP*

Long Syntax: ISO.019 received Data Packet Local *source_NSAP –> destination_NSAP*

Description: An ISO CLNP Data packet was received with destination NSAP indicating one of the router's NSAP's.

ISO.020

Level: P–TRACE

Short Syntax: ISO.020 sent ERR pkt *destination_NSAP*

Long Syntax: ISO.020 sent Error packet *destination_NSAP*

Description: An ISO CLNP Error packet was sent on receipt of a bad packet.

ISO.021

Level: UE–ERROR

Short Syntax: ISO.021 SRAM err–no NSAP for sbnet

Long Syntax: ISO.021 SRAM error–no NSAP for subnet

Description: A subnet was defined with no NSAP defined for the subnet or domain.

ISO.022

Level: UE–ERROR

Short Syntax: ISO.022 SRAM err – unconcted sbnet

Long Syntax: ISO.022 SRAM error – unconnected subnet

Description: A subnet was defined with no NSAP defined for the subnet or domain.

ISO.023

Level: UE–ERROR

Short Syntax: ISO.023 SRAM err–rte not insrted err=*error_code* Rt *Destination*

Long Syntax: ISO.023 SRAM error – route not intserted error code = *error_code* Route to *Destination*

Description: A statically configured route could not be inserted into routing table.

ISO.024

Level: UE–ERROR

Short Syntax: ISO.024 SRAM err–no adj structs

Long Syntax: ISO.024 SRAM error–no adjacency structures available

Description: Not enough ajacency structures have been configured.

ISO.025

Level: UE–ERROR

Short Syntax: ISO.025 SRAM err–bad ES rte no sub dom = *domain* int = *interface*

Long Syntax: ISO.025 SRAM error–bad static encoded ES route – no subnet domain = *domain* int = *interface*

Description: An encoded end system route was defined for a non–existent subnet.

ISO.026

Deleted: Message deleted at Release 9.0.

ISO.027

Level: UE–ERROR

Short Syntax: ISO.027 SRAM err–bad glbl conf

Long Syntax: ISO.027 SRAM error–bad global configuration

Description: The OSI forwarder has been enabled, but either no domains have been defined, or the number of routes or adjacency is set to 0.

ISO.028

Level: UE–ERROR

Short Syntax: ISO.028 SRAM err–not enough mem

Long Syntax: ISO.028 SRAM error–not enough memory

Description: The OSI forwarder could not get the memory needed to operate.

ISO.029

Level: UE–ERROR

Short Syntax: ISO.029 OSI configured to be disabled

Long Syntax: ISO.029 OSI forwarder is configured to be disabled

Description: The OSI forwarder has not been enabled, either because no global information has been entered or the forwarder has been explicitly disabled.

ISO.030

Level: UE–ERROR

Short Syntax: ISO.030 OSI not starting – check config

Long Syntax: ISO.030 OSI forwarder not starting – check configuration

Description: The OSI forwarder is not starting because of the way it's configured.

ISO.031

Level: UE–ERROR

Short Syntax: ISO.031 rcvd echo dest unkwn *source_NSAP –> destination_NSAP*

Long Syntax: ISO.031 received echo packet – destination unknown *source_NSAP –> destination_NSAP*

Description: An ISO CLNP echo packet is received but can not be routed since there is no routing table entry for destination.

ISO.032

Level: UE–ERROR

***Short Syntax*:** ISO.032 no iob avail to snd echo pkt

***Long Syntax*:** ISO.032 no i/o buffer available to send echo packet

Description: An attempt to send an ISO CLNP echo packet failed because of a lack of system i/o buffers.

ISO.033

Level: UE–ERROR

***Short Syntax*:** ISO.033 cnnt fwd echo pkt hndlr err (=*error_code*) *source_NSAP –> destination_NSAP*

***Long Syntax*:** ISO.033 cannot send an echo packet, handler error (err=*error_code*) *source_NSAP –> destination_NSAP*

Description: An ISO CLNP echo packet couldn't be sent because of a handler error.

ISO.034

Level: P–TRACE

***Short Syntax*:** ISO.034 sent ECHO rply pkt *destination_NSAP*

***Long Syntax*:** ISO.034 sent ECHO reply packet *destination_NSAP*

Description: An ISO CLNP ECHO reply packet was sent on receipt of a bad packet.

ISO.035

Level: P–TRACE

***Short Syntax*:** ISO.035 sent ECHO pkt rqst *destination_NSAP*

***Long Syntax*:** ISO.035 sent ECHO request packet *destination_NSAP*

Description: An ISO CLNP ECHO request packet was sent on receipt of a bad packet.

ISO.036

Level: P–TRACE

***Short Syntax*:** ISO.036 rcvd ECHO rqst *source_NSAP –> destination_NSAP*

***Long Syntax*:** ISO.036 received Echo Request *source_NSAP –> destination_NSAP*

Description: An ISO CLNP Echo packet was received.

ISO.037

Level: P–TRACE
Short Syntax: ISO.037 rcvd ECHO rply *source_NSAP –> destination_NSAP*
Long Syntax: ISO.037 received ECHO reply *source_NSAP –> destination_NSAP*
Description: An ISO CLNP ECHO reply was received.

ISO.038

Level: P–TRACE
Short Syntax: ISO.038 DNA pkt forwarded via OSI at level *rtg_lvl*
Long Syntax: ISO.038 DNA packet forwarded via OSI at level *rtg_lvl*
Description: A DNA packet was received and then passed to OSI for forwarding.

ISO.039

Level: P–TRACE
Short Syntax: ISO.039 DNA pkt translated to OSI pkt *source_NSAP –> destination_NSAP*
Long Syntax: ISO.039 DNA pkt translated to OSI pkt: *source_NSAP –> destination_NSAP*
Description: A DNA data packet was successfully translated to an OSI data packet.

ISO.040

Level: P–TRACE
Short Syntax: ISO.040 Translation of DNA pkt to OSI pkt failed
Long Syntax: ISO.040 Translation of DNA pkt to OSI pkt failed
Description: An attempt to translate a DNA data packet to an OSI data packet failed.

ISO.041

Level: P–TRACE
Short Syntax: ISO.041 OSI pkt translated to DNA pkt *src –> dst*
Long Syntax: ISO.041 OSI pkt translated to DNA pkt: *src –> dst*
Description: An OSI data packet was successfully translated to a DNA data packet.

ISO.042

Level: P–TRACE
Short Syntax: ISO.042 Translation of OSI pkt to DNA pkt failed
Long Syntax: ISO.042 Translation of OSI pkt to DNA pkt failed
Description: An attempt to translate an OSI data packet to a DNA data packet failed.

ISO.043

Level: P–TRACE

Short Syntax: ISO.043 OSI pkt forwarded via DNA at level *rtg_lvl*

Long Syntax: ISO.043 OSI packet forwarded via DNA at level *rtg_lvl*

Description: An OSI packet was received and then passed to DNA for forwarding.

ISO.044

Level: UE–ERROR

Short Syntax: ISO.044 Can't send echo message to local router

Long Syntax: ISO.044 Can't send an echo message to the local router.

Description: An attempt was made to send an echo message to the local router. This could occur if a user enters the send command from the console with the local router's NSAP as the destination address.

ISO.045

Level: UE–ERROR

Short Syntax: ISO.045 Error PDU rcvd from *src_nsap* on nt *network ID* dropped – SP, MS or E/R flag set

Long Syntax: ISO.045 Error PDU received from *src_nsap* on network *network ID* dropped because either the segmentation permitted, more segments, or error report flag was set

Description: An error report PDU was received with either the segmentation permitted, more segments, or error report flag set. These flags are always supposed to be zero for an error PDU. The error PDU is dropped.

LLC

Logical Link Control

This chapter describes IEEE 802.2 Logical Link Control messages. For information about message content and how to use the message, refer to the Introduction.

LLC.001

Level: C–TRACE

Short Syntax: LLC.001 Sent *frame_type*, *src_mac–>dst_mac*, rif saps *src_sap–>dst_sap*, nt *network*

Long Syntax: LLC.001 Sent *frame_type*, *src_mac–>dst_mac*, rif saps *src_sap–>dst_sap*, network *network*

Description: LLC is sending a frame. Possible frame types are: SABME_C0 or SABME_C1 (Set Asynchronous Balanced Mode Extended), DM_R0 or DM_R1 (Disconnected Mode), DISC_C0 or DISC_C1 (Disconnect), RR_C0 or RR_C1 or RR_R0 or RR_R1 (Receiver Ready), RNR_C0 or RNR_C1 or RNR_R0 or RNR_R1 (Receiver Not Ready), REJ_C0 or REJ_C1 or REJ_R0 or REJ_R1 (Reject), UA_R0 or UA_R1 (Unnumbered Acknowledgement), FRMR_R0 or FRMR_R1 (Frame Reject), and I_C0 or I_C1 or I_R0 or I_R1 (Information Frame). The abbrevation suffixes are C0 (command, poll bit off), C1 (command, poll bit on), R0 (response, final bit off), and R1 (response, final bit on).

LLC.002

Level: C–TRACE

Short Syntax: LLC.002 ev=*llc_event* in st=*llc_state*, *llc2_connection*, nt *network*

Long Syntax: LLC.002 event=*llc_event* in state=*llc_state*, *llc2_connection*, network *network*

Description: An event occurred on an llc2 connection. The LLC2 FSM (Finite State Machine) has been called to process the event. The LLC2 connection is uniquely identified by the combination destination MAC address, source MAC address, destination sap, and source sap on a particular network. The possible events are: SET_ABME (user request to connect to remote), SET_ADM (user request to disconnect from remote), SEND_BTU (user request to send data), FLOW_REQ_ON (user request to turn off local busy condition), FLOW_REQ_OFF (user request to turn on local busy condition), T1_EXP (T1 timer expiration), T2_EXP (T2 timer expiration), Ti_EXP (Ti timer expiration), OS_I_C0 or OS_I_C1 or OS_I_R0 or OS_I_R1 (Ns on I–frame is out of sequence), I_C0 or I_C1 or I_R0 or I_R1 (valid I–frame received), RR_C0 or RR_C1 or RR_R0 or RR_R1 (RR frame received), RNR_C0 or RNR_C1 or RNR_R0 or RNR_R1 (RNR frame received), REJ_C0 or REJ_C1 or REJ_R0 or REJ_R1 (REJ frame received), UA_R0 or UA_R1 (UA frame received), DISC_C0 or DISC_C1 (DISC frame received), DM_R0 or DM_R1 (DM frame received), FRMR_R0 or FRMR_R1 (FRMRframe received), BAD_FRAME_0 or BAD_FRAME_1 (received frame will generate FRMR), SABME_C0 or SABME_C1 (SABME frame received), and SEND_I_POLL (Sending I frame with Poll bit on). The abbreviation suffixes are C0 (command, poll bit off), C1 (command, poll bit on), R0 (response, final bit off), and R1 (response, final bit on).

LLC.003

Level: C–INFO

Short Syntax: LLC.003 *llc_state–>llc_state, llc2_connection,* nt *network*

Long Syntax: LLC.003 *llc_state* to *llc_state, llc2_connection,* network *network*

Description: There is LLC2 state change. The possible states are: DISCONNECTED (initial state), LINK_OPENING (link establishment in progress), DISCONNECTING (DISC sent, awaiting DM), FRMR_SENT (frmr sent), LINK_OPENED (normal state), LOCAL_BUSY (local is busy), REJECTION (remote sent an out of sequence frame), CHECKPOINTING (poll sent, awaiting response sending of data suspended), CKPT_LB (combination state), CKPT_REJ (combination state), RESETTING (awaiting user response to reset), REMOTE_BUSY (remote is busy), LB_RB (combination state), REJ_LB (combination state), REJ_RB (combination state), CKPT_REJ_LB (combination state), CKPT_CLR (clearing from CKPT_LB state), CKPT_REJ_CLR (clearing from CKPT_REJ_LB state), REJ_LB_RB (combination state), FRMR_RECEIVED (received frmr). The abbreviations above are CKPT=CHECKPOINTING, CLR=CLEARING, LB=LOCAL BUSY, RB=REMOTE BUSY, and REJ=REJECTION.

LLC.004

Level: C–INFO

Short Syntax: LLC.004 Up evt *user_event* args *user_value*/*event_reason* on *llc2–conn*

Long Syntax: LLC.004 Upcall user event *user_event user_value event_reason* on *llc2–conn*

Description: LLC2 event upcall is occurring. Some of the arguments on the upcall are shown. User Cookie is meaningful to the router software running over the LLC subsystem. Event reason sometimes further specifies the event. The possible upcall events are: CONN_IND (cookie=session, reason=none), CONN_IND_PASS (cookie=sap, reason=none), CONN_CONFIRM (cookie=session, reason=none), DISC_IND (cookie=session, possible reasons: local term (disconnecting), remote term, conn refused, local term (disconnected)), RESET_IND (cookie=session, possible reasons: local reset, remote reset, frmr rcvd, frmr sent), RESET_CONF (cookie=session, reason=none), FLOW_IND (cookie=session, possible reasons: flow off, flow on), and DISC_CONFIRM (cookie=session, reason=none).

LLC.005

Level: C–INFO

Short Syntax: LLC.005 prim *user_primitive* sap *SAP_value* on nt *network*

Long Syntax: LLC.005 user primitive *user_primitive* sap *SAP_value* on network *network*

Description: A sap–related LLC user–primitive was called. The possible SAP primitives are: OPEN_SAP, CLOSE_SAP, CLOSE_SAP_FORCED, MODIFY_SAP, OPEN_STATION, and UNITDATA.

LLC.006

Level: C–INFO

Short Syntax: LLC.006 prim *user_primitive*, *llc2_connection*, nt *network*

Long Syntax: LLC.006 primitive *user_primitive*, *llc2_connection*, network *network*

Description: A llc2 connection non–data user–primitive was called. The possible primitives are: CLOSE_STATION, CLOSE_STATION_FORCED, CONNECT_REQUEST, CONNECT_RESPONSE, DISCONNECT_REQUEST, RESET_REQUEST, RESET_RESPONSE, FLOW_REQ.

LLC.007

Level: C–TRACE

Short Syntax: LLC.007 data prim, *llc2_connection*, nt *network*

Long Syntax: LLC.007 data primitive, *llc2_connection*, network *network*

Description: A DATA_REQUEST data primitive was called. DATA_REQUEST passes the data in buffer memory.

LLC.008

Level: C–TRACE

Short Syntax: LLC.008 data prim, *llc2_connection*, nt *network*

Long Syntax: LLC.008 data primitive,*llc2_connection*, network *network*

Description: A DATA_LOCAL data primitive was called. DATA_LOCAL passes the data in data memory.

LLC.009

Level: C–TRACE

Short Syntax: LLC.009 unitdata prim, sap *SAP_value*, nt *network*

Long Syntax: LLC.009 unitdata primitive, sap *SAP_value* network *network*

Description: A UNITDATA llc1 data primitive was called.

LLC.010

Level: UI–ERROR

Short Syntax: LLC.010 out q too big, *llc2_connection*, nt *network*

Long Syntax: LLC.010 outbound queue too big, *llc2_connection*, network *network*

Description: The outbound queue has grown grossly large. The llc2 connection is being automatically terminated.

Cause: LLC application is not responding to flow control.

Action: Contact customer service.

LLC.011

Level: UI–ERROR

Short Syntax: LLC.011 No buf to dup I–frame, *llc2_connection*, nt *network*

Long Syntax: LLC.011 No buffer available to duplicate I–frame, *llc2_connection*, network *network*

Description: No buffer available to duplicate I–frame.

Cause: Severe packet buffer shortage.

Action: Check memory statistics in GWCON to verify packet buffer level. Reduce buffer usage of other router software. Reduce buffer usage by reducing LLC connections, by changing LLC configuration, especially making sure that LLC Transmit and Receive windows are normal sizes.

LLC.012

Level: UI–ERROR

Short Syntax: LLC.012 No mem to dup I–frame, *llc2_connection*, nt *network*

Long Syntax: LLC.012 No memory available to duplicate I–frame, *llc2_connection*, network *network*

Description: No memory to duplicate I–frame.

Cause: Memory shortage.

Action: Reduce memory usage by reducing tables in other software. Reduce memory by reducing LLC connections, by changing LLC configuration, especially making sure that LLC Transmit and Receive windows are normal sizes.

LLC.013

Level: UI–ERROR

Short Syntax: LLC.013 No buf for LLC frame, *llc2_connection*, nt *network*

Long Syntax: LLC.013 No buffer for LLC frame, *llc2_connection*, network *network*

Description: A buffer could not be obtained to to build an LLC Supervisory or Unnumbered frame. No loss of data integrity has occurred yet, but unless buffers for this purpose become available within a few seconds, the other end of the LLC2 connection will most likely terminate this LLC connection as part of the normal LLC2 protocol.

Cause: Severe packet buffer shortage.

Action: Check memory statistics in GWCON to verify packet buffer level.

LLC.014

Level: UI–ERROR

Short Syntax: LLC.014 fr type inv, *llc2_connection*, nt *network*

Long Syntax: LLC.014 frame type invalid, *llc2_connection*, network *network*

Description: The frame type the LLC is trying to build is invalid.

Cause: Hardware failure or software bug.

Action: Contact customer service.

LLC.015

Level: UI–ERROR

Short Syntax: LLC.015 Inv LLC2 ev *event_code_number*

Long Syntax: LLC.015 Invalid LLC2 FSM event *event_code_number*

Description: The LLC2 Finite State Machine (FSM) was called with an event that was out of range.

Cause: Hardware failure or software bug.

Action: Contact customer service.

LLC.016

Level: UI–ERROR

Short Syntax: LLC.016 inv nt typ *network_type* on nt *network*

Long Syntax: LLC.016 invalid network type *network_type* on network *network*

Description: An OPEN SAP operation was tried on a network type that LLC does not support. Network types Token–Ring, Ethernet, and FDDI are supported.

Cause: Software bug.

Action: Contact customer service.

LLC.017

Level: UI–ERROR

***Short Syntax*:** LLC.017 dup sap *SAP_value* on nt *network*

***Long Syntax*:** LLC.017 duplicate sap *SAP_value* on network *network*

Description: A OPEN SAP operation was tried on a sap that has already been opened.

Cause: Software bug.

Action: Contact customer service.

LLC.018

Level: UI–ERROR

***Short Syntax*:** LLC.018 No mem for sap blk on nt *network*

***Long Syntax*:** LLC.018 No memory for SAP control block on network *network*

Description: Unable to allocate memory for SAP control block.

Cause: Severe shortage of memory.

Action: Reduce table sizes in other protocols, use system with less protocols, expand memory in router.

LLC.019

Level: UI–ERROR

***Short Syntax*:** LLC.019 No mem for stn blk on nt *network*

***Long Syntax*:** LLC.019 No memory for station control block on network *network*

Description: Unable to allocate memory for station control block.

Cause: Severe shortage of memory.

Action: Reduce table sizes in other protocols, use system with less protocols, expand memory in router. Reduce number of LLC2 connections.

LLC.020

Level: U–INFO

***Short Syntax*:** LLC.020 UI frm drp *llc2_connection*, nt *network*

***Long Syntax*:** LLC.020 UI frame dropped, *llc2_connection*, network *network*

Description: UI frame refused by the local application within the router.

Cause: The frame was not the type the local application wanted to handle.

Action: None.

LLC.021

Level: U–INFO

***Short Syntax*:** LLC.021 TST frm refused *llc2_connection*, nt *network*

***Long Syntax*:** LLC.021 TEST frame refused, *llc2_connection*, network *network*

Description: TEST frame refused by the local application within the router. The frame is passed on to the bridge code, etc.

Cause: The frame was not the type the local application wanted to handle.

Action: None.

LLC.022

Level: U–INFO

***Short Syntax*:** LLC.022 XID frm refused *llc2_connection*, nt *network*

***Long Syntax*:** LLC.022 XID frame refused, *llc2_connection*, network *network*

Description: XID frame refused by the local application within the router. The frame is passed on to the bridge code, etc.

Cause: The frame was not the type the local application wanted to handle.

Action: None.

LLC.023

Level: C–INFO

***Short Syntax*:** LLC.023 Upcall frm *frame_type*, *src_mac–>dst_mac*, rif saps *src_sap–>dst_sap*, nt *network*

***Long Syntax*:** LLC.023 Upcall frame *frame_type*, *src_mac–>dst_mac*, rif saps *src_sap–>dst_sap*, network *network*

Description: LLC makes an upcall providing the LLC with a unidata frame. The possible unidata frames are: TEST_C0 or TEST_C1 or TEST_R0 or TEST_R1 (TEST frame), XID_C0 or XID_C1 or XID_R0 or XID_R1 (Exchange Identification frame), UI_C0 or UI_R0 (Unnumbered Information). The abbreviation suffixes are C0 (command, poll bit off), C1 (command, poll bit on), R0 (response, final bit off), and R1 (response, final bit on).

LLC.024

Level: UI–ERROR

Short Syntax: LLC.024 llc2 out drp, rsn *reason_code*, *llc2_connection*, nt *network*

Long Syntax: LLC.024 llc2 outbound frame dropped, reason *reason_code*, *llc2_connection*, network *network*

Description: The sending of an LLC2 related outbound frame failed. The reason_code is the internal error code for the failure.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for network_name.

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

LLC.025

Level: UI–ERROR

Short Syntax: LLC.025 *frame_type* out frm drp, rsn *reason_code*, *llc2_connection*, nt *network*

Long Syntax: LLC.025 *frame_type* outbound frame dropped, reason *reason_code*, *llc2_connection*, network *network*

Description: The sending of the user's UNITDATA or an LLC–generated XID or TEST response outbound frame failed. The possible frame test are: TEST_C0 or TEST_C1 or TEST_R0 or TEST_R1 (TEST frame), XID_C0 or XID_C1 or XID_R0 or XID_R1 (Exchange Identification frame), UI_C0 or UI_R0 (Unnumbered Information frame), and unexpected (not one of the above types). The abbreviation suffixes are C0 (command, poll bit off), C1 (command, poll bit on), R0 (response, final bit off), and R1 (response, final bit on).

Cause: See LLC.024.

Action: See LLC.024

LLC.026

Level: UI–ERROR

***Short Syntax*:** LLC.026 No mem for cfg blk on nt *network*

***Long Syntax*:** LLC.026 No memory for LLC CONF BLOCK on network *network*

Description: Unable to allocate memory for an LLC_CONF_BLOCK at initialization time. LLC configuration defaults are used.

Cause: Severe shortage of memory.

Action: Reduce table sizes in other protocols, use system with less protocols, expand memory in router.

LLC.027

Level: U–INFO

***Short Syntax*:** LLC.027 Read LLC Cfg for nt *network*

***Long Syntax*:** LLC.027 Read LLC Configuration record for network *network*

Description: LLC Configuration record read for this network. This only occurs at initialization time. The values in the LLC configuration record are used as default value on the network.

LLC.028

Level: U–INFO

***Short Syntax*:** LLC.028 Inv acc *access_priority* for nt *network*

***Long Syntax*:** LLC.028 Inv access priority *access_priority* for network *network*

Description: access_priority, on a network that that is not a token ring must be zero because it is not used.

Cause: As devices are deleted and added, it is possible for one of the LLC config records to contain a non–zero access priority on a non–Token–Ring LAN interface.

Action: None. You may reconfigure the LLC config on this network to avoid getting this message.

LLC.029

Level: UI–ERROR

***Short Syntax*:** LLC.029 Inv acc *access_priority* for nt *network*

***Long Syntax*:** LLC.029 Inv acc *access_priority* for network *network*

Description: The access priority is greater than 7. A default of 0 is used.

Cause: Configuration memory corruption.

Action: Reconfigure the LLC on this network to avoid getting this message.

LLC.030

Level: C–TRACE

Short Syntax: LLC.030 Inv hw type *hardware_type* in cfg for nt *network*

Long Syntax: LLC.030 Invalid hardware type *hardware_type* for network *network*

Description: An LLC config record exists for an interface that does not have a LAN hardware type.

Cause: As devices are deleted and added, it is possible for one of the LLC config records to contain an interface that is no longer a LAN interface.

Action: None. Situation is not harmful.

LLC.031

Level: C–TRACE

Short Syntax: LLC.031 Inv int *interface_number* in cfg

Long Syntax: LLC.031 Invalid interface *interface_number* in config

Description: An LLC config record exists for an interface that does not exist.

Cause: As devices are deleted and added, it is possible for one of the LLC config records to contain an invalid interface number.

Action: None. Situation is not harmful.

LLC.032

Level: C–INFO

Short Syntax: LLC.032 Sent *frame_type*, *src_mac–>dst_mac*, rif saps *src_sap–>dst_sap*, nt *network*

Long Syntax: LLC.032 Sent *frame_type*, *src_mac–>dst_mac*, rif saps *src_sap–>dst_sap*, network *network*

Description: LLC user is sending a frame, or LLC itself is sending a TEST or XID response frame. The possible frame types are: TEST_C0 or TEST_C1 or TEST_R0 or TEST_R1 (TEST frame), XID_C0 or XID_C1 or XID_R0 or XID_R1 (Exchange Identification frame), UI_C0 or UI_R0 (Unnumbered Information frame). The abbreviation suffixes are C0 (command, poll bit off), C1 (command, poll bit on), R0 (response, final bit off), and R1 (response, final bit on).

LLC.033

Level: C–INFO

Short Syntax: LLC.033 frm to LLC, frm *frame_type*, *src_mac–>dst_mac*, rif saps *src_sap–>dst_sap*, nt *network*

Long Syntax: LLC.033 frm to LLC, frm *frame_type*, *src_mac–>dst_mac*, rif saps *src_sap–>dst_sap*, network *network*

Description: LLC subsystem itself is responding to a TEST or XID frame. The possible frame types are: TEST_C0 or TEST_C1 (TEST frame), and XID_C0 or XID_C1 (Exchange Identification frame). The abbreviation suffixes are: C0=(command, pollbit off), and C1=(command, pollbit on),

LNM

LAN Network Manager

This chapter describes LAN Network Manager (LNM) agent messages. LNM provides IBM–compatible network management of the ASRT bridge. For information about message content and how to use the message, refer to the Introduction.

LNM.001

Level: C–INFO

Short Syntax: LNM.001 Configuring port *port_number*

Long Syntax: LNM.001 Configuring port *port_number*

Description: LNM is beginning Configuration of the specified port.

LNM.002

Level: C–INFO

Short Syntax: LNM.002 Configuration complete port *port_number* nt *network*

Long Syntax: LNM.002 Configuration complete port *port_number* network *network*

Description: LNM has completed the Configuration of the specified port.

LNM.003

Level: U–INFO

Short Syntax: LNM.003 LNM configured for port *port_number*, port does not exist in Bridge Configuration

Long Syntax: LNM.003 LNM configured for port *port_number*, but the port is not configured in the Bridge Configuration

Description: The port is configured in the LNM configuration, but not in the SRT configuration.

Cause: User configuration error.

Action: Reconfigure LNM or SRT. Ensure Bridge is enabled.

LNM.004

Level: U–INFO

***Short Syntax*:** LNM.004 LNM configured for port *port_number*, is not SRB port

***Long Syntax*:** LNM.004 LNM configured for port *port_number*, is not configured for SRB

Description: The port is configured in the LNM configuration, but is not configured as an SRB port in the SRT configuration.

Cause: User configuration error.

Action: Reconfigure LNM or SRT.

LNM.005

Level: U–INFO

***Short Syntax*:** LNM.005 LNM configured for port *port_number*, is not token ring

***Long Syntax*:** LNM.005 LNM configured for port *port_number*, is not a token ring interface

Description: The port is configured in the LNM configuration, but the interface is not a Token–Ring interface.

Cause: User configuration error.

Action: Reconfigure LNM or the interface.

LNM.006

Level: UI–ERROR

***Short Syntax*:** LNM.006 No iorb to transmit packet

***Long Syntax*:** LNM.006 No buffer available to copy one or more packets

Description: No buffer available to copy one or more packets in order to send through LLC.

Cause: Severe packet buffer shortage.

Action: Check memory statistics in GWCON to verify packet buffer level.

Cause: Traffic peak using all available buffers.

Action: This is the problem if this message occurs very infrequently.

LNM.007

Level: C–INFO

***Short Syntax*:** LNM.007 Initializing port *port_number* nt *network*

***Long Syntax*:** LNM.007 Initializing port *port_number* network *network*

Description: LNM is beginning initialization of the specified port.

LNM.008

Level: C–INFO
Short Syntax: LNM.008 Initialization complete port *port_number* nt *network*
Long Syntax: LNM.008 Initialization complete port *port_number* network *network*
Description: LNM has completed the initialization of the specified port.

LNM.009

Level: C–INFO
Short Syntax: LNM.009 Activating LLC for port *port_number* nt *network*
Long Syntax: LNM.009 Activating LLC for port *port_number* network *network*
Description: LNM is activating the connection to LLC for the specified port.

LNM.010

Level: C–INFO
Short Syntax: LNM.010 Activating REM for port *port_number* nt *network*
Long Syntax: LNM.010 Activating REM for port *port_number* network *network*
Description: LNM is activating the Ring Error Monitor on the specified port.

LNM.011

Level: C–INFO
Short Syntax: LNM.011 Activating RPS for port *port_number* nt *network*
Long Syntax: LNM.011 Activating RPS for port *port_number* network *network*
Description: LNM is activating the Ring Parameter Server on the specified port.

LNM.012

Level: C–INFO
Short Syntax: LNM.012 Activating CRS for port *port_number* nt *network*
Long Syntax: LNM.012 Activating CRS for port *port_number* network *network*
Description: LNM is activating the Configuration Report Server for the specified port.

LNM.013

Level: C–INFO
Short Syntax: LNM.013 Activating LRM for port *port_number* nt *network*
Long Syntax: LNM.013 Activating LRM for port *port_number* network *network*
Description: LNM is activating the LAN Reporting Mechanism for the specified port.

LNM.014

Level: C–INFO

***Short Syntax*:** LNM.014 Activating MAC frame int for port *port_number* nt *network*

***Long Syntax*:** LNM.014 Activating MAC frame interface for port *port_number* network *network*

Description: LNM is activating the interface to the Token–Ring for the transfer of MAC frames to and from the specified port.

LNM.015

Level: C–INFO

***Short Syntax*:** LNM.015 Proc net up ind for port *port_number* nt *network*

***Long Syntax*:** LNM.015 Processing network up indication for port *port_number* network *network*

Description: LNM received an indication that an interface that LNM has been enabled on is now up. LNM will perform processing necessary to start processing requests to or from the interface.

LNM.016

Level: C–INFO

***Short Syntax*:** LNM.016 Proc net dwn ind for port *port_number* nt *network*

***Long Syntax*:** LNM.016 Processing network down indication for port *port_number* network *network*

Description: LNM received an indication that an interface that LNM has been enabled on is now down. LNM will terminate processing requests to or from the interface.

LNM.017

Level: UI–ERROR

***Short Syntax*:** LNM.017 No memory to queue event

***Long Syntax*:** LNM.017 Nomemory available to create an LNM event queue block

Description: Nomemory available to create an LNM event queue block. This is a fatal condition and in all probability indicates a memory leak.

LNM.018

Level: C–INFO

***Short Syntax*:** LNM.018 Rem cn req refused port *port_number* nt *network*

***Long Syntax*:** LNM.018 Remote connection request refused for port *port_number* network *network*

Description: LNM received an indication that a connection request initiated by a remote station was received. LNM does not accept remote connection requests, so the connection request will be refused.

LNM.019

Level: C–INFO

***Short Syntax*:** LNM.019 cn cnfm rcvd but not cnctg link *link* port *port_number* nt *network*

***Long Syntax*:** LNM.019 A connect confirm indication was received but the link is not in connecting state for link *link* port *port_number* network *network*

Description: LNM received an indication that a previously issued connection request has been confirmed by LLC, but the state of the link indicates that no connection request is outstanding.

Cause: The outstanding connection request may have been cancelled due to a netdown condition.

LNM.020

Level: C–INFO

***Short Syntax*:** LNM.020 disc rcvd when cnctg link *link* port *port_number* nt *network*

***Long Syntax*:** LNM.020 A disconnect indication was received while the link is in connecting state for link *link* port *port_number* network *network*

Description: LNM received a disconnect indication while a previously issued connection request is outstanding.

LNM.021

Level: C–INFO

***Short Syntax*:** LNM.021 disc rcvd but conn not act port *port_number* nt *network*

***Long Syntax*:** LNM.021 A disconnect indication was received but the connection is not active on port *port_number* network *network*

Description: LNM received an indication that the specified link has been disconnected, but the state of the link indicates that the connection is not active.

Cause: The connection may have been closed due to a netdown condition.

LNM.022

Level: C–INFO

Short Syntax: LNM.022 reset rcvd link *link* port *port_number* nt *network*

Long Syntax: LNM.022 A reset indication was received for link *link* port *port_number* network *network*

Description: LNM received a reset indication for the specified link. LNM will return a reset response.

LNM.023

Level: C–INFO

Short Syntax: LNM.023 cannot open conn SAP clsd port *port_number* nt *network*

Long Syntax: LNM.023 Cannot open a connection SAP closed on port *port_number* network *network*

Description: LNM attempted to open a connection, but found that the LNM SAP had been closed.

Cause: The SAP may have been closed due to a netdown condition.

LNM.024

Level: C–INFO

Short Syntax: LNM.024 cannot open conn lnk in use lnk link port *port_number* nt *network*

Long Syntax: LNM.024 Cannot open a connection link in use link *link* port *port_number* network *network*

Description: LNM attempted to open a connection, but found that the requested link is already in use.

LNM.025

Level: C–INFO

Short Syntax: LNM.025 open sta fld rtn = *retval* lnk *link* port *port_number* nt *network*

Long Syntax: LNM.025 Open station failed return = *retval* link *link* port *port_number* network *network*

Description: LNM attempted to open a station, but LLC rejected the operation.

LNM.026

Level: C–INFO

Short Syntax: LNM.026 conn req fld rtn = *retval* lnk *link* port *port_number* nt *network*

Long Syntax: LNM.026 Connect req failed return = *retval* link *link* port *port_number* network *network*

Description: LNM attempted to open a connection, but LLC rejected the operation.

LNM.027

Level: C–INFO

Short Syntax: LNM.027 disc req fld rtn = *retval* lnk *link* port *port_number* nt *network*

Long Syntax: LNM.027 Disconnect req failed return = *retval* link *link* port *port_number* network *network*

Description: LNM attempted to disconnect a connection, but LLC rejected the operation.

LNM.028

Level: C–INFO

Short Syntax: LNM.028 netdwn rcvd clsg LNM SAP port *port_number* nt *network*

Long Syntax: LNM.028 Netdown received closing LNM SAP port *port_number* network *network*

Description: LNM received a network down indication for the specified port. LNM is closing the LNM SAP X'F4' as a result.

LNM.029

Level: C–INFO

Short Syntax: LNM.029 netup rcvd opening LNM SAP port *port_number* nt *network*

Long Syntax: LNM.029 Netup received, opening LNM SAP port *port_number* network *network*

Description: LNM received a network up indication for the specified port. LNM is opening the LNM SAP X'F4' as a result.

LNM.030

Level: C–INFO

Short Syntax: LNM.030 No rsrc for open LNM SAP port *port_number* nt *network*

Long Syntax: LNM.030 No resources for opening LNM SAP port *port_number* network *network*

Description: LLC indicated that not enough resources exist for opening the LNM SAP X'F4'. LNM will not be enabled as a result.

LNM.031

Level: C–INFO

Short Syntax: LNM.031 LNM UI frm not sent rsn = *reason* port *port_number* nt *network*

Long Syntax: LNM.031 LNM UI LLC frame not sent reason = *reason* port *port_number* network *network*

Description: LNM attempted to send a UI frame via LLC, but the frame could not be sent for the indicated reason.

LNM.032

Level: C–INFO

Short Syntax: LNM.032 LNM UI frm not sent net dwn port *port_number* nt *network*

Long Syntax: LNM.032 LNM UI LLC frame not sent *network* down port *port_number* network *network*

Description: LNM attempted to send a UI frame via LLC, but the frame could not be sent because the network interface is down.

LNM.033

Level: C–INFO

Short Syntax: LNM.033 LNM I frm not sent conn clsd port *port_number* nt *network*

Long Syntax: LNM.033 LNM I frame not sent connection closed port *port_number* network *network*

Description: LNM attempted to send an I frame via LLC, but the frame could not be sent because the connection has been closed.

Cause: The connection may have been closed because the network interface went down.

LNM.034

Level: C–INFO

Short Syntax: LNM.034 LNM I frm not sent rsn = *reason* link *link* port *port_number* nt *network*

Long Syntax: LNM.034 LNM I frame not sent reason = *reason* link *link* port *port_number* network *network*

Description: LNM attempted to send an I frame via LLC, but the frame could not be sent for the indicated reason.

LNM.035

Level: C–INFO

Short Syntax: LNM.035 packet rcvd but no connection on link *link* port *port_number* nt *network*

Long Syntax: LNM.035 An LLC packet was received but no connection exists for link number: link port *port_number* net *network*

Description: LNM received an LLC packet for an inactive link, possibly indicating that a previously activated link has become inactive.

Cause: The previous connection request may have been cancelled due to a netdown condition.

LNM.036

Level: C–INFO

Short Syntax: LNM.036 *server* PARSE error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.036 *server* LLC parsing error, code = *error* port *port* nt *network msgptr*

Description: LNM received a LLC packet which contained a architectural syntax error and could not be parsed properly. The code defines the specific parsing failure.

Cause: Implementation error.

LNM.037

Level: C–INFO

Short Syntax: LNM.037 *server* EXECUTION error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.037 *server* EXECUTION error, code = *error* port *port* nt *network msgptr*

Description: LNM received a LLC packet which, although syntactically correct, could not be executed.

Cause: The request in the packet cannot be executed or is not supported.

LNM.038

Level: C–INFO

Short Syntax: LNM.038 *server* PCK_ALLOC error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.038 *server* packet allocation error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component was unable to allocate a packet buffer.

Cause: Either the system is very busy, or more packet buffers need to be allocated.

LNM.039

Level: C–INFO

Short Syntax: LNM.039 *server* GET_CHAR error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.039 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component was unable to obtain the next character from a LLC packet.

Cause: Implementation error.

LNM.040

Level: C–INFO

Short Syntax: LNM.040 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.040 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component was unable to perform a packet character operation. See lssdefs.h at ELS message LNM_40 for the specific error code.

Cause: Implementation error.

LNM.041

Level: C–INFO

Short Syntax: LNM.041 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.041 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component was unable to perform a buffer operation. See lssdefs.h at ELS message LNM_41 for the specific error code.

Cause: Implementation error.

LNM.042

Level: C–INFO

Short Syntax: LNM.042 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.042 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component was unable to perform a timer operation. See lssdefs.h at ELS message LNM_42 for the specific error code.

Cause: Implementation error.

LNM.043

Level: C–INFO

Short Syntax: LNM.043 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.043 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component was unable to perform a socket operation. See lssdefs.h at ELS message LNM_43 for the specific error code.

Cause: Implementation error.

LNM.044

Level: C–INFO

Short Syntax: LNM.044 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.044 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component was unable to perform memory list. See lssdefs.h at ELS message LNM_44 for the specific error code.

Cause: Implementation error.

LNM.045

Level: C–INFO

Short Syntax: LNM.045 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.045 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component was unable to perform an LSS parse operation. See lssdefs.h at ELS message LNM_45 for the specific error code.

Cause: Bad packet received from TRD.

LNM.046

Level: C–INFO

Short Syntax: LNM.046 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.046 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component received an error return status from LLC. See lssdefs.h at ELS message LNM_46 for the specific error code.

Cause: Implementation or execution error.

LNM.047

Level: C–INFO

Short Syntax: LNM.047 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.047 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component was unable to perform an LSCM operation. See lssdefs.h at ELS message LNM_47 for the specific error code.

Cause: Configuration error.

LNM.048

Level: C–INFO

Short Syntax: LNM.048 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.048 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component received an error return status from LRM. See lssdefs.h at ELS message LNM_48 for the specific error code.

Cause: Implementation or execution error.

LNM.049

Level: C–INFO

Short Syntax: LNM.049 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.049 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component received an error return status from LBS. See lssdefs.h at ELS message LNM_49 for the specific error code.

Cause: Implementation or execution error.

LNM.050

Level: C–INFO

Short Syntax: LNM.050 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.050 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component received an error return status from CRS. See lssdefs.h at ELS message LNM_50 for the specific error code.

Cause: Implementation or execution error.

LNM.051

Level: C–INFO

Short Syntax: LNM.051 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.051 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component received an error return status from REM. See lssdefs.h at ELS message LNM_51 for the specific error code.

Cause: Implementation or execution error.

LNM.052

Level: C–INFO

Short Syntax: LNM.052 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.052 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component received an error return status from RPS. See lssdefs.h at ELS message LNM_52 for the specific error code.

Cause: Implementation or execution error.

LNM.053

Level: C–INFO

Short Syntax: LNM.053 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.053 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component received an error return status from TRD. See lssdefs.h at ELS message LNM_53 for the specific error code.

Cause: Implementation or execution error.

LNM.054

Level: C–INFO

Short Syntax: LNM.054 *server* error, code = *error* port *port* nt *network msgptr*

Long Syntax: LNM.054 *server* error, code = *error* port *port* nt *network msgptr*

Description: The indicated server component received a system error return status. See lssdefs.h at ELS message LNM_54 for the specific error code.

Cause: Implementation error.

LNM.055

Level: C–INFO

Short Syntax: LNM.055 packet rcvd but no connection nt *network*

Long Syntax: LNM.055 An LLC packet was received but no connection exists for net *network*

Description: LNM received a LLC packet for an inactive link, possibly indicating that a previously activated link has become inactive. The data is discarded.

Cause: The previous connection request may have been cancelled due to a netdown condition.

LNM.056

Level: C–INFO

Short Syntax: LNM.056 packet rcvd but SAP not open nt *network*

Long Syntax: LNM.056 AN llC packet was received but the LNM SAP is not open for net *network*

Description: LNM received an LLC packet, but the LNM SAP is not open, possibly indicating that the interface has gone down. The data is discarded.

Cause: The LNM SAP may have been closed due to a netdown condition.

LNM.057

Level: U–INFO

Short Syntax: LNM.057 This LNM release supports only one LAN to one WAN bridge

Long Syntax: LNM.057 This LNM release supports only one LAN to one WAN bridge

Description: The first release of LNM (14.0a) is restricted to DNX 300 with LAN to WAN only.

Cause: User configuration error.

Action: Reconfigure bridge to be LAN to WAN or use any later release.

LNM.058

Level: C–TRACE

Short Syntax: LNM.058 LNM–*major–vector direction*, link *link*, port *port*, nt *network*

Long Syntax: LNM.058 LNM protocol message *major–vector direction*, link *link*, port *port*, network *network*

Description: This message traces all incoming and outgoing IBM LAN network Manager protocol messages. major–vectors values: <Listmajor–vectors from IBMTken–Ring Architecture Manual>. direction rcvd or sent link values: 0–3 if LINK is established. 242 for non–LINK messages (UNITDATA messages)

LNM.059

Level: C–TRACE

Short Syntax: LNM.059 MAC–*MAC–vector direction*, port *port*, nt *network*

Long Syntax: LNM.059 MAC protocol message *MAC–vector direction*, port *port*, network *network*

Description: This message traces all incoming and outgoing MAC messages. MAC–vector values: <ListMAC–vectorsfromIBMTokn–Ring Architecture Manual>. direction rcvd or sent

LNM.060

Level: UI–ERROR

Short Syntax: LNM.060 Drp LNM frm, len *frame–length*, nt *network*

Long Syntax: LNM.060 Dropping LNM frame, length *frame–length*, network *network*

Description: An incoming message is being dropped for one of two reasons. (1) The length of the frame is zero and LNM is defensively discarding the packet, or (2) LNM can not obtain an internal LNM buffer, which is never expected to happen. The length of the frame is displayed so you can tell if it reason (1) or (2). Inform customer service.

MCF

MAC Filtering

This chapter describes MAC Filtering messages. For information about message content and how to use the message, refer to the Introduction.

MCF.001

Level: P–TRACE
Short Syntax: MCF.001 MCF enbl
Long Syntax: MCF.001 MAC Filtering enabled
Description: The MAC Filtering database has been enabled.

MCF.002

Level: P–TRACE
Short Syntax: MCF.002 MCF dsbl
Long Syntax: MCF.002 MAC Filtering disabled
Description: The MAC Filtering database has been disabled.

MCF.003

Level: UI–ERROR
Short Syntax: MCF.003 MCF init–err no mem
Long Syntax: MCF.003 MAC Filtering no memory for initialization
Description: The MAC Filtering database initialization has failed to allocate memory for the MAC Filter Control structures.

MCF.004

Level: UI–ERROR
Short Syntax: MCF.004 MCF init–err bd ifc nmbr
Long Syntax: MCF.004 MAC Filtering bad interface number given in initialization
Description: The MAC Filtering database initialization has a non–existent interface configured with a filter.

MCF.005

Level: UI–ERROR

Short Syntax: MCF.005 MCF init–err gen flt db

Long Syntax: MCF.005 MAC Filtering database initialization error

Description: The MAC Filtering database initialization has encountered an error in creating the filter database.

MCF.006

Level: U–TRACE

Short Syntax: MCF.006 MCF add filt at *name* ok

Long Syntax: MCF.006 MAC Filtering initialized filter at *name* successfully

Description: The MAC Filter configured on at the given direction and interface has been successfully initialized and is in effect.

MCF.007

Level: U–TRACE

Short Syntax: MCF.007 flt filter exc frm *source–>dest,* nt *network* int *intname/intnum*

Long Syntax: MCF.007 MAC Filter filter excludes frame *source–>dest,* network *network* interface *intname*/*intnum*

Description: The specified MAC Filter has matched a frame on the given direction and interface. The frame was excluded from further processing.

MCF.008

Level: U–TRACE

Short Syntax: MCF.008 flt *filter* inc frm *source–>dest,* nt *network* int *intname*/*intnum*

Long Syntax: MCF.008 MAC Filter filter includes frame source–>dest, network *network* interface intname/intnum

Description: The specified MAC Filter has matched a frame on the given direction and interface. The frame was included in further processing.

MCF.009

Level: U–TRACE

Short Syntax: MCF.009 flt *filter* tag(tag) frm *source–>dest,* nt *network* int *intname*/*intnum*

Long Syntax: MCF.009 MAC Filter *filter* tags(tag) frame *source–>dest,* network *network* interface *intname*/*intnum*

Description: The specified MAC Filter has matched a frame on the given direction and interface. The frame was filtered according to the configured action.

Panic “mcfimem”

Short Syntax: MCF init fail, no mem

Description: The MAC Filtering initialization failed to allocate sufficient memory to complete initialization.

Action: Contact customer service.

MSPF

Multicast Extensions to OSPF

This chapter describes messages for multicast extensions to OSPF. For information about message content and how to use the message, refer to the Introduction.

MSPF.001

Level: UI–ERROR

Short Syntax: MSPF.001 No buf for IGMP poll, ifc *OSPF_interface*

Long Syntax: MSPF.001 No buffer to send IGMP Host Membership Query on interface *OSPF_interface*

Description: An IGMP Host Membership Query could not be sent out the specified interface, due to buffer shortages. No attempt will be made to send another one until the next poll interval elapses.

MSPF.002

Level: UI–ERROR

Short Syntax: MSPF.002 IGMP poll fails, ifc *OSPF_interface* rsn *failure_code*

Long Syntax: MSPF.002 Can't send IGMP Host Membership Query on interface *OSPF_interface* reason: *failure_code*

Description: An IGMP Host Membership Query could not be sent out the specified interface, due to the specified reason. No attempt will be made to send another one until the next poll interval elapses.

MSPF.003

Level: P–TRACE

Short Syntax: MSPF.003 Rcvd IGMP Report *IP_source –> IP_destination*, nt *network ID*

Long Syntax: MSPF.003 Received IGMP Host Membership Report *IP_source –> IP_destination*, net *network ID*

Description: An IGMP Host Membership Report has been received on the specified interface.

MSPF.004

Level: U–TRACE

Short Syntax: MSPF.004 No ifc for IGMP *IP_source –> IP_destination*, nt *network ID*

Long Syntax: MSPF.004 No matching interface for received IGMP, *IP_source –> IP_destination* net *network ID*

Description: An IGMP message has been received on an interface having no attached multicast–enabled OSPF interfaces. The IGMP message is discarded.

MSPF.005

Level: UE–ERROR

Short Syntax: MSPF.005 Bad IGMP xsum *IP_source –> IP_destination*, nt *network ID*

Long Syntax: MSPF.005 Received bad IGMP checksum, *IP_source –> IP_destination* net *network ID*

Description: An IGMP message has been received having a bad IGMP checksum. The message is discarded.

MSPF.006

Level: U–TRACE

Short Syntax: MSPF.006 Bad IGMP type *IP_source –> IP_destination*, nt *network ID*

Long Syntax: MSPF.006 Received bad IGMP type, *IP_source –> IP_destination* net *network ID*

Description: An IGMP message has been received having an unrecognized type field. This may be a DVMRP packet. The message is discarded.

MSPF.007

Level: UE–ERROR

Short Syntax: MSPF.007 Unexp IGMP Query *IP_source –> IP_destination*, nt *network ID*

Long Syntax: MSPF.007 Unexpected IGMP Host Membership Query, *IP_source –> IP_destination* net *network ID*

Description: An IGMP Host Membership Query has been received on an interface where the receiving router itself is sending Host Membership Queries (i.e., the router itself is the Designated Router). This is unexpected. Host Membership Queries are ignored in any case.

MSPF.008

Level: P–TRACE

Short Syntax: MSPF.008 Rcvd IGMP Query *IP_source –> IP_destination*, nt *network ID*

Long Syntax: MSPF.008 Received IGMP Host Membership Query, *IP_source –> IP_destination* net *network ID*

Description: An IGMP Host Membership Query has been received. These are ignored by MOSPF routers.

MSPF.009

Level: P–TRACE

Short Syntax: MSPF.009 Rcvd dgram *IP_source –> IP_destination*, from *receiving_interface*

Long Syntax: MSPF.009 Received IP multicast datagram, *IP_source –> IP_destination*, from *receiving_interface*

Description: An IP datagram has been received that has a class D address, indicating IP multicast. An attempt will be made to forward the datagram.

MSPF.010

Level: P–TRACE

Short Syntax: MSPF.010 Fwrd dgram *IP_source –> IP_destination*, nt *network ID*

Long Syntax: MSPF.010 Forwarded IP multicast datagram, *IP_source –> IP_destination*, net *network ID*

Description: An IP datagram has been forwarded out the specified interface as a data–link multicast.

MSPF.011

Level: P–TRACE

Short Syntax: MSPF.011 Fwrd dgram *IP_source –> IP_destination*, nbr *IP_gw_address*

Long Syntax: MSPF.011 Forwarded IP multicast datagram, *IP_source –> IP_destination*, neighbor *IP_gw_address*

Description: An IP datagram has been forwarded to a specific neighbor, as a data–link unicast.

MSPF.012

Level: P–TRACE

Short Syntax: MSPF.012 Local delivery, *IP_source –> IP_destination*

Long Syntax: MSPF.012 Local delivery of multicast datagram, *IP_source –> IP_destination*

Description: An IP datagram has been delivered to one of the router's internal applications.

MSPF.013

Level: UE–ERROR

Short Syntax: MSPF.013 Bad IP option, *IP_source –> IP_destination*

Long Syntax: MSPF.013 Multicast datagram discarded due to bad option, *IP_source –> IP_destination*

Description: An IP multicast datagram has been received, containing a bad IP option (misformatted or inappropriate for multicast). The datagram is discarded w/o returning an ICMP message.

MSPF.014

Level: UE–ERROR

Short Syntax: MSPF.014 Can't fwd *IP_source –> IP_destination*, rsn: *reason*

Long Syntax: MSPF.014 Can't forward multicast *IP_source –> IP_destination*, due to reason: *reason*

Description: An IP multicast datagram has not been forwarded, due to the specified reason.

MSPF.015

Level: P–TRACE

Short Syntax: MSPF.015 Lcl orig *IP_source –> IP_destination*

Long Syntax: MSPF.015 Locally originated multicast, *IP_source –> IP_destination*

Description: An IP datagram has been originated by one of the router's internal applications; an attempt is being made to forward it. Such datagrams are always forwarded out the interface associated with the packet source (if any), regardless of any other forwarding decision.

PPP

Point–to–Point Protocol

This chapter describes Point–to–Point Protocol messages. For information about message content and how to use the message, refer to the Introduction.

PPP.001

Level: C–INFO

Short Syntax: PPP.001 Req brng up IP, addr = *ip_address* nt *network ID*

Long Syntax: PPP.001 Request to bring up IP, local address = *ip_address*, on network *network ID*

Description: ppp_prinit routine called for IP protocol

PPP.002

Level: C–INFO

Short Syntax: PPP.002 Srl prt up, nt *network ID*

Long Syntax: PPP.002 Serial port came up successfully, on network *network ID*

Description: ppp_slftst2 routine liked the results of the load and init.

PPP.003

Level: C–INFO

Short Syntax: PPP.003 Mnt nt *network ID*

Long Syntax: PPP.003 Doing maint, on network *network ID*

Description: Entering ppp_mnt

PPP.004

Level: C–INFO

Short Syntax: PPP.004 Nt opn fr outb *protocol_name*, nt *network ID*

Long Syntax: PPP.004 Outbound data discarded, not open for protocol *protocol_name*, on network *network ID*

Description: ppp_send was called for IP data when IP state is not open (OK).

PPP.005

Level: U–INFO

Short Syntax: PPP.005 Bd IP pkt xmt typ=*type*, nt *network ID*

Long Syntax: PPP.005 Bad IP packet to transmit: type = *type*,, on network *network ID*

Description: slhc returned bad code for IP packet.

PPP.006

Level: CE–ERROR

Short Syntax: PPP.006 I_ERR on rcv nt *network ID*

Long Syntax: PPP.006 Packet received with I_ERR set, on network *network ID*

Description: ppp_in received packet with I_ERR set.

PPP.007

Level: UE–ERROR

Short Syntax: PPP.007 Rcv Bd fr addr *bad_address*, nt *network ID*

Long Syntax: PPP.007 Received packet with bad frame address = *bad_address*,, on network *network ID*

Description: ppp_in got a frame with address byte not 0xff.

PPP.008

Level: UE–ERROR

Short Syntax: PPP.008 Rcv Bd fr cntrl *bad_control*, nt *network ID*

Long Syntax: PPP.008 Received packet with bad frame control field = *bad_control*,, on network *network ID*

Description: ppp_in got a frame with control byte not = 3 (UI).

PPP.009

Level: UE–ERROR

Short Syntax: PPP.009 Rcv inv prtcl *bad_protocol*, nt *network ID*

Long Syntax: PPP.009 Received packet with invalid protocol = *bad_protocol*,, on network *network ID*

Description: ppp_in got a frame with protocol not valid (as opposed to unknown).

PPP.010

Level: CE–ERROR

Short Syntax: PPP.010 Nt opn fr inb *protocol_name*, nt *network ID*

Long Syntax: PPP.010 Inbound data discarded, not open for protocol *protocol_name*, on network *network ID*

Description: ppp_in received data when protocol state is not open.

PPP.011

Level: CE–ERROR

Short Syntax: PPP.011 Nt opn fr inb *control_protocol_name*, nt *network ID*

Long Syntax: PPP.011 Inbound *control_protocol_name*, discarded, not open for IPCP on network *network ID*

Description: ppp_in received control protocol data when LCP state is not open.

PPP.012

Level: CE–ERROR

Short Syntax: PPP.012 PAP nt supp nt *network ID*

Long Syntax: PPP.012 Received PAP packet, PAP unsupported, on network *network ID*

Description: ppp_in received a packet with PAP protocol, which we don't support.

PPP.013

Level: CE–ERROR

Short Syntax: PPP.013 prot *unsup_prot*, nt supp nt *network ID*

Long Syntax: PPP.013 Received packet with unsupported protocol *unsup_prot*, on network *network ID*

Description: ppp_in received a packet with a protocol which we don't support.

PPP.014

Level: C–TRACE

Short Syntax: PPP.014 *fsm_name,/fsm_state, routine_name*, nt *network ID*

Long Syntax: PPP.014 FSM = *fsm_name,*, state = *fsm_state,*, called *routine_name,*, on network *network ID*

Description: Called the specified fsm routine.

PPP.015

Level: UI–ERROR

Short Syntax: PPP.015 *fsm_name,/fsm_state,* snd bd cd *code,* xmt, nt *network ID*

Long Syntax: PPP.015 FSM = *fsm_name,*, state = *fsm_state,*, tried to send bad code *code,*, on network *network ID*

Description: fsm_send called to send packet with bad code.

PPP.016

Level: P–TRACE

Short Syntax: PPP.016 *fsm_name,/fsm_state,* snd *code,*, id *id,* len *len,*, nt *network ID*

Long Syntax: PPP.016 FSM = *fsm_name,*, state = *fsm_state,*, sending *code,*, id *id,*, len *len,*, on network *network ID*

Description: fsm_send about to send fsm message.

PPP.017

Level: P–TRACE

Short Syntax: PPP.017 *fsm_name,/fsm_state,* rcv *code,*, id *id,* len *len,*, nt *network ID*

Long Syntax: PPP.017 FSM = *fsm_name,*, state = *fsm_state,*, received *code,*, id *id,*, len *len,*, on network *network ID*

Description: fsm_proc received fsm message.

PPP.018

Level: CE–ERROR

Short Syntax: PPP.018 *fsm_name, msg_type,* retr exc nt *network ID*

Long Syntax: PPP.018 *fsm_name,* FSM, *msg_type,* retries exceeded, on network *network ID*

Description: Too many retries of a config request or terminate request.

PPP.019

Level: C–TRACE

Short Syntax: PPP.019 LCP/*lcp_state, routine_name,* nt *network ID*

Long Syntax: PPP.019 LCP, state = *lcp_state,*, called *routine_name,*, on network *network ID*

Description: Called the specified lcp routine.

PPP.020

Level: UE–ERROR

Short Syntax: PPP.020 Bd lcp rej id, exp *exp_id*, gt *got_id*,, nt *network ID*

Long Syntax: PPP.020 Bad LCP reject id, expected *exp_id*,, got *got_id*,, on network *network ID*

Description: lcp_rej got reject with bad id.

PPP.021

Level: UE–ERROR

Short Syntax: PPP.021 Bd lcp rej lngth, nt *network ID*

Long Syntax: PPP.021 Bad LCP reject length, on network *network ID*

Description: lcp_rej got reject with bad length.

PPP.022

Level: UE–ERROR

Short Syntax: PPP.022 Bd lcp rej opt *lcp_option*,, nt *network ID*

Long Syntax: PPP.022 Bad LCP reject option = *lcp_option*,, on network *network ID*

Description: lcp_rej got reject containing out–of–range option.

PPP.023

Level: UE–ERROR

Short Syntax: PPP.023 out–ordr lcp rej opt *lcp_option*,, nt *network ID*

Long Syntax: PPP.023 Bad LCP reject option = *lcp_option*,, on network *network ID*

Description: lcp_rej got reject containing out–of–order option.

PPP.024

Level: UE–ERROR

Short Syntax: PPP.024 Bd lcp req hdr lngth, nt *network ID*

Long Syntax: PPP.024 Bad LCP request header length, on network *network ID*

Description: lcp_req got request with bad header length.

PPP.025

Level: UE–ERROR

Short Syntax: PPP.025 Bd lcp req opt *lcp_option*, shrt, nt *network ID*

Long Syntax: PPP.025 Bad LCP request option = *lcp_option*,, data too short, on network *network ID*

Description: lcp_req got request containing option with insufficient data.

PPP.026

Level: C–TRACE

***Short Syntax*:** PPP.026 lcp req rslt: *lcp_rslt,*, opt *lcp_option,*, ln *opt_len,*, nt *network ID*

***Long Syntax*:** PPP.026 lcp request result: *lcp_rslt,*, option = *lcp_option,*, length = *opt_len,*, on network *network ID*

Description: Result, so far, of processing one option.

PPP.027

Level: UE–ERROR

***Short Syntax*:** PPP.027 Bd lcp ack id, exp *exp_id*, gt *got_id,*, nt *network ID*

***Long Syntax*:** PPP.027 Bad lcp ack id, expected *exp_id,*, got *got_id,*, on network *network ID*

Description: lcp_ack got config ack with bad id.

PPP.028

Level: UE–ERROR

***Short Syntax*:** PPP.028 Bd lcp ack lngth, nt *network ID*

***Long Syntax*:** PPP.028 Bad lcp ack length, on network *network ID*

Description: lcp_ack got config ack with bad length.

PPP.029

Level: UE–ERROR

***Short Syntax*:** PPP.029 msmtchd lcp ack, nt *network ID*

***Long Syntax*:** PPP.029 mis–matched data in lcp ack, on network *network ID*

Description: lcp_ack got ack whose data doesn't match our request.

PPP.030

Level: UE–ERROR

***Short Syntax*:** PPP.030 Bd lcp nak id, exp *exp_id*, gt *got_id,*, nt *network ID*

***Long Syntax*:** PPP.030 Bad LCP nak id, expected *exp_id,*, got *got_id,*, on network *network ID*

Description: lcp_nak got nak with bad id.

PPP.031

Level: UE–ERROR
Short Syntax: PPP.031 Bd lcp nak lngth, nt *network ID*
Long Syntax: PPP.031 Bad LCP nak length, on network *network ID*
Description: lcp_nak got nak with bad length.

PPP.032

Level: UE–ERROR
Short Syntax: PPP.032 Bd lcp nak opt *lcp_option*,, nt *network ID*
Long Syntax: PPP.032 Bad LCP nak option = *lcp_option*,, on network *network ID*
Description: lcp_nak got nak containing out–of–range option.

PPP.033

Level: UE–ERROR
Short Syntax: PPP.033 out–ordr lcp nak opt *lcp_option*,, nt *network ID*
Long Syntax: PPP.033 Bad LCP nak option = *lcp_option*,, on network *network ID*
Description: lcp_nak got nak containing out–of–order option.

PPP.034

Level: UE–ERROR
Short Syntax: PPP.034 Bd lcp nak opt *lcp_option*, shrt, nt *network ID*
Long Syntax: PPP.034 Bad LCP nak option = *lcp_option*,, data too short, on network *network ID*
Description: lcp_nak got nak containing option with insufficient data.

PPP.035

Level: P_TRACE
Short Syntax: PPP.035 mk mru *mru*
Long Syntax: PPP.035 making max receive unit with value *mru*
Description: lcp_option built mru.

PPP.036

Level: P_TRACE
Short Syntax: PPP.036 mk accm 0x*accm*
Long Syntax: PPP.036 making accm = 0x*accm*
Description: lcp_option built accm.

PPP.037

Level: P_TRACE
Short Syntax: PPP.037 mk aut 0x*auth*
Long Syntax: PPP.037 making authorization protocol with value 0x*auth*
Description: lcp_option built authorization.

PPP.038

Level: P_TRACE
Short Syntax: PPP.038 mk mag 0x*magic_number*
Long Syntax: PPP.038 making magic number with value 0x*magic_number*
Description: *lcp_option* built magic number.

PPP.039

Level: P_TRACE
Short Syntax: PPP.039 mk pfc
Long Syntax: PPP.039 making protocol compression
Description: *lcp_option* built protocol compression.

PPP.040

Level: P_TRACE
Short Syntax: PPP.040 mk acfc
Long Syntax: PPP.040 making address/control field compression
Description: *lcp_option* built address/control compression.

PPP.041

Level: P_TRACE
Short Syntax: PPP.041 mk qp 0x*protocol, period*
Long Syntax: PPP.041 making quality protocol = 0x*protocol*,, period = *period*
Description: *lcp_option* built quality.

PPP.042

Level: P_TRACE
Short Syntax: PPP.042 mk fcs
Long Syntax: PPP.042 making 32–bit fcs
Description: *lcp_option* built 32–bit fcs.

PPP.043

Level: P_TRACE
Short Syntax: PPP.043 mk lcp unk *option*
Long Syntax: PPP.043 making unknown lcp option *option*
Description: *lcp_option* built an unrecognized option.

PPP.044

Level: P_TRACE
Short Syntax: PPP.044 ck mru *mru*
Long Syntax: PPP.044 checking max receive unit with value *mru*
Description: lcp_check processed mru.

PPP.045

Level: P_TRACE
Short Syntax: PPP.045 ck accm 0x*accm*
Long Syntax: PPP.045 checking accm = 0x*accm*
Description: lcp_check processed accm.

PPP.046

Level: P_TRACE
Short Syntax: PPP.046 ck aut 0x*auth*
Long Syntax: PPP.046 checking authorization protocol with value 0x*auth*
Description: lcp_check processed authorization.

PPP.047

Level: P_TRACE
Short Syntax: PPP.047 ck mag 0x*magic_number*
Long Syntax: PPP.047 checking magic number with value 0x*magic_number*
Description: lcp_check processed magic number.

PPP.048

Level: P_TRACE
Short Syntax: PPP.048 ck pfc
Long Syntax: PPP.048 checking protocol compression
Description: lcp_check processed protocol compression.

PPP.049

Level: P_TRACE
Short Syntax: PPP.049 ck acfc
Long Syntax: PPP.049 checking address/control field compression
Description: lcp_check processed address/control compression.

PPP.050

Level: P_TRACE
Short Syntax: PPP.050 ck qp 0x*protocol, period*
Long Syntax: PPP.050 checking quality protocol = 0x*protocol*,, period = *period*
Description: lcp_check processed quality.

PPP.051

Level: P_TRACE
Short Syntax: PPP.051 ck fcs
Long Syntax: PPP.051 checking 32–bit fcs
Description: lcp_check processed 32–bit fcs.

PPP.052

Level: P_TRACE
Short Syntax: PPP.052 ck lcp unk *option*
Long Syntax: PPP.052 checking unknown lcp option *option*
Description: lcp_check processed an unrecognized option.

PPP.053

Level: C–TRACE
Short Syntax: PPP.053 *state*, *routine_name*, nt *network ID*
Long Syntax: PPP.053 state = *state*,, called *routine_name*,, on network *network ID*
Description: Called the specified cp routine.

PPP.054

Level: UE–ERROR
Short Syntax: PPP.054 Bd *control_protocol*, ack id, exp *exp_id*, gt *got_id*,, nt *network ID*
Long Syntax: PPP.054 Bad *control_protocol*, ack id, expected *exp_id*,, got *got_id*,, on network *network ID*
Description: xxcp_ack got config ack with bad id.

PPP.055

Level: UE–ERROR

Short Syntax: PPP.055 Bd *control_protocol*, ack lngth, nt *network ID*

Long Syntax: PPP.055 Bad *control_protocol*, ack length, on network *network ID*

Description: xxcp_ack got config ack with bad length.

PPP.056

Level: UE–ERROR

Short Syntax: PPP.056 msmtchd *control_protocol*, ack, nt *network ID*

Long Syntax: PPP.056 mis–matched data in *control_protocol*, ack, on network *network ID*

Description: xxcp_ack got ack whose data doesn't match our request.

PPP.057

Level: UE–ERROR

Short Syntax: PPP.057 Bd *control_protocol*, nak id, exp *exp_id*, gt *got_id*,, nt *network ID*

Long Syntax: PPP.057 Bad *control_protocol*, nak id, expected *exp_id*,, got *got_id*,, on network *network ID*

Description: xxcp_nak got nak with bad id.

PPP.058

Level: UE–ERROR

Short Syntax: PPP.058 Bd *control_protocol*, nak lngth, nt *network ID*

Long Syntax: PPP.058 Bad *control_protocol*, nak length, on network *network ID*

Description: xxcp_nak got nak with bad length.

PPP.059

Level: UE–ERROR

Short Syntax: PPP.059 Bd *control_protocol*, nak opt *ipcp_option*,, nt *network ID*

Long Syntax: PPP.059 Bad *control_protocol*, nak option = *ipcp_option*,, on network *network ID*

Description: xxcp_nak got nak containing out–of–range option.

PPP.060

Level: UE–ERROR

Short Syntax: PPP.060 out–ordr *control_protocol*, nak opt *ipcp_option*,, nt *network ID*

Long Syntax: PPP.060 Bad *control_protocol*, nak option = *ipcp_option*,, on network *network ID*

Description: xxcp_nak got nak containing out–of–order option.

PPP.061

Level: UE–ERROR

Short Syntax: PPP.061 Bd *control_protocol*, nak opt *ipcp_option*, shrt, nt *network ID*

Long Syntax: PPP.061 Bad *control_protocol*, nak option = *ipcp_option*,, data too short, on network *network ID*

Description: xxcp_nak got nak containing option with insufficient data.

PPP.062

Level: UE–ERROR

Short Syntax: PPP.062 Bd *control_protocol*, rej id, exp *exp_id*, gt *got_id*,, nt *network ID*

Long Syntax: PPP.062 Bad *control_protocol*, reject id, expected *exp_id*,, got *got_id*,, on network *network ID*

Description: xxcp_rej got reject with bad id.

PPP.063

Level: UE–ERROR

Short Syntax: PPP.063 Bd *control_protocol*, rej lngth, nt *network ID*

Long Syntax: PPP.063 Bad *control_protocol*, reject length, on network *network ID*

Description: xxcp_rej got reject with bad length.

PPP.064

Level: UE–ERROR

Short Syntax: PPP.064 Bd *control_protocol*, rej opt *ipcp_option*,, nt *network ID*

Long Syntax: PPP.064 Bad *control_protocol*, reject option = *ipcp_option*,, on network *network ID*

Description: xxcp_rej got reject containing out–of–range option.

PPP.065

Level: UE–ERROR

***Short Syntax*:** PPP.065 out–ordr *control_protocol*, rej opt *ipcp_option*,, nt *network ID*

***Long Syntax*:** PPP.065 Bad *control_protocol*, reject option = *ipcp_option*,, on network *network ID*

Description: xxcp_rej got reject containing out–of–order option.

PPP.066

Level: P_TRACE

***Short Syntax*:** PPP.066 mk ads *src_addr, dest_addr*

***Long Syntax*:** PPP.066 making IPCP addresses option, addresses = *src_addr, dest_addr*

Description: ipcp_option built (deprecated) IPCP addresses option.

PPP.067

Level: P_TRACE

***Short Syntax*:** PPP.067 mk cmp 0x*comp_protocol, slots,/slot_compress*

***Long Syntax*:** PPP.067 making compression option 0x*comp_protocol*,, slots = *slots*,, slot_compress = *slot_compress*

Description: ipcp_option built compression option.

PPP.068

Level: P_TRACE

***Short Syntax*:** PPP.068 mk ad *ip_address*

***Long Syntax*:** PPP.068 making IPCP address option, address = *ip_address*

Description: ipcp_option built ipcp address option.

PPP.069

Level: P_TRACE

***Short Syntax*:** PPP.069 mk *protocol*, unk *option*

***Long Syntax*:** PPP.069 making unknown *protocol*, option *option*

Description: {ipcp,dncp}_option built unrecognized option.

PPP.070

Level: P_TRACE

Short Syntax: PPP.070 ck ads *src_addr, dest_addr*

Long Syntax: PPP.070 checking IPCP addresses option, addresses = *src_addr, dest_addr*

Description: ipcp_check processed (deprecated) IPCP addresses option.

PPP.071

Level: P_TRACE

Short Syntax: PPP.071 ck cmp 0x*comp_protocol, slots,/slot_compress*

Long Syntax: PPP.071 checking compression option 0x*comp_protocol,*, slots = *slots,*, slot_compress = *slot_compress*

Description: ipcp_check processed compression option.

PPP.072

Level: P_TRACE

Short Syntax: PPP.072 ck ad *ip_address*

Long Syntax: PPP.072 checking IPCP address option, address = *ip_address*

Description: ipcp_check processed ipcp address option.

PPP.073

Level: P_TRACE

Short Syntax: PPP.073 ck *control_protocol*, unk *option*

Long Syntax: PPP.073 checking unknown *control_protocol*, option *option*

Description: ipcp_check processed ipcp unrecognized option.

PPP.074

Level: UE–ERROR

Short Syntax: PPP.074 Bd *control_protocol*, req hdr lngth, nt *network ID*

Long Syntax: PPP.074 Bad *control_protocol*, request header length, on network *network ID*

Description: xxcp_req got request with bad header length.

PPP.075

Level: UE–ERROR

Short Syntax: PPP.075 Bd *control_protocol*, req opt *ipcp_option*, shrt, nt *network ID*

Long Syntax: PPP.075 Bad *control_protocol*, request option = *ipcp_option,*, data too short, on network *network ID*

Description: xxcp_req got request containing option with insufficient data.

PPP.076

Level: C–TRACE

Short Syntax: PPP.076 *control_protocol*, req rslt: *ipcp_rslt*,, opt *ipcp_option*,, ln *opt_len*,, nt *network ID*

Long Syntax: PPP.076 *control_protocol*, request result: *ipcp_rslt*,, option = *ipcp_option*,, length = *opt_len*,, on network *network ID*

Description: Result, so far, of processing one option.

PPP.077

Level: UE–ERROR

Short Syntax: PPP.077 bd rcv len, pk *hdr_len*, dr *i_bxfr*, nt *network ID*

Long Syntax: PPP.077 bad length on received data, packet length = *hdr_len*,, driver says *i_bxfr*,, on network *network ID*

Description: The length field of an LCP or NCP packet didn't match the i_bxfr of the iorb.

PPP.078

Level: C–INFO

Short Syntax: PPP.078 Mdm sts chg, DCD *dcd* CTS *cts* nt *network ID*

Long Syntax: PPP.078 Modem status changed DCD = *dcd* CTS = *cts* on network *network ID*

Description: A modem status change has occurred. The present state is described.

PPP.079

Level: UE–ERROR

Short Syntax: PPP.079 prt ref rcv, prt 0x*bad_prot* nt *network ID*

Long Syntax: PPP.079 protocol reject received for protocol 0x*bad_prot* on network *network ID*

Description: Got a protocol reject packet from the link.

PPP.080

Level: UE–ERROR

Short Syntax: PPP.080 rc bd cd *packet_type*, prt *prot_type*, nt *network ID*

Long Syntax: PPP.080 Received bad code (*packet_type*,) for prot *prot_type*,, on network *network ID*

Description: A packet from the net had a type which is not supported for that protocol.

PPP.081

Level: UE–ERROR

Short Syntax: PPP.081 rc bd mgc 0x*rcv_magic_num*,, ours 0x*our_magic_num*, nt *network ID*

Long Syntax: PPP.081 Received bad magic number 0x*rcv_magic_num*,, ours is 0x*our_magic_num*,, on network *network ID*

Description: Didn't get magic number we wanted. If we got our own (the two args match) the link is looped back.

PPP.082

Level: UE–ERROR

Short Syntax: PPP.082 lpbk nt *network ID*

Long Syntax: PPP.082 link appears to be looped back on network *network ID*

Description: Excessive magic number collisions while trying to configure link.

PPP.083

Level: UI–ERROR

Short Syntax: PPP.083 Srl prt fl: 0x*status*, nt *network ID*

Long Syntax: PPP.083 Serial port failed init, stat: 0x*status*,, network *network ID*

Description: ppp_slftst2 observed bad status in (netp–>n_idctp)–>d_flg after init.

PPP.084

Level: C–INFO

Short Syntax: PPP.084 Req brng up DN nt *network ID*

Long Syntax: PPP.084 Request to bring up DECNET IV, on network *network ID*

Description: ppp_prinit routine called for Decnet IV protocol.

PPP.085

Level: UE–ERROR

Short Syntax: PPP.085 rc no mgc nt *network ID*

Long Syntax: PPP.085 Received message without a magic number, on network *network ID*

Description: A received packet which should have had a magic number (ECHO REQ, ECHO ACK, QUALITY REPORT, DISC REQ), didn't.

PPP.086

Level: C–INFO

Short Syntax: PPP.086 Req brng up IPX nt *network ID*

Long Syntax: PPP.086 Request to bring up IPX, on network *network ID*

Description: ppp_prinit routine called for IPX protocol.

PPP.087

Level: C–INFO

Short Syntax: PPP.087 Req brng up SRT nt *network ID*

Long Syntax: PPP.087 Request to bring up SRT, on network *network ID*

Description: ppp_prinit routine called for SRT protocol.

PPP.088

Level: C–INFO

Short Syntax: PPP.088 BNCP changed SR seg num from *oldsegnum* to *newsegnum* on port *port* nt *network ID*

Long Syntax: PPP.088 Bridging control protocol changed source route segment number from *oldsegnum* to *newsegnum* on port *port*, network *network ID*

Description: As a result of negotiating the source route line ID, the local side of the link changed its source route segment number.

PPP.089

Level: C–TRACE

Short Syntax: PPP.089 DROP: rcvd STB bdgd pkt but bdging dsbld on nt *network*

Long Syntax: PPP.089 Dropping the received Spanning Tree Bridged packet but bridging is disabled on network *network*

Description: A Bridged packet is received on this PPP interface even though the Spanning Tree Bridging is not enabled on this interface or STB is disabled in the box.

PPP.090

Level: C–INFO

Short Syntax: PPP.090 Req brng up AppleTalk nt *network ID*

Long Syntax: PPP.090 Request to bring up AppleTalk, on network *network ID*

Description: ppp_prinit routine called for Appletalk protocol.

PPP.091

Level: UE–ERROR

Short Syntax: PPP.091 ATCP add opt rejected on nt *network ID* – no common net num

Long Syntax: PPP.091 ATCP address option rejected on network *network ID* – no common network number

Description: An ATCPconfiguration reject will be sent because the AppleTalk Address option did not contain a common network number for the PPP link.

PPP.092

Level: UE–ERROR

Short Syntax: PPP.092 ATCP add opt rejected on nt *network ID* – remote's node ID invalid *node_id*

Long Syntax: PPP.092 ATCP address option rejected on network *network ID* – remote side's node ID is invalid *node_id*

Description: An ATCP configuration reject will be sent because the AppleTalk Address option from the remote side contained an invalid node ID.

PPP.093

Level: C–INFO

Short Syntax: PPP.093 Req brng up OSI nt *network ID*

Long Syntax: PPP.093 Request to bring up OSI, on network *network ID*

Description: ppp_prinit routine called for OSI protocol.

Panic "pppimem"

Short Syntax: PPP interface initialization failed, no memory.

Description: The PPP interface failed to allocate sufficient memory to complete initialization.

Action: Contact customer service.

Panic "pppiprt"

Short Syntax: PPP: unsupported protocol during initialization

Description: The PPP network handler detected an unsupported protocol during initialization.

Action: Contact customer service.

Panic “pppidev”

Short Syntax: PPP: wrong device type

Description: The PPP network handler detected PPP configured on a device other than I_LOUIE or I_ATC2 during init.

Action: Contact customer service.

RIP

Routing Information Protocol

This chapter describes Routing Information Protocol (RIP) messages. RIP, in this context, is the IP version (other versions are found in DDS, IPX, and XNS subsystems.)For information about message content and how to use the message, refer to the Introduction.

RIP.001

Level: UE–ERROR

Short Syntax: RIP.001 bd ver *version_number* frm hst *source_IP_address*

Long Syntax: RIP.001 bad version *version_number* received from host *source_IP_address*

Description: The version field in the RIP header did not match the current version.

Cause: This is probably caused by an error in the source host.

Action: Contact the manufacturer of the source host and report the problem.

RIP.002

Level: U–TRACE

Short Syntax: RIP.002 rq frm *source_IP_address*

Long Syntax: RIP.002 request received from host *source_IP_address*

Description: A RIP routing table request was received from another host. A routing table update will be sent to it.

RIP.003

Level: U–INFO

Short Syntax: RIP.003 trc on *tracing_file* frm *source_IP_address*

Long Syntax: RIP.003 trace on to *tracing_file* received from host *source_IP_address*

Description: A request from a host to turn RIP tracing on to a given log file was received. The router ignores this request.

RIP.004

Level: U–INFO

Short Syntax: RIP.004 trc off frm *source_IP_address*

Long Syntax: RIP.004 trace off received from host *source_IP_address*

Description: A request from a host to turn RIP tracing off was received. The router ignores this request.

RIP.005

Level: C–TRACE

Short Syntax: RIP.005 rsp frm *source_IP_address*

Long Syntax: RIP.005 response received from host *source_IP_address*

Description: A RIP routing table update was received. Note that it may take more than one response packet to transmit the entire routing table, especially if the routing table is large.

RIP.006

Level: UE–ERROR

Short Syntax: RIP.006 bd cmd *command_code* frm *source_IP_address*

Long Syntax: RIP.006 bad command code *command_code* received from host *source_IP_address*

Description: A RIP message was received with an unrecognized command code.

Cause: This is probably caused by an error or out of date software in the source host.

Action: Contact the manufacturer of the source host and report the problem.

RIP.007

Level: UE–ERROR

Short Syntax: RIP.007 rsp frm off nt *source_IP_address*

Long Syntax: RIP.007 response received from off network host *source_IP_address*

Description: A RIP routing update response was received from a machine which was not directly attached to the network the response came in on. The packet is discarded.

Cause: Since normal RIP software is generally written to send data only to connected nets, this is probably indicative of a hostile event.

Action: Examine audit trails and other information to determine the original source host.

RIP.008

Level: UE–ERROR

Short Syntax: RIP.008 sbnt rt *destination_IP_network* non–subnt intfc hst *next_hop_IP_address*

Long Syntax: RIP.008 subnet route *destination_IP_network* on non–subnetted interface from host *next_hop_IP_address*

Description: An apparent subnet route (i.e. the 'rest' field of the Internet address contained non–zero data) was received over an interface that is not marked as subnetted in the router.

Cause: This is probably caused by incorrect configuration, either in the router or in the host sending the traffic.

Action: Correct the incorrect configuration.

RIP.009

Level: U–TRACE

Short Syntax: RIP.009 dyn rt to *destination_IP_network* frm *next_hop_IP_address* dis

Long Syntax: RIP.009 dynamic route to *destination_IP_network* from *next_hop_IP_address* disallowed

Description: A dynamic route was received but is being ignored because the configuration of RIP on the router does not allow dynamic routes except for those in a table, and this route was not in that table.

RIP.010

Level: U–INFO

Short Syntax: RIP.010 nt *destination_IP_address* unrch via *next_hop_IP_address*, del

Long Syntax: RIP.010 network *destination_IP_address* now unreachable via router *next_hop_IP_address*, deleted

Description: An incoming RIP update from the router that was previously listed as the next hop to the destination network has announced that the destination is unreachable (i.e. at metric 'infinity'). The RIP route to that destination is being deleted.

RIP.011

Level: U–INFO

Short Syntax: RIP.011 updt nt *destination_IP_network* hps *metric* via *next_hop_IP_address*

Long Syntax: RIP.011 update route to net *destination_IP_network* at metric *metric* hops via router *next_hop_IP_address*

Description: A new (better) route to the given destination has been learned via RIP and has been installed.

RIP.012

Level: C–TRACE

Short Syntax: RIP.012 snd rqst *source_IP_address*

Long Syntax: RIP.012 send request from address *source_IP_address*

Description: The router is sending a RIP request from each of the addresses associated with an interface which has just come up.

RIP.013

Level: C–TRACE

Short Syntax: RIP.013 snd brd to *destination_IP_address packet_count* pkts *number_of_routes* rtes

Long Syntax: RIP.013 sending broadcast response to address *destination_IP_address* in *packet_count* packets with

Long Syntax: *number_of_routes* routes

Description: The router is sending a normal RIP broadcast update (triggered either by a timer or a change in the routing table) to the specified address.

RIP.014

Level: C–INFO

Short Syntax: RIP.014 snd to *destination_IP_address packet_count* pkts *number_of_routes* rtes

Long Syntax: RIP.014 sending response to address *destination_IP_address* in *packet_count* packets with *number_of_routes* routes

Description: The router is sending a RIP update (triggered by a request from another host) to the specified address.

RIP.015

Level: CI–ERROR

Short Syntax: RIP.015 cnt all pkt

Long Syntax: RIP.015 cannot allocate packet for transmission

Description: When RIP went to allocate a packet for transmission (either for a request or reply), none was available.

RIP.016

Level: C–TRACE

Short Syntax: RIP.016 snd pkt *destination_IP_address*

Long Syntax: RIP.016 sending packet to *destination_IP_address*

Description: A RIP packet (either a routing table update, or when an interface first comes up, a request) was sent.

RIP.017

Level: UI–ERROR

Short Syntax: RIP.017 err *output_error_code* sndng pkt nt *network*

Long Syntax: RIP.017 error code *output_error_code* when sending packet out net *network*

Description: An outgoing reply packet was dropped as the result of some problem in the router.

Cause: There are many potential causes of this problem, such as an overloaded output queue, a down network, etc.

Action: Consult logging output from the relevant *network* subsystem for more information.

RIP.018

Level: U–INFO

Short Syntax: RIP.018 nt rt to *destination_IP_address* tmd out

Long Syntax: RIP.018 network route to *destination_IP_address* timed out

Description: A route to a destination via some other router in the routing database has not been heard from for a while and is now being marked as unreachable.

RIP.019

Level: U–INFO

Short Syntax: RIP.019 nt rt to *destination_IP_address* del

Long Syntax: RIP.019 network route to *destination_IP_address* deleted

Description: A route to a destination via some other router in the routing database has not been heard from for a while, has been marked unreachable, and is now being deleted. Panic ‘‘ripudperr’’

Short Syntax: rip udp port not avail

Description: Another application registered previously with rip’s UDP port.

Action: Contact customer service.

RTMP

AppleTalk Phase 1 Routing Table Maintenance Protocol

This chapter describes AppleTalk Phase 1 Routing Table Maintenance Protocol (RTMP) messages. RTMP is part of the AppleTalk Phase 1 protocol family. For information about message content and how to use the message, refer to the Introduction.

RTMP.001

Level: CE–ERROR

Short Syntax: RTMP.001 Req frm */src_node* nt *network*, port ntwk num 0

Long Syntax: RTMP.001 Request from */src_node* net *network*, port's network number 0

Description: An RTMP Request or Route Data Request packet was received on an interface whose port network number was still zero. The request will be ignored.

Cause: Port has not yet gleaned network number from seed router.

Action: Wait until network number is gleaned.

Cause: No seed router on network for network number.

Action: Reconfigure a router to be seed.

RTMP.002

Level: P–TRACE

Short Syntax: RTMP.002 data pkt frm */src_node* nt *network*

Long Syntax: RTMP.002 data packet from */src_node* net *network*

Description: An RTMP data packet has been received.

RTMP.003

Level: U–INFO

Short Syntax: RTMP.003 nt num inferred *net_number* nt *network*

Long Syntax: RTMP.003 net number inferred *net_number* net *network*

Description: A net number has been inferred from an RTMP data packet and has been assigned to the specified interface.

RTMP.004

Level: UE–ERROR

Short Syntax: RTMP.004 nt nmbrs cnflct *net_num* frm */src_node* wth *net_num* on nt *network*

Long Syntax: RTMP.004 net numbers conflict *net_num* from */src_node* with *net_num* on nt *network*

Description: The source net number of an RTMP packet conflicts with the current known net number for the specified interface.

Cause: Configuration error in some host on the network.

Action: Make sure that only one network number is being seeded by multiple routers on the same network.

RTMP.005

Level: UE–ERROR

Short Syntax: RTMP.005 bd net *net_num* in RTMP frm *src_net/src_node*

Long Syntax: RTMP.005 bad net *net_num* in RTMP from *src_net/src_node*

Description: An illegal network number was found in an RTMP data packet from the specified router.

RTMP.006

Level: UI–ERROR

Short Syntax: RTMP.006 nt rtng tbl ovrfl, dsc *net_num*

Long Syntax: RTMP.006 network routing table overflow, discarding *net_num*

Description: Insertion of the specified net into the routing table was not performed due to routing table overflow.

Action: If the problem is chronic, reconfigure the router with a larger AppleTalk routing table.

RTMP.007

Level: U–INFO

Short Syntax: RTMP.007 rte to *net_num* via *gw_net*/*gw_node* excds max hps, disc

Long Syntax: RTMP.007 rte to *net_num* via *gw_net*/*gw_node* exceeds max hops, discarded

Description: An RTMP data packet contained a new route to the specified net, but at too large a hop count. The route was discarded.

RTMP.008

Level: U–INFO

***Short Syntax*:** RTMP.008 new rte to *net_num* via *gw_net*/*gw_node*, hops hops

***Long Syntax*:** RTMP.008 new route to *net_num* via *gw_net*/*gw_node*, hops hops

Description: A new route was added to the routing table via the indicated first hop.

RTMP.009

Level: U–INFO

***Short Syntax*:** RTMP.009 rte to *net_num* via *gw_net*/*gw_node* dltd, hopc excded

***Long Syntax*:** RTMP.009 rte to *net_num* via *gw_net*/*gw_node* deleted, hop count exceeded

Description: The route to the indicated network was deleted from the routing table due to a new route with too large a hop count.

RTMP.010

Level: U–INFO

***Short Syntax*:** RTMP.010 rte to *net_num* aged away

***Long Syntax*:** RTMP.010 rte to *net_num* aged away

Description: The route to the indicated network was deleted from the routing table due to aging.

RTMP.011

Level: UI–ERROR

***Short Syntax*:** RTMP.011 no mem RTMP brdcst nt *network*, *packet_count* pkts snt

***Long Syntax*:** RTMP.011 no memory for RTMP broadcast net *network*, *packet_count* packets sent

Description: No memory was available for a buffer to send an RTMP data packet. The reported number of packets was sent before the error occurred.

RTMP.012

Level: UI–ERROR

***Short Syntax*:** RTMP.012 brdcst disc nt *network* rsn *error_code*

***Long Syntax*:** RTMP.012 broadcast discarded net *network* reason *error_code*

Description: An outgoing RTMP broadcast packet was not successfully transmitted for the specified reason.

RTMP.013

Level: UE–ERROR

Short Syntax: RTMP.013 rqst, bd src node *src_node* nt *network*

Long Syntax: RTMP.013 Request, bad source node *src_node* net *network*

Description: An RTMP Request or Route Data Request was received with an illegal source address (0 or 255).

RTMP.014

Level: P–TRACE

Short Syntax: RTMP.014 rqst rcv frm *src_node* nt *network*

Long Syntax: RTMP.014 Request received from *src_node* net *network*

Description: An RTMP Request was received from the specified host. An RTMP Response will be sent.

RTMP.015

Level: UE–ERROR

Short Syntax: RTMP.015 rqst, unk func *RTMP_function* frm *src_node* nt *network*, disc

Long Syntax: RTMP.015 Request, unkown function *RTMP_function* from *src_node* net *network*

Description: A RTMP Request was sent with an unknown function code. The packet will be ignored.

RTMP.016

Level: UI–ERROR

Short Syntax: RTMP.016 Resp dsc nt *network* rsn *error_code*

Long Syntax: RTMP.016 Response discarded net *network* reason *error_code*

Description: A RTMP Response was not transmitted for the specified reason.

RTMP.017

Level: P–TRACE

Short Syntax: RTMP.017 brdcst nt *network* pkts *packet_count*

Long Syntax: RTMP.017 broadcast net *network* packets *packet_count*

Description: The indicated number of RTMP data packets was broadcast on the specified interface.

RTMP.018

Level: UI–ERROR

Short Syntax: RTMP.018 no mem for rtng tbl

Long Syntax: RTMP.018 no memory for routing table

Description: The router had insufficient memory for its routing table allocation.

Cause: Not enough memory to support this configuration of the router.

RTMP.019

Level: U–INFO

Short Syntax: RTMP.019 del nt *net_num* rt via *node_num* nt *network*

Long Syntax: RTMP.019 del network *net_num* route via *node_num* net *network*

Description: The route to the indicated network has been deleted from the routing table.

RTMP.020

Level: UI–ERROR

Short Syntax: RTMP.020 rtng tbl ovfl, dest *net_number*

Long Syntax: RTMP.020 routing table overflow, destination *net_number*

Description: The route to the indicated network was not inserted in the routing table due to insufficient space.

Action: Increase the configured size of the routing table.

RTMP.021

Level: UE–ERROR

Short Syntax: RTMP.021 Rqst short (*length*) frm *src_node* nt *network*

Long Syntax: RTMP.021 Request too short (*length* bytes) from *src_node* net *network*

Description: The RTMP request packet was too short to contain the required RTMP header data. The packet will be discarded.

RTMP.022

Level: UE–ERROR

Short Syntax: RTMP.022 Dta short (*length*) frm *src_node* nt *network*

Long Syntax: RTMP.022 Data packet short (*length* bytes) from *src_node* net *network*

Description: The RTMP Data or Response packet was too short to contain the required RTMP header data. The packet will be discarded.

RTMP.023

Level: UE–ERROR

Short Syntax: RTMP.023 Dta bd len (*length*) frm *src_net/src_node* nt *network*

Long Syntax: RTMP.023 Data bad length (*length* bytes) from *src_net/src_node* net *network*

Description: The RTMP Data or Response packet did not have an even (or zero) number of RTMP routing tuples. The packet will be discarded.

RTMP.024

Level: UE–ERROR

Short Syntax: RTMP.024 Dta bd ID len (*ID_length*) frm *src_net/src_node* nt *network*

Long Syntax: RTMP.024 Data bad sender's node ID length (*ID_length* bits) from *src_net/src_node* net *network*

Description: A RTMP Data or Response packet was received where the Sender's ID length was not 8 bits. This implementation requires this to be 8 bits. The packet will be discarded.

RTMP.025

Level: UE–ERROR

Short Syntax: RTMP.025 Dta ilg net 0 from *src_node* nt *network*

Long Syntax: RTMP.025 Data illegal sender's network number 0 from *src_node* net *network*

Description: A RTMP Data or Response packet was received with a sender's network number of 0. The packet will be discarded.

Cause: The sending node has a software bug. It should not be sending RTMP Data or Response when the network number is zero.

R2MP

AppleTalk Phase 2 Routing Table Maintenance Protocol

This chapter describes the AppleTalk Phase 2 Routing Table Maintenance Protocol (RTMP) messages. RTMP is part of the AppleTalk Phase 2 protocol family. For information about message content and how to use the message, refer to the Introduction.

R2MP.001

Deleted: Message deleted.

R2MP.002

Deleted: Message deleted.

R2MP.003

Level: U–INFO

*Short Syntax***:** R2MP.003 nt num inferred *net_number* nt *network*

*Long Syntax***:** R2MP.003 net number inferred *net_number* net *network*

Description: A net number has been inferred from an RTMP data packet and has been assigned to the specified interface.

R2MP.004

Level: UE–ERROR

*Short Syntax***:** R2MP.004 nt nmbrs cnflct frm *net_num/src_node* not in *net_num–net_num* on nt *network*

*Long Syntax***:** R2MP.004 net numbers conflict from *net_num/src_node* not in *net_num–net_num* on nt *network*

Description: The source net number of an RTMP packet conflicts with the current known net range for the specified interface.

Cause: Configuration error in some host on the network.

Action: Make sure that only one network range is being seeded by multiple routers on the same network.

R2MP.005

Level: UE–ERROR

***Short Syntax*:** R2MP.005 bd net *net_range* in RTMP frm *src_net/src_node*

***Long Syntax*:** R2MP.005 bad net *net_range* in RTMP from *src_net/src_node*

Description: An illegal network range was found in an RTMP data packet from the specified router.

R2MP.006

Level: UI–ERROR

***Short Syntax*:** R2MP.006 nt rtng tbl ovrfl, dsc *net_range*

***Long Syntax*:** R2MP.006 network routing table overflow, discarding *net_range*

Description: Insertion of the specified net into the routing table was not performed due to routing table overflow.

Action: If the problem is chronic, reconfigure the router with a larger AppleTalk routing table.

R2MP.007

Level: U–INFO

***Short Syntax*:** R2MP.007 rte to *net_range* via *gw_net*/*gw_node* excds max hps, disc

***Long Syntax*:** R2MP.007 rte to *net_range* via *gw_net*/*gw_node* exceeds max hops, discarded

Description: An RTMP data packet contained a new route to the specified net, but at too large a hop count. The route was discarded.

R2MP.008

Level: U–INFO

***Short Syntax*:** R2MP.008 new rte to *net_range* via *gw_net*/*gw_node*, hops hops

***Long Syntax*:** R2MP.008 new route to *net_range* via *gw_net*/*gw_node*, hops hops

Description: A new route was added to the routing table via the indicated first hop.

R2MP.009

Level: U–INFO

***Short Syntax*:** R2MP.009 rte to *net_range* via *gw_net*/*gw_node* dltd, hopc excded

***Long Syntax*:** R2MP.009 rte to *net_range* via *gw_net*/*gw_node* deleted, hopcount exceeded

Description: The route to the indicated network was deleted from the routing table due to a new route with too large a hop count.

R2MP.010

Level: U–INFO

Short Syntax: R2MP.010 rte to *net_range* aged away

Long Syntax: R2MP.010 rte to *net_range* aged away

Description: The route to the indicated network was deleted from the routing table due to aging.

R2MP.011

Level: UI–ERROR

Short Syntax: R2MP.011 no mem RTMP brdcst nt *network*, *packet_count* pkts snt

Long Syntax: R2MP.011 no memory for RTMP broadcast net *network*, *packet_count* packets sent

Description: No memory was available for a buffer to send an RTMP data packet. The reported number of packets was sent before the error occurred.

R2MP.012

Level: UI–ERROR

Short Syntax: R2MP.012 Outgng disc nt *network* rsn *error_code*

Long Syntax: R2MP.012 Outgoing discarded net *network* reason *error_code*

Description: An outgoing RTMP packet was not successfully transmitted for the specified reason.

R2MP.013

Deleted: Message deleted.

R2MP.014

Level: P–TRACE

Short Syntax: R2MP.014 rqst rcv frm *src_net/src_node* nt *network*

Long Syntax: R2MP.014 Request received from *src_net/src_node* net *network*

Description: An RTMP Request was received from the specified host. An RTMP Response will be sent.

R2MP.015

Deleted: Message deleted.

R2MP.016

Level: UI–ERROR

Short Syntax: R2MP.016 Resp dsc nt *network* rsn *error_code*

Long Syntax: R2MP.016 Response discarded net *network* reason *error_code*

Description: An RTMP Response was not transmitted for the specified reason.

R2MP.017

Level: P–TRACE

Short Syntax: R2MP.017 Snt nt *network* pkts *packet_count*

Long Syntax: R2MP.017 Sent net *network* packets *packet_count*

Description: The indicated number of RTMP data packets was sent on the specified interface.

R2MP.018

Level: UI–ERROR

Short Syntax: R2MP.018 no mem for rtng tbl

Long Syntax: R2MP.018 no memory for routing table

Description: The router had insufficient memory for its routing table allocation.

Cause: Not enough memory to support this configuration of the router.

R2MP.019

Level: U–INFO

Short Syntax: R2MP.019 del nt *net_range* rt via *net_num/node_num* nt *network*

Long Syntax: R2MP.019 del network *net_range* route via *net_num/node_num* net *network*

Description: The route to the indicated network has been deleted from the routing table.

R2MP.020

Level: UI–ERROR

Short Syntax: R2MP.020 rtng tbl ovfl, dest *net_range*

Long Syntax: R2MP.020 routing table overflow, destination *net_range*

Description: The route to the indicated network was not inserted in the routing table due to insufficient space.

Action: Increase the configured size of the routing table.

R2MP.021

Deleted: Message deleted.

R2MP.022

Deleted: Message deleted.

R2MP.023

Level: UE–ERROR

Short Syntax: R2MP.023 Dta bd len (*length*) frm *src_net/src_node* nt *network*

Long Syntax: R2MP.023 Data bad length (*length* bytes) from *src_net/src_node* net *network*

Description: The RTMP Data or Response packet did not have an even (or zero) number of RTMP routing tuples. The packet will be discarded.

R2MP.024

Level: UE–ERROR

Short Syntax: R2MP.024 Dta bd ID len (*ID_length*) frm *src_net/src_node* nt *network*

Long Syntax: R2MP.024 Data bad sender's node ID length (*ID_length* bits) from *src_net/src_node* net *network*

Description: A RTMP Data or Response packet was received where the Sender's ID length was not 8 bits. This implementation requires this to be 8 bits. The packet will be discarded.

R2MP.025

Deleted: Message deleted.

R2MP.026

Level: UE–ERROR

Short Syntax: R2MP.026 Dta bd vers (version) frm *src_net/src_node* nt *network*

Long Syntax: R2MP.026 Data bad version (version) from *src_net/src_node* net *network*

Description: The RTMP Data or Response packet did not have the correct version number (0x82) in the first RTMP routing tuple. The packet will be discarded.

R2MP.027

Level: P–TRACE

Short Syntax: R2MP.027 RDR rcv frm *src_net/src_node* nt *network*

Long Syntax: R2MP.027 Route Data Request received from *src_net/src_node* net *network*

Description: A RTMP Route Data Request or Extended Route Data Request was received from the specified host. RTMP Data will be sent.

R2MP.028

Level: UE–ERROR

Short Syntax: R2MP.028 bad netrange *net_first–net_last* nt *network* spans *net_first–net_last*

Long Syntax: R2MP.028 Bad netrange *net_first–net_last* net *network* spans *net_first–net_last*

Description: A netrange overlaps either an interface netrange or an existing net. The first netrange will be discarded.

Cause: Bad network configuration.

R2MP.029

Level: UI–ERROR

Short Syntax: R2MP.029 filtered int netrange *net_first–net_last* nt *network*

Long Syntax: R2MP.029 Filtered Interface netrange *net_first–net_last* net *network*

Description: An interface netrange is filtered by its own net filter. The interface will be disabled. The user should reconfigure either the filter or the interface netrange.

R2MP.030

Level: UE–ERROR

Short Syntax: R2MP.030 filtered net net on nt *network*

Long Syntax: R2MP.030 Filtered net net on net *network*

Description: A net was filtered by an interface net filter.

R2MP.031

Level: UE–ERROR

Short Syntax: R2MP.031 filtered netrange *net_first–net_last* frm *src_net/src_node* on nt *network*

Long Syntax: R2MP.031 Filtered netrange *net_first–net_last* from *src_net/src_node* on net *network*

Description: A netrange from another router was filtered by an interface net filter.

R2MP.032

Level: CE–ERROR

***Short Syntax*:** R2MP.032 Req frm *src_net/src_node* nt *network*, port ntwk num 0

***Long Syntax*:** R2MP.032 Request from *src_net/src_node* net *network*, port's network number 0

Description: A RTMP Request or Route Data Request packet was received on an interface whose port network number was still zero. The request will be ignored.

Cause: Port has not yet gleaned network number from seed router.

Action: Wait until network number gleaned.

Cause: No seed router on network for network number.

Action: Reconfigure a router to be seed.

R2MP.033

Level: P–TRACE

***Short Syntax*:** R2MP.033 data pkt frm *src_net/src_node* nt *network*

***Long Syntax*:** R2MP.033 data packet from *src_net/src_node* net *network*

Description: A RTMP data packet has been received.

R2MP.034

Level: UE–ERROR

***Short Syntax*:** R2MP.034 rqst, bd src node *src_net/src_node* nt *network*

***Long Syntax*:** R2MP.034 Request, bad source node *src_net/src_node* net *network*

Description: A RTMP Request or Route Data Request was received with an illegal source address (0 or 255).

R2MP.035

Level: UE–ERROR

***Short Syntax*:** R2MP.035 rqst, unk func *R2MP_function* frm *src_net/src_node* nt *network*, disc

***Long Syntax*:** R2MP.035 Request, unkown function *R2MP_function* from *src_net/src_node* net *network*

Description: A RTMP Request was sent with an unknown function code. The packet will be ignored.

R2MP.036

Level: UE–ERROR

Short Syntax: R2MP.036 Rqst short (*length*) frm *src_net/src_node* nt *network*

Long Syntax: R2MP.036 Request too short (*length* bytes) from *src_net/src_node* net *network*

Description: The RTMP request packet was too short to contain the required RTMP header data. The packet will be discarded.

R2MP.037

Level: UE–ERROR

Short Syntax: R2MP.037 Dta short (*length*) frm *src_net/src_node* nt *network*

Long Syntax: R2MP.037 Data packet short (*length* bytes) from *src_net/src_node* net *network*

Description: The RTMP Data or Response packet was too short to contain the required RTMP header data. The packet will be discarded.

R2MP.038

Level: UE–ERROR

Short Syntax: R2MP.038 ilg rtmp net 0 from *src_net/src_node* nt *network*

Long Syntax: R2MP.038 illegal rtmp net number 0 from *src_net/src_node* net *network*

Description: A RTMP Data or Response packet with a sender's network number of 0 was received. The packet will be discarded.

Cause: Sending node has software bug, should not send RTMP Data or Response when network number is zero.

SDLC

Synchronous Data Link Control

This chapter describes Synchronous Data Link Control (SDLC) protocol messages. For information about message content and how to use the message, refer to the Introduction.

SDLC.001

Level: C–INFO

Short Syntax: SDLC.001 nt *network ID* – SDLC support installed for QSL

Long Syntax: SDLC.001 SDLC supoprt installed for QSL, on network *network ID*

Description: DLSw SDLC has been initialized for operation over the serial device and is now available for use in the router.

SDLC.002

Level: C–INFO

Short Syntax: SDLC.002 dflt cfg used on stn *address*, nt *network ID*

Long Syntax: SDLC.002 No remote configuration was defined for SDLC address *Address* – default settings used, on network *network ID*

Description: An open was attempted on a remote SDLC station, but a configuration record was not defined via 'ADD REMOTE–SECONDARY'. The station will be opened using default configuration value. This station will be listed in the monitor process 'LIST REMOTE ALL' command. An asterisk '*' next to the station *address* signifies that a default configuration is in use.

SDLC.003

Level: CE–ERROR

Short Syntax: SDLC.003 no mem to copy to SDLC, nt *network ID*

Long Syntax: SDLC.003 Cannot copy a user buffer to to SDLC – Out of memory, on network *network ID*

Description: There is currently no memory available to copy user data to SDLC. Another attempt will be made at a later time.

SDLC.004

Level: U–INFO

Short Syntax: SDLC.004 frame dropped nt *network ID* not active

Long Syntax: SDLC.004 Inbound frame was dropped – SDLC not yet active on network *network ID*.

Description: A frame was received on an interface that is not yet owned by an SDLC client. All packets received are dropped until an SDLC client opens the port for use.

SDLC.005

Level: CE–ERROR

Short Syntax: SDLC.005 nt *network ID* congested – pkt droppd

Long Syntax: SDLC.005 Packet dropped due to no SDLC credit or memory shortage, on network *network ID* . Temporary.

Description: An incoming packet was dropped due to no SDLC receive credit or a temporary memory shortage.

SDLC.006

Level: CE–ERROR

Short Syntax: SDLC.006 nt *network ID* – I_ERR on rcv

Long Syntax: SDLC.006 Packet received with I_ERR set, on network *network ID*

Description: real_sdlc_in received a packet with I_ERR set

SDLC.007

Level: CE–ERROR

Short Syntax: SDLC.007 nt *network ID* – tx to dev fail (*status*)

Long Syntax: SDLC.007 Failure to send packet to device, on network *network ID*, status =*status*

Description: The call to netsend() failed while trying to send a frame from SDLC.

SDLC.008

Level: P–TRACE

Short Syntax: SDLC.008 tx *count* bytes to *address* (nt *network ID*):octets

Long Syntax: SDLC.008 Transmit to link station *count, address* bytes, on network *network ID*: octets

Description: An SDLC frame has been transmitted. This is the entire frame, including the SDLC header. All transmitted SDLC frames are logged with this message. To log only transmitted SDLC I–frames, use SDLC_53.

SDLC.009

Level: P–TRACE

Short Syntax: SDLC.009 rx *count* bytes from *address* (nt *network ID*):octets

Long Syntax: SDLC.009 Received *count* bytes from link station *address*, on network *network ID*: octets

Description: An SDLC frame has been received. This is the entire frame, including the SDLC header. All received SDLC frames are logged with this message. To log only received SDLC I–frames, use SDLC_52.

SDLC.010

Level: C–INFO

Short Syntax: SDLC.010 port ACTIVE, nt *network ID*

Long Syntax: SDLC.010 Request to bring up SDLC, on network *network ID*

Description: An entity in the router has attached to the interface and can now use SDLC services.

SDLC.011

Level: C–INFO

Short Syntax: SDLC.011 port INACTIVE, nt *network ID*

Long Syntax: SDLC.011 Request to bring down SDLC, on network *network ID*

Description: An entity in the router is no longer using SDLC services on this interface.

SDLC.012

Level: C–INFO

Short Syntax: SDLC.012 Link status: *Exception*, nt *network ID*

Long Syntax: SDLC.012 Link status change *Exception* occurred, on network *network ID*

Description: An interface signal has changed state. Note: an unwieldy number of these messages will be generated when the interface is operating in half duplex mode.

SDLC.013

Level: C–INFO

Short Syntax: SDLC.013 addr *Address* –> NRM, nt *network ID*

Long Syntax: SDLC.013 station *address* is now UP, on network *network ID*

Description: The SDLC link is now operating in Normal Response Mode, meaning that a SDLC connection is now in progress.

SDLC.014

Level: C–INFO

Short Syntax: SDLC.014 SNRM refused, addr *Address* nt *network ID*

Long Syntax: SDLC.014 Remote station refused SNRM, link station *address* remains DOWN on network *network ID*

Description: An attempt by the router to connect to a remote link station has been refused.

SDLC.015

Level: C–INFO

Short Syntax: SDLC.015 addr *Address* –> NDM, nt *network ID*

Long Syntax: SDLC.015 station *address* is now DOWN, on network *network ID*

Description: The SDLC link is now operating in Normal Disconnect Mode, meaning that a SDLC connection has been terminated in an orderly fashion.

SDLC.016

Level: U–INFO

Short Syntax: SDLC.016 Lnkstn *address* excpt *Exception*, nt *network ID*

Long Syntax: SDLC.016 Exception *Address* occurred on Link Station *Exception*, on network *network ID*

Description: The SDLC protocol has been initialized.

SDLC.017

Level: CE–ERROR

Short Syntax: SDLC.017 nt *network ID*: rx bcast on mpt line – dropped

Long Syntax: SDLC.017 network *network ID*: received a broadcast frame from a secondary station on a multipoint line – dropped

Description: SDLC received a frame to the broadcast address on a multipoint line. The frame was dropped.

SDLC.018

Level: UE–ERROR

Short Syntax: SDLC.018 SDLC not up on nt *network ID* – no LINK config

Long Syntax: SDLC.018 network *network ID*: SDLC not brought up because no LINK configuration is defined

Description: SDLC could not be initialized because there is no SDLC link configuration for this interface.

SDLC.019

Level: UE–ERROR

Short Syntax: SDLC.019 nt *network ID*: signal ctl rq failed – *reason*

Long Syntax: SDLC.019 network *network ID*: signal control request failed because *reason*

Description: SDLC could not control one or more signals on the interface. This could occur if an attempt is made to run SDLC over an unsupported interface.

SDLC.020

Level: U–INFO

Short Syntax: SDLC.020 nt *network ID* stn *address*: DLC_LINK_FAULT_CONDITION

Long Syntax: SDLC.020 network *network ID* SDLC station *address*: DLC_LINK_FAULT_CONDITION

Description: SDLC has detected a fault on the link, and all active SDLC connections on the link have been terminated. This is usually due to a loss of DSR, CTS, or DCD on a full–duplex line, or loss of DSR on a half–duplex line.

SDLC.021

Level: U–INFO

Short Syntax: SDLC.021 nt *network ID* stn *address*: DLC_RX_EXCEED_WINDOW_SIZE

Long Syntax: SDLC.021 network *network ID* SDLC station *address*: DLC_RX_EXCEED_WINDOW_SIZE

Description: SDLC has received more than the number of frames configured as the RECEIVE WINDOW before SDLC could respond.

SDLC.022

Level: U–INFO

Short Syntax: SDLC.022 nt *network ID* stn *address*: DLC_RX_LOCAL_PROTOCOL_ERROR

Long Syntax: SDLC.022 network *network ID* SDLC station *address*: DLC_RX_LOCAL_PROTOCOL_ERROR

Description: A SDLC protocol error has been detected. As a result, the SDLC connection has been terminated.

SDLC.023

Level: U–INFO

Short Syntax: SDLC.023 nt *network ID* stn *address*:
DLC_XID_RETRY_LIMIT_REACHED

Long Syntax: SDLC.023 network *network ID* SDLC station *address*:
DLC_XID_RETRY_LIMIT_REACHED

Description: The remote link station is not responding to XID frames sent by the router.

SDLC.024

Level: U–INFO

Short Syntax: SDLC.024 nt *network ID* stn *address*:
DLC_TEST_RETRY_LIMIT_REACHED

Long Syntax: SDLC.024 network *network ID* SDLC station *address*:
DLC_TEST_RETRY_LIMIT_REACHED

Description: The remote link station is not responding to TEST frames sent by the router.

SDLC.025

Level: U–INFO

Short Syntax: SDLC.025 nt *network ID* stn *address*:
DLC_SNRM_RETRY_LIMIT_REACHED

Long Syntax: SDLC.025 network *network ID* SDLC station *address*:
DLC_SNRM_RETRY_LIMIT_REACHED

Description: The remote link station is not responding to SNRM frames sent by the router. The connection attempt has failed.

SDLC.026

Level: U–INFO

Short Syntax: SDLC.026 nt *network ID* stn *address*:
DLC_POLL_RETRY_LIMIT_REACHED

Long Syntax: SDLC.026 network *network ID* SDLC station *address*:
DLC_POLL_RETRY_LIMIT_REACHED

Description: The remote link station is not responding to polls (RR or RNR) sent by the router. As a result, the connection has been terminated.

SDLC.027

Level: U–INFO

Short Syntax: SDLC.027 nt *network ID* stn *address*: DLC_RX_FRMR_INV_CTL_FIELD

Long Syntax: SDLC.027 network *network ID* SDLC station *address*: DLC_RX_FRMR_INV_CTL_FIELD

Description: SDLC has received a Frame Reject (FRMR) frame indicating that the remote link station received a frame with an invalid control field.

SDLC.028

Level: U–INFO

Short Syntax: SDLC.028 nt *network ID* stn *address*: DLC_RX_FRMR_INV_LENGTH

Long Syntax: SDLC.028 network *network ID* SDLC station *address*: DLC_RX_FRMR_INV_LENGTH

Description: SDLC has received a Frame Reject (FRMR) frame indicating that the remote link station received a frame that was too short.

SDLC.029

Level: U–INFO

Short Syntax: SDLC.029 nt *network ID* stn *address*: DLC_RX_FRMR_LONG_I_FIELD

Long Syntax: SDLC.029 network *network ID* SDLC station *address*: DLC_RX_FRMR_LONG_I_FIELD

Description: SDLC has received a Frame Reject (FRMR) frame indicating that the remote link station received a frame that was too long.

SDLC.030

Level: U–INFO

Short Syntax: SDLC.030 nt *network ID* stn *address*: DLC_RX_FRMR_INV_NR

Long Syntax: SDLC.030 network *network ID* SDLC station *address*: DLC_RX_FRMR_INV_NR

Description: SDLC has received a Frame Reject (FRMR) frame indicating that the remote link station received a frame with an invalid N(r) in the control field.

SDLC.031

Level: U–INFO

Short Syntax: SDLC.031 nt *network ID* stn *address*: DLC_RX_FRMR_NO_I_FIELD

Long Syntax: SDLC.031 network *network ID* SDLC station *address*: DLC_RX_FRMR_NO_I_FIELD

Description: SDLC has received a Frame Reject (FRMR) frame indicating that the remote link station received an I–frame with no data in the I field.

SDLC.032

Level: U–INFO

Short Syntax: SDLC.032 nt *network ID* stn *address*: DLC_RX_FRAME_INV_CTL_FIELD

Long Syntax: SDLC.032 network *network ID* SDLC station *address*: DLC_RX_FRAME_INV_CTL_FIELD

Description: SDLC has received a frame with an invalid control field.

SDLC.033

Level: U–INFO

Short Syntax: SDLC.033 nt *network ID* stn *address*: DLC_RX_FRAME_INV_LENGTH

Long Syntax: SDLC.033 network *network ID* SDLC station *address*: DLC_RX_FRAME_INV_LENGTH

Description: SDLC has received a frame that was too short.

SDLC.034

Level: U–INFO

Short Syntax: SDLC.034 nt *network ID* stn *address*: DLC_RX_FRAME_LONG_I_FIELD

Long Syntax: SDLC.034 network *network ID* SDLC station *address*: DLC_RX_FRAME_LONG_I_FIELD

Description: SDLC has received a frame that was too long.

SDLC.035

Level: U–INFO

Short Syntax: SDLC.035 nt *network ID* stn *address*: DLC_RX_FRAME_INV_NR

Long Syntax: SDLC.035 network *network ID* SDLC station *address*: DLC_RX_FRAME_INV_NR

Description: SDLC has received a frame with an invalid N(r) in the control field.

SDLC.036

Level: U–INFO

Short Syntax: SDLC.036 nt *network ID* stn *address*: DLC_RX_DM

Long Syntax: SDLC.036 network *network ID* SDLC station *address*: DLC_RX_DM

Description: SDLC has received a Disconnected Mode (DM) frame. This is sent by a remote secondary link station to indicate that a previously received DISC frame has been accepted. The link disconnection is now complete.

SDLC.037

Level: U–INFO

***Short Syntax*:** SDLC.037 nt *network ID* stn *address*: DLC_RX_RD

***Long Syntax*:** SDLC.037 network *network ID* SDLC station *address*: DLC_RX_RD

Description: SDLC has received a Request Disconnect (RD) frame. The SDLC client should respond to this by sending a Disconnect (DISC) frame.

SDLC.038

Level: U–INFO

***Short Syntax*:** SDLC.038 nt *network ID* stn *address*: DLC_RX_RIM

***Long Syntax*:** SDLC.038 network *network ID* SDLC station *address*: DLC_RX_RIM

Description: SDLC has received a Request Initialization Mode (RIM) frame. The SDLC client should respond to this by sending a Set Initialization Mode (SIM) frame.

SDLC.039

Level: U–INFO

***Short Syntax*:** SDLC.039 nt *network ID* stn *address*: DLC_LINK_INACTIVITY_DETECTION

***Long Syntax*:** SDLC.039 network *network ID* SDLC station *address*: DLC_LINK_INACTIVITY_DETECTION

Description: Reserved for possible future use.

SDLC.040

Level: U–INFO

***Short Syntax*:** SDLC.040 nt *network ID* stn *address*: DLC_TX_FRMR_INV_CTL_FIELD

***Long Syntax*:** SDLC.040 network *network ID* SDLC station *address*: DLC_TX_FRMR_INV_CTL_FIELD

Description: SDLC has entered a frame reject (FRMR) state because it has received a frame with an invalid control field.

SDLC.041

Level: U–INFO

***Short Syntax*:** SDLC.041 nt *network ID* stn *address*: DLC_TX_FRMR_INV_LENGTH

***Long Syntax*:** SDLC.041 network *network ID* SDLC station *address*: DLC_TX_FRMR_INV_LENGTH

Description: SDLC has entered a frame reject (FRMR) state because it has received a frame that was too short.

SDLC.042

Level: U–INFO

Short Syntax: SDLC.042 nt *network ID* stn *address*: DLC_TX_FRMR_LONG_I_FIELD

Long Syntax: SDLC.042 network *network ID* SDLC station *address*: DLC_TX_FRMR_LONG_I_FIELD

Description: SDLC has entered a frame reject (FRMR) state because it has received a frame that was too long.

SDLC.043

Level: U–INFO

Short Syntax: SDLC.043 nt *network ID* stn *address*: DLC_TX_FRMR_INV_NR

Long Syntax: SDLC.043 network *network ID* SDLC station *address*: DLC_TX_FRMR_INV_NR

Description: SDLC has entered a frame reject (FRMR) state because it has received a frame with an invalid N(r) in the control field.

SDLC.044

Level: U–INFO

Short Syntax: SDLC.044 nt *network ID* stn *address*: DLC_RX_SNRM_WHILE_IN_NRM

Long Syntax: SDLC.044 network *network ID* SDLC station *address*: DLC_RX_SNRM_WHILE_IN_NRM

Description: Reserved for possible future use.

SDLC.045

Level: U–INFO

Short Syntax: SDLC.045 nt *network ID* stn *address*: DLC_PORT_DISABLED

Long Syntax: SDLC.045 network *network ID* SDLC station *address*: DLC_PORT_DISABLED

Description: The interface has been disabled by the user from the SDLC console. All current connections have been terminated.

SDLC.046

Level: U–INFO

Short Syntax: SDLC.046 nt *network ID* stn *address*: DLC_PORT_ENABLED

Long Syntax: SDLC.046 network *network ID* SDLC station *address*: DLC_PORT_ENABLED

Description: The interface has been enabled by the user from the SDLC console.

SDLC.047

Level: U–INFO

Short Syntax: SDLC.047 nt *network ID*: CLOSED

Long Syntax: SDLC.047 network *network ID* SDLC link: DLC_STATION_CLOSED

Description: The interface has been closed by SDLC. SDLC is no longer running over this interface.

SDLC.048

Level: U–INFO

Short Syntax: SDLC.048 nt *network ID* stn *address*: DISABLED

Long Syntax: SDLC.048 network *network ID* SDLC station *address*: DLC_LS_DISABLED

Description: A remote link station has been disabled on this interface from the SDLC console. Any existing connection was terminated.

SDLC.049

Level: U–INFO

Short Syntax: SDLC.049 nt *network ID* stn *address*: ENABLED

Long Syntax: SDLC.049 network *network ID* SDLC station *address*: DLC_LS_ENABLED

Description: A remote link station has been enabled on this interface from the SDLC console.

SDLC.050

Level: P–TRACE

Short Syntax: SDLC.050 nt *network ID* stn *address* – rx UI:*octets*

Long Syntax: SDLC.050 network *network ID* received UI from SDLC addraddress *octets*:

Description: An Unnumbered Information (UI) frame has been received on this interface.

SDLC.051

Level: P–TRACE

Short Syntax: SDLC.051 nt *network ID* stn *address* – tx UI:*octets*

Long Syntax: SDLC.051 network *network ID* sent UI to SDLC addraddress *octets*:

Description: An Unnumbered Information (UI) frame has been transmitted on this interface.

SDLC.052

Level: P–TRACE

Short Syntax: SDLC.052 nt *network ID* – rx I on address *byte_count* bytes:*octets*

Long Syntax: SDLC.052 network *network ID* received I from SDLC addr *address byte_count octets* bytes:

Description: An Information (I) frame has been received on this interface. To log all received SDLC frames, use SDLC_9.

SDLC.053

Level: P–TRACE

Short Syntax: SDLC.053 nt *network ID* – tx I on address *byte_count* bytes:*octets*

Long Syntax: SDLC.053 network *network ID* sent I to SDLC addraddress *byte_count octets* bytes:

Description: An Information (I) frame has been transmitted on this interface. To log all received SDLC frames, use SDLC_8.

SDLC.054

Level: U–INFO

Short Syntax: SDLC.054 nt *network ID* stn *address* – MaxBTU too large for link – adjusted (*oldBTUSize –> newBTUSize*)

Long Syntax: SDLC.054 network *network ID* station *address oldBTUSize newBTUSize*, Max BTU size too large for link – adjusted (–>)

Description: The max BTU size configured for a remote link station exceeds that defined for the link. The value for the remote link station has been temporarily adjusted. To avoid this message in the future, change the max BTU size with the 'SET REMOTE MAX–PACKET' command.

SDLC.055

Level: U–INFO

Short Syntax: SDLC.055 nt *network ID* stn *address* – Rx wdw sz not compat w/modulo – adjusted (*oldRxWindow –> newRxWindow*)

Long Syntax: SDLC.055 network *network ID* station *address oldRxWindow newRxWindow*, Window size is inconsistent with modulo for link – adjusted (–>)

Description: The modulo for this link has been changed, rendering the window sizes for all pre–defined remote link stations invalid. The window size has been temporarily adjusted. When a link is configured for mod–8, the valid window sizes are 0 to 7. When the link is configured for mod–128, the valid window sizes are 8 to 128. To avoid this message in the future, change the receive window size with the 'SET REMOTE RECEIVE–WINDOW' command.

SDLC.056

Level: U–INFO

Short Syntax: SDLC.056 nt *network ID* stn *address* – Tx wdw sz not compat w/modulo – adjusted (*oldTxWindow –> newTxWindow*)

Long Syntax: SDLC.056 network *network ID* station *address oldTxWindow newTxWindow*, Window size is inconsistent with modulo for link – adjusted (–>)

Description: The modulo for this link has been changed, rendering the window sizes for all pre–defined remote link stations invalid. The window size has been temporarily adjusted. When a link is configured for mod–8, the valid window sizes are 0 to 7. When the link is configured for mod–128, the valid window sizes are 8 to 128. To avoid this message in the future, change the transmit window size with the 'SET REMOTE TRANSMIT–WINDOW' command.

SDLC.057

Level: U–INFO

Short Syntax: SDLC.057 nt *network ID* – Link cfg corrupted – using default

Long Syntax: SDLC.057 network *network ID* – Link configuration corrupted, using defaults.

Description: The link configuration was somehow corrupted, possibly due to a software upgrade. A default link configuration has been created. Before operating SDLC, review the newly– created configuration and adjust as necessary.

SDLC.058

Level: U–INFO

Short Syntax: SDLC.058 nt *network ID* – cfg XID/TEST timeout corrupted – fixed

Long Syntax: SDLC.058 network *network ID* – Configured XID/TEST timeout corrupted – fixed.

Description: An invalid XID/TEST timeout value was detected in the link configuration, possibly due to a software upgrade. It has been temporarily changed to a valid value. To avoid this message in the future, set the XID/TEST timeout from the SDLC config prompt for this interface.

SDLC.059

Level: U–INFO

Short Syntax: SDLC.059 nt *network ID* – cfg XID/TEST retry count corrupted – fixed

Long Syntax: SDLC.059 network *network ID* – Configured XID/TEST retry count corrupted – fixed.

Description: An invalid XID/TEST retry value was detected in the link configuration, possibly due to a software upgrade. It has been temporarily changed to a valid value. To avoid this message in the future, set the XID/TEST retry count from the SDLC config prompt for this interface.

SDLC.060

Level: U–INFO

Short Syntax: SDLC.060 nt *network ID* – cfg SNRM timeout corrupted – fixed

Long Syntax: SDLC.060 network *network ID* – Configured XID/TEST timeout value corrupted – fixed.

Description: An invalid SNRM timeout value was detected in the link configuration, possibly due to a software upgrade. It has been temporarily changed to a valid value. To avoid this message in the future, set the SNRM timeout from the SDLC config prompt for this interface.

SDLC.061

Level: U–INFO

Short Syntax: SDLC.061 nt *network ID* – cfg SNRM retry count corrupted – fixed

Long Syntax: SDLC.061 network *network ID* – Configured SNRM retry count corrupted – fixed.

Description: An invalid SNRM retry value was detected in the link configuration, possibly due to a software upgrade. It has been temporarily changed to a valid value. To avoid this message in the future, set the SNRM retry count from the SDLC config prompt for this interface.

SDLC.062

Level: U–INFO

Short Syntax: SDLC.062 nt *network ID* – cfg POLL timeout corrupted – fixed

Long Syntax: SDLC.062 network *network ID* – Configured POLL timeout value corrupted – fixed.

Description: An invalid POLL timeout value was detected in the link configuration, possibly due to a software upgrade. It has been temporarily changed to a valid value. To avoid this message in the future, set the POLL timeout from the SDLC config prompt for this interface.

SDLC.063

Level: U–INFO

Short Syntax: SDLC.063 nt *network ID* – cfg inter–POLL delay corrupted – fixed

Long Syntax: SDLC.063 network *network ID* – Configured inter–POLL delay value corrupted – fixed.

Description: An invalid inter–POLL delay value was detected in the link configuration, possibly due to a software upgrade. It has been temporarily changed to a valid value. To avoid this message in the future, set the inter–POLL delay from the SDLC config prompt for this interface.

SDLC.064

Level: U–INFO

Short Syntax: SDLC.064 nt *network ID* – cfg POLL retry count corrupted – fixed

Long Syntax: SDLC.064 network *network ID* – Configured POLL retry count corrupted – fixed.

Description: An invalid POLL retry value was detected in the link configuration, possibly due to a software upgrade. It has been temporarily changed to a valid value. To avoid this message in the future, set the POLL retry count from the SDLC config prompt for this interface.

SDLC.065

Level: U–INFO

Short Syntax: SDLC.065 nt *network ID* – cfg inactivity timeout corrupted – fixed

Long Syntax: SDLC.065 network *network ID* – Configured inactivity timeout value corrupted – fixed.

Description: Reserved for possible future use.

SDLC.066

Level: U–INFO

Short Syntax: SDLC.066 nt *network ID* – cfg RTS hold duration corrupted – fixed

Long Syntax: SDLC.066 network *network ID* – Configured RTS hold duration value corrupted – fixed.

Description: An invalid RTS hold value was detected in the link configuration, possibly due to a software upgrade. It has been temporarily changed to a valid value. To avoid this message in the future, set the RTS hold value from the SDLC config prompt for this interface.

SDLC.067

Level: U–INFO

Short Syntax: SDLC.067 nt *network ID* – cfg max frame size corrupted – fixed

Long Syntax: SDLC.067 network *network ID* – Configured max frame size value corrupted – fixed.

Description: An invalid maximum frame size value was detected in the link configuration, possibly due to a software upgrade. It has been temporarily changed to a valid value. To avoid this message in the future, set the maximum frame size from the SDLC config prompt for this interface.

SDLC.068

Level: C–INFO

Short Syntax: SDLC.068 nt *network ID* – link ctrs reset by usr

Long Syntax: SDLC.068 network *network ID* – SDLC link counters were reset by user.

Description: The link counters have been reset from the SDLC console.

SDLC.069

Level: U–INFO

Short Syntax: SDLC.069 nt *network ID* – cannot reset link ctrs

Long Syntax: SDLC.069 network *network ID* – SDLC link counters cannot be reset.

Description: The link counters could not be reset from the SDLC console. This is usually a temporary condition. Retry the operation.

SDLC.070

Level: C–INFO

Short Syntax: SDLC.070 nt *network ID* lnk stn *address* – link ctrs reset by usr

Long Syntax: SDLC.070 network *network ID* Link station *address* – SDLC link counters were reset by user.

Description: The counters for a remote secondary station have been reset from the SDLC console.

SDLC.071

Level: U–INFO

Short Syntax: SDLC.071 nt *network ID* lnk stn *address* – cannot reset rem ctrs

Long Syntax: SDLC.071 network *network ID* Link station *address* – SDLC link counters cannot be reset.

Description: The counters for a remote secondary station could not be reset from the SDLC console. This is usually a temporary condition. Retry the operation.

SDLC.072

Level: U–INFO

Short Syntax: SDLC.072 nt *network ID* stn *address*: CLOSED

Long Syntax: SDLC.072 network *network ID* SDLC station *address*: DLC_SAP_CLOSED

Description: The remote link station has been closed and is no longer active.

SDLC.073

Level: CE–ERROR

Short Syntax: SDLC.073 nt *network ID*: lo prio buffer alloc failed

Long Syntax: SDLC.073 network *network ID*: low priority buffer request failed

Description: SDLC failed to allocate memory from the router's heap because it has already allocated its fair share. SDLC will recover from this usually temporary problem.

SDLC.074

Level: CE–ERROR

Short Syntax: SDLC.074 nt *network ID*: hi prio buffer alloc failed

Long Syntax: SDLC.074 network *network ID*: high priority buffer request failed

Description: SDLC failed to allocate a high–priority buffer from the router's heap. As a result, SDLC will temporarily experience a loss of performance. SDLC sessions could possibly be lost if the condition persists.

SDLC.075

Level: CE–ERROR

Short Syntax: SDLC.075 nt *network ID*: buffer alloc failed – credit ok

Long Syntax: SDLC.075 network *network ID*: buffer alloc request failed, but credit is okay

Description: SDLC failed to allocate memory from the router's heap because the heap is exhausted. SDLC will recover from this usually temporary problem.

SDLC.076

Level: CE–ERROR

Short Syntax: SDLC.076 nt *network ID*: cannot xmit I–frame. Will retransmit

Long Syntax: SDLC.076 network *network ID*: cannot transmit I–frame. Will re–transmit later

Description: SDLC could not send an Information (I) frame to the interface. SDLC will attempt to re–transmit it later.

SDLC.077

Level: CE–ERROR

Short Syntax: SDLC.077 nt *network ID*: cannot xmit S–frame – dropped

Long Syntax: SDLC.077 network *network ID*: cannot transmit S–frame – dropped

Description: SDLC could not send a Supervisory (S) frame to the interface. SDLC will attempt to re–transmit it later.

SDLC.078

Level: CE–ERROR

Short Syntax: SDLC.078 nt *network ID*: cannot xmit U–frame. SDLC will recover

Long Syntax: SDLC.078 network *network ID*: cannot transmit U–frame – SDLC will recover

Description: SDLC could not send an Unnumbered Information (UI) frame to the interface. SDLC will attempt to retransmit it later.

SDLC.079

Level: CE–ERROR

Short Syntax: SDLC.079 nt *network ID*: cannot xmit XID/TEST frame. Will re–xmit

Long Syntax: SDLC.079 network *network ID*: cannot transmit U–frame – SDLC will re–transmit

Description: SDLC could not send an XID or TEST frame to the interface. SDLC will attempt to re–transmit it later.

SDLC.080

Level: CE–ERROR

Short Syntax: SDLC.080 nt *network ID*: rx inv frame type – dropped

Long Syntax: SDLC.080 network *network ID*: received invalid frame type – dropped

Description: SDLC received an invalid frame type. This frame was ignored.

SDLC.081

Level: CE–ERROR

Short Syntax: SDLC.081 nt *network ID*: rx frame from sec stn not polled – dropped

Long Syntax: SDLC.081 network *network ID*: received a frame from a secondary station that was not polled – dropped

Description: SDLC received a frame from a remote link station that it had not polled. This frame was ignored. This error may also occur if the poll response timeout is too short.

SDLC.082

Level: CE–ERROR

Short Syntax: SDLC.082 nt *network ID*: rx UI frame from sec stn w/o F–bit – dropped

Long Syntax: SDLC.082 network *network ID*: received a UI frame from a secondary station without the Final bit set – dropped

Description: SDLC received an Unnumbered Information (UI) frame without the Final (F) bit set. This frame was dropped.

SDLC.083

Level: CE–ERROR

Short Syntax: SDLC.083 nt *network ID*: rx bcast on mpt line – dropped

Long Syntax: SDLC.083 network *network ID*: received a broadcast frame from a secondary station on a multipoint line – dropped

Description: SDLC received a frame to the broadcast address on a multipoint line. The frame was dropped.

SDLC.084

Level: UE–ERROR

Short Syntax: SDLC.084 SDLC not up on nt *network ID* – no LINK config

Long Syntax: SDLC.084 network *network ID*: SDLC not brought up because no LINK configuration is defined

Description: SDLC could not be initialized because there is no SDLC link configuration for this interface.

SDLC.085

Level: UE–ERROR

Short Syntax: SDLC.085 nt *network ID*: signal ctl rq failed – reason

Long Syntax: SDLC.085 network *network ID*: signal control request failed becausereason

Description: SDLC could not control one or more signals on the interface. This could occur if an attempt is made to run SDLC over an unsupported interface.

SDLC.086

Level: CE–ERROR

Short Syntax: SDLC.086 HDX, DCD hi during xmit, nt *network ID*

Long Syntax: SDLC.086 HDX, DCD went high during HDX transmit, on network *network ID*

Description: DCD went high during transmission of a frame in half duplex mode. This is a protocol violation, and the interface will go down shortly in order to correct the problem.

SDLC.087

Level: C–INFO

Short Syntax: SDLC.087 HDX, CTS now low. Premature DCD recovery complete, nt *network ID*

Long Syntax: SDLC.087 HDX, CTS now low. Premature DCD recovery complete, on network *network ID*

Description: The CTS signal has transitioned to low while the interface was recovering from a half–duplex protocol violation. The link is now in the correct state and data transfer may resume.

SDLC.088

Level: CE–ERROR

Short Syntax: SDLC.088 HDX, unsolicited signal while idle, nt *network ID*

Long Syntax: SDLC.088 HDX, unsolicited signal while idle, on network *network ID*

Description: A signal was asserted by the connected device while the line was idle. When operating in half–duplex mode, only the DTR/DSR signal should be asserted on an idle interface.

SDLC.089

Level: CE–ERROR

Short Syntax: SDLC.089 HDX, DSR or CTS low during xmit, nt *network ID*

Long Syntax: SDLC.089 HDX, DSR or CTS went low during HDX transmit, on network *network ID*

Description: The DSR or CTS signal went low during transmission. This is a protocol violation, and the interface will go down shortly in order to correct the problem.

SDLC.090

Level: CE–ERROR

Short Syntax: SDLC.090 nt *network ID*: rx inv frame type while closing link address – dropped

Long Syntax: SDLC.090 network *network ID*: received inappropriate frame while closing linkaddress – dropped

Description: SDLC received a frame other than a UA while trying to close the link station. This frame was ignored.

SDLC.091

Level: CE–ERROR

*Short Syntax***:** SDLC.091 nt *network ID*: tx *frame_size* byte frame too large for link stn *address* – refused

*Long Syntax***:** SDLC.091 network *network ID*: Attempt to transmit *frame_size* address byte frame refused – too large for link station

Description: An attempt was made by an SDLC user to transmit a frame that is larger than the value specified in the SDLC configuration for that link station. To specify a larger frame size, use the 'SET REMOTE–SECONDARY MAX–PACKET' command from the SDLC Configuration or Console process. If the configuration was created automatically by DLSw, use the 'ADD REMOTE–SECONDARY' command to modify the maximum packet size.

SL

Serial Line Network

This chapter describes Serial Line (SL) network interface messages. For information about message content and how to use the message, refer to the Introduction.

SL.001

Level: CI–ERROR

Short Syntax: SL.001 no bfr avl slftst nt *network*

Long Syntax: SL.001 no buffer available for selftest network *network*

Description: A packet buffer was not available when the interface selftest needed one.

SL.002

Level: CE–ERROR

Short Syntax: SL.002 modem err no DCD/CTS slftst nt *network*

Long Syntax: SL.002 modem error with no DCD/DTS detected during selftest network *network*

Description: During the self-test, the interface card detected that certain signals from modem necessary to indicate that it is present and ready to operate were not detectable.

SL.003

Level: CE–ERROR

Short Syntax: SL.003 no echo ack tmout slftst nt *network*

Long Syntax: SL.003 no echo acknowledgement received, timeout selftest network *network*

Description: During self-test, an Echo Acknowledgement was not received from the other end of the line within the timeout period. The network fails self-test.

SL.004

Level: U–INFO

Short Syntax: SL.004 bd ln DN act *actual_length*, clm *claimed_length* nt *network*

Long Syntax: SL.004 bad length in DECNet packet actual *actual_length* claimed *claimed_length* network *network*

Description: A bad length field or truncated packet was detected in a DECnet packet on the serial line.

SL.005

Level: U–INFO

Short Syntax: SL.005 bd rcv pkt hdr opc *opcode* nt *network*

Long Syntax: SL.005 bad received packet header opcode *opcode* network *network*

Description: A network control packet was received with a packet type which did not match any known opcode.

SL.006

Level: U–INFO

Short Syntax: SL.006 bd rcv pkt hdr prt *protocol* nt *network*

Long Syntax: SL.006 bad received packet header protocol *protocol* network *network*

Description: A serial line packet was received with a higher level protocol which did not match any known protocol.

SL.007

Level: U–TRACE

Short Syntax: SL.007 st slftst nt *network*

Long Syntax: SL.007 start selftest network *network*

Description: self-test is being started on the serial line.

SL.008

Level: C–TRACE

Short Syntax: SL.008 snd ech nt *network*

Long Syntax: SL.008 send echo packet network *network*

Description: An echo request is being sent on the serial line.

SL.009

Level: C–TRACE

Short Syntax: SL.009 rcv tst nt *network*

Long Syntax: SL.009 receive test packet network *network*

Description: A test packet has been received on the serial line.

SL.010

Level: C–TRACE

Short Syntax: SL.010 rcv ack nt *network*

Long Syntax: SL.010 receive echo acknowledgement packet network *network*

Description: An echo acknowledgement packet has been received on the serial line.

SL.011

Level: C–TRACE

Short Syntax: SL.011 rcv req nt *network*

Long Syntax: SL.011 receive echo request packet network *network*

Description: An echo acknowledgement packet has been received on the serial line.

SL.012

Level: C–TRACE

Short Syntax: SL.012 snd tst nt *network*

Long Syntax: SL.012 sending maintenance test packet network *network*

Description: A test packet used for network maintenance is being sent on the serial line.

SL.013

Level: C–TRACE

Short Syntax: SL.013 DROP: rcvd STB bdgd pkt but bdging dsbld on nt *network*

Long Syntax: SL.013 Dropping the received Spanning Tree Bridged packet but bridging is disabled on network *network*

Description: A Bridged packet is received on this serial line interface even though the Spanning Tree Bridging is not enabled on this interface or STB is disabled in the box.

SL.014

Level: U–TRACE

Short Syntax: SL.014 Dual Ser bd slot *slot_num* fware dwnld *ld_code*

Long Syntax: SL.014 Dual Serial Line board in slot *slot_num* firmware download *ld_code*

Description: Dual Serial Line Interface board firmware download has completed with the status shown.

SL.015

Level: U–TRACE

Short Syntax: SL.015 Dual Ser bd slot *slot_num* PUD stat *pud_stat*

Long Syntax: SL.015 Dual Serial Line board slot *slot_num* Power–On Diagnostics status *pud_stat*

Description: Dual Serial Line Board Power–On Diagnostics status completed with the code shown. See Power–On Diagnostics manual for encoding.

SL.016

Deleted: Message deleted.

SL.017

Deleted: Message deleted.

SL.018

Deleted: Message deleted.

SL.019

Level: UE–ERROR

Short Syntax: SL.019 cbl typ *cable_type* nt compt wth lvl cnvt typ *level_converter_type*, nt *network*

Long Syntax: SL.019 Cable of type *cable_type* is not compatible with level converter of type *level_converter_type*, network *network*

Description: The cable and the level converter on the interface are not compatible with each other. The self-test will fail.

Cause: Wrong cable type for level converter.

Action: Use correct cable type.

Cause: If cable_type is “none”, no cable.

Action: Connect adapter cable.

Cause: Cable broken so that it does not indicate cable type correctly (very unlikely).

Action: Replace cable.

SL.020

Level: UI–ERROR

Short Syntax: SL.020 *cable_type* can't be usd wth *internal_external* clk, nt *network*

Long Syntax: SL.020 *cable_type* cable cannot be used with *internal_external* clocking enabled, network *network*

Description: There is an incompatibility between the mode of the cable (DCE or DTE) and the type of clocking used. The interface will not be brought up.

Cause: DTE cable with internal clocking.

Action: Use DCE cable, or external clocking.

Cause: DCE cable with external or mixed clocking.

Action: Use DTE cable, or internal clocking.

SL.021

Level: CE–ERROR

Short Syntax: SL.021 slf tst fl bcs of mdm sts, CTS = *cts*, DSR = *dsr*, DCD = *dcd*, nt *network*

Long Syntax: SL.021 Self test failed because of modem status, CTS = *cts*, DSR = *dsr*, DCD = *dcd*, nt *network*

Description: The interface failed self test because at least one of the modem signals was off.

Cause: Cable not connected to modem.

Action: Connect cable.

Cause: Modem not powered up.

Action: Power up modem.

Cause: Modem does not have good connection to other end of line (especially DCD OFF).

Action: Solve modem problem.

SL.022

Level: C–INFO

Short Syntax: SL.022 Modem status change CTS = *cts*, DSR = *dsr*, DCD = *dcd*, nt *network ID*

Long Syntax: SL.022 Modem status change CTS = *cts*, DSR = *dsr*, DCD = *dcd*, on network *network ID*

Description: A modem status change has occurred. The present state is described.

SL.023

Level: CE–ERROR

Short Syntax: SL.023 int dwn bcs of mdm sts, CTS = *cts*, DSR = *dsr*, DCD = *dcd*, nt *network*

Long Syntax: SL.023 Interface down because of modem status, CTS = *cts*, DSR = *dsr*, DCD = *dcd*, nt *network*

Description: The interface was brought down because one of the modem signals was off.

SL.024

Level: UI–ERROR

Short Syntax: SL.024 conf frame sz *configured_size* too large, reducing to *maximum_size*, nt *network*

Long Syntax: SL.024 Configured frame size of *configured_size* bytes too large, reducing to *maximum_size* bytes, network *network*

Description: The user–configured frame size for this interface is larger than the maximum that is allowed for the particular serial line device. The size is reduced to the largest allowable one.

Cause: Configuration in excess of allowable size.

Action: Reconfigure size and restart.

SL.025

Level: U–INFO

Short Syntax: SL.025 bad len SRT actual=*actual_length*, claim=*claimed_length* nt *network*

Long Syntax: SL.025 bad length in SRT packet actual *actual_length* claimed *claimed_length* network *network*

Description: A bad length field or truncated packet was detected in a SRT bridge packet

SL.026

Level: U–INFO

Short Syntax: SL.026 Packet shorter than SL hdr, actual=*actual_length*, nt *network*

Long Syntax: SL.026 Packet received is shorter than Serial line header, actual *actual_length* network *network*

Description: An Incoming Serial Line packet is shorter than a minimally required Serial Line header. Such packet can not be processed and hence freed.

SL.027

Level: UI–ERROR

Short Syntax: SL.027 No level conv, disabling nt *network*

Long Syntax: SL.027 No level converter, disabling network *network*

Description: There is no level converter on this port of the Quad Serial Line card. The self-test will fail, and future self-tests will be cancelled.

Cause: No level converter.

Action: Add level converter to port.

Cause: Defective level converter which reads as not installed.

Action: Replace level converter.

SL.028

Level: UI–ERROR

Short Syntax: SL.028 Unk level conv *converter_type*, disabling nt *network*

Long Syntax: SL.028 Unknown level converter type *converter_type*, disabling network *network*

Description: There is a level converter of an unknown type on this port of the Quad Serial Line card. The self-test will fail, and future self-tests will be cancelled.

Cause: Unknown type of level converter.

Action: Upgrade to newer software that supports this type of level converter.

Cause: Defective level converter which reads as unknown type.

Action: Replace level converter.

SL.029

Level: UI–ERROR

Short Syntax: SL.029 conf frame sz *configured_size* too small, increasing to *minimum_size*, nt *network*

Long Syntax: SL.029 Configured frame size of *configured_size* bytes too small, increasing to *minimum_size* bytes, network *network*

Description: The user–configured frame size for this interface is smaller than the minimum that is required for the particular serial line device. The size is increased to the minimum allowable one.

Cause: Configuration below minimum allowable size.

Action: Reconfigure size and restart.

SL.030

Level: UI–ERROR

Short Syntax: SL.030 Maint buff busy, nt *network*

Long Syntax: SL.030 Maintenance buffer busy, network *network*

Description: The maintenance routine is attempting to send a maintenance frame for Serial data–link, but the buffer for sending this has not been freed since the previous maintenance. If this happens 3 times in a row, maintenance will fail at the other end of the serial line.

Cause: Device hung.

Action: Device will fail maintenance, and then self-test, which may provide more diagnostic information.

Cause: Device has greater than a 4 second backlog of traffic to send.

Action: Reduce congestion.

SL.031

Level: CI–ERROR

Short Syntax: SL.031 No pkt fr read cntrs nt *network*

Long Syntax: SL.031 No packet for reading counters, network *network*

Description: A buffer could not be allocated when needed to read the statistics counters from the Dual Serial interface. The counters should be successfully updated at the next maintenance, 4 seconds later.

Cause: This may be caused by temporary traffic loads. Many other causes are possible.

Action: If the message persists, contact customer service.

SL.032

Level: UI–ERROR

Short Syntax: SL.032 Pkt buf alloc # *buffer_number* failed, nt *network*

Long Syntax: SL.032 Packet buffer allocation number *buffer_number* failed, network *network*

Description: The allocation of a packet buffer for a DNX 300 serial line failed. This may be due to a shortage of either buffer memory or heap memory. (See the output of ELS messages GW.061, GW.063, and GW.064.) The interface will still try and come up, but with a reduced number of receive packet buffers.

Cause: Shortage of buffer memory. (Particularly if preceded by ELS message GW.064.)

Action: Upgrade size of buffer memory.

Action: Choose smaller buffer size on the interface.

Cause: Shortage of heap memory. (Particularly if preceded by ELS message GW.063.)

Action: Reduce routing table sizes. Increase size of data memory.

Cause: Buffer allocation routine did not accurately predict how many buffers could be allocated.

Action: On some configurations, some portions of the buffer memory are unusable. The pre–allocator does not take this into account, so a few buffer allocations may fail.

SL.033

Level: UI–ERROR

Short Syntax: SL.033 Pkt buff # *buffer_number* crossed page, dropped, nt *network*

Long Syntax: SL.033 Packet buffer number *buffer_number* crossed 64 Kbyte page boundary, dropped, nt *network*

Description: A packet buffer was allocated for the DNX 300 serial line, but that buffer crossed a 64 Kbyte memory boundary in the buffer memory. Since all of the receive buffers for a DNX 300 serial line must be in the same 64 Kbyte buffer memory packet, that buffer will be ignored, and no further buffers will be allocated for the device. The interface will still try and come up, but with a reduced number of receive packet buffers.

Cause: System buffer size overhead so great that 15 buffers with 4095 bytes of data cannot be allocated in one 64 Kbyte page in buffer memory. The DNX 300 serial line normally requests 15 receive buffers. This is the maximum number that can be allocated from one 64 Kbyte page with the maximum size. If the header, wrap, or trail size, if particularly large, the 15 packet buffers may not fit into one 64 Kbyte page. (See the ‘‘Buffer sizes Total’’ column in the +BUFFERS command.)

Action: If buffer_number is close to 15, there should not be a serious performance degradation, and no action is required.

Action: If more buffers are needed, there may be some reconfiguration of the router which can provide for more buffers. If tunneling is not needed, turn it off to reduce the ‘‘Wrap’’ size. If the buffer size on the serial line can be reduced, that should increase the number of buffers.

Action: If there is no configuration that allows an adequate number of receive buffers, upgrade to a DNX 350 serial line, which does not have this 64 Kbyte restriction.

Panic “slbdtbl”

Short Syntax: SL: sl_llc tbl out of date

Description: The Serial line LLC table is out of date.

Action: Contact Customer Service.

Panic “slnsl”

Short Syntax: SL: sl_prinit clld, nt SL

Description: The initialization routine for the serial line handler was called with a network which was not a serial line.

Cause: Probably a software generation error.

Action: Contact Customer Service.

Panic “slbdp”

Short Syntax: SL: bd prt fr SL

Description: The initialization routine for the serial line handler was called for a protocol which the serial line does not support.

Cause: Probably a software generation error.

Action: Contact Customer Service.

Panic “slnsrly”

Short Syntax: SL: srly_prinit clld, nt SRLY

Description: The initialization routine for the serial line handler was called with a network which was not a SDLC relay line.

Cause: Probably a software generation error.

Action: Contact Customer Service.

Fatal “slupo”

Short Syntax: SL: Unkwn prot SL out

Description: An unknown protocol was seen on a request to send a packet.

Panic “slccbfm”

Short Syntax: SL: can’t create buffer for maintenance

Description: The Serial Line handler was unable to allocate a permanent buffer for sending maintenance packets.

Cause: Severe heap memory shortage.

Action: Reconfigure to use less memory, or upgrade memory size.

SNMP

Simple Network Management Protocol

This chapter describes Simple Network Management Protocol (SNMP) messages. SNMP is the IP network management protocol. For information about message content and how to use the message, refer to the Introduction.

SNMP.001

Level: P–TRACE

Short Syntax: SNMP.001 rcvd pkt frm hst *source_address*

Long Syntax: SNMP.001 received packet from host *source_address*

Description: This message is generated for each SNMP packet received from a remote host.

SNMP.002

Level: P–TRACE

Short Syntax: SNMP.002 snt pkt to hst *dest_addr*ess

Long Syntax: SNMP.002 sent packet to host *dest_addr*ess

Description: This message is generated for each SNMP packet sent to a remote host.

SNMP.003

Level: UE–ERROR

Short Syntax: SNMP.003 rcvd non–SNMP pkt frm hst *source_address* (err= *value*)

Long Syntax: SNMP.003 received non–SNMP packet from host *source_address* (error code = *value*)

Description: This message is generated by a first–level reasonableness check of an incoming SNMP packet. The error codes have the following meanings: 1 – packet does not begin with SEQUENCE (0x30) 2 – packet sequence length too small 3 – packet sequence length improperly encoded (in one byte) 4 – first packet field not an ASN.1 INTEGER 5 – packet sequence length improperly encoded (in two bytes) 6 – first packet field not an ASN.1 INTEGER 7 – some other error was detected

Cause: Another node on the network sent an improperly formed SNMP packet to the router.

Action: Examine the remote node, specified in the error message, for errors.

SNMP.004

Level: UE–ERROR

Short Syntax: SNMP.004 bad ver version frm hst *source_address*

Long Syntax: SNMP.004 bad version number version from host *source_address*

Description: This message indicates that an SNMP packet contained an incorrect version number.

Cause: Either the router or the Network Manager is running an incompatible version.

Action: Update (or back out) one version of SNMP.

Cause: A bad packet slipped through the first–level error checks.

Action: Check the network for wild packets.

SNMP.005

Level: U–TRACE

Short Syntax: SNMP.005 no access: comm "*community*", hst *source_address*

Long Syntax: SNMP.005 no access to community "*community*" from host *source_address*

Description: This message indicates that an SNMP request from a remote host specified a community which does not exist or a community which did not list that host's IP address as acceptable.

Cause: The remote host is using the wrong community name.

Action: Update the remote hosts Network Manager.

Cause: The defined community in the router is incorrect.

Action: Correct the community name or add the remote host's IP address to the community's list.

SNMP.006

Level: UE–ERROR

Short Syntax: SNMP.006 bad appl type *appl_type* frm hst *source_address*

Long Syntax: SNMP.006 bad application type *appl_type* from host *source_address*

Description: This message indicates that an SNMP packet had a bad application type. That is, it was not a GET, GETNEXT or SET request.

Cause: The remote host is in error (perhaps sending response packets).

Action: Check the remote host.

Cause: A bad packet slipped through the first–level error checks.

Action: Check the network for wild packets.

SNMP.007

Level: UI–ERROR

Short Syntax: SNMP.007 no free pkr bfr

Long Syntax: SNMP.007 no packet buffer available

Description: This message is generated when SNMP cannot allocate a packet in which to construct an SNMP response.

SNMP.008

Level: U–TRACE

Short Syntax: SNMP.008 R/O access for SET: hst *source_address*, comm "*community*"

Long Syntax: SNMP.008 SET request from host *source_address* has read–only access on community "*community*"

Description: This message indicates that a SET request came in on a community which only provides read–only access to the MIB.

Action: Provide a community which allows sets or get the remote host to stop sending SET requests.

SNMP.009

Level: UI–ERROR

Short Syntax: SNMP.009 TRAP: no free pkt bfr

Long Syntax: SNMP.009 TRAP: no free packer buffer available

Description: This message is generated whenever SNMP cannot send an event because it cannot allocate storage.

SNMP.010

Level: P–TRACE

Short Syntax: SNMP.010 FAKE: pkt dscrd frm hst *source_address*

Long Syntax: SNMP.010 FAKE: packet discarded from host *source_address*

Description: This message is generated by the SNMP fake–out routine.

Cause: An SNMP packet arrived and the router does not have SNMP support.

SNMP.011

Level: U–TRACE

Short Syntax: SNMP.011 FAKE: EGP neighbor *IP_address* lost

Long Syntax: SNMP.011 FAKE: EGP neighbor *IP_address* lost

Description: This message is generated by the SNMP, EGP neighbor loss fake–out routine.

Cause: EGP tried to generate a neighbor loss event but the router does not have SNMP installed.

SNMP.012

Level: C–TRACE

Short Syntax: SNMP.012 comm *name* added

Long Syntax: SNMP.012 community *name* added

Description: This message is generated by the SNMP configuration routine when it reads a new community in from SRAM.

SNMP.013

Level: UE–ERROR

Short Syntax: SNMP.013 rcvd non–SNMP pkt frm hst *source_address*

Long Syntax: SNMP.013 received non–SNMP packet from host *source_address*

Description: This message is generated by a reasonableness check of an incoming SNMP packet. This check is done just before processing the PDU.

Cause: Another node on the network sent an improperly formed SNMP packet to the router, and the packet slipped through the first level error checks.

Action: Examine the remote node, specified in the error message, for errors.

SNMP.014

Level: UE–ERROR

Short Syntax: SNMP.014 bad ovarlen *source_address* frm hst *ovarlen*

Long Syntax: SNMP.014 length of variable to be sent out exceeds max length *source_address* from host *ovarlen*

Description: This message is generated by a length check on the variable to be sent out.

Cause: Another node on the network sent an improperly formed SNMP packet to the router, and the packet slipped through the first level error checks.

Action: Examine the remote node, specified in the error message, for errors.

SNMP.015

Level: P–TRACE

Short Syntax: SNMP.015 rcvd get–req pkt frm hst *source_address*

Long Syntax: SNMP.015 received a get–request packet from host *source_address*

Description: This message is generated for each SNMP packet received from a remote host of the type get–request.

SNMP.016

Level: P–TRACE

Short Syntax: SNMP.016 rcvd get–nxt pkt frm hst *source_address*

Long Syntax: SNMP.016 received a get–next packet from host *source_address*

Description: This message is generated for each SNMP packet received from a remote host of the type get–next.

SNMP.017

Level: P–TRACE

Short Syntax: SNMP.017 rcvd set–req pkt frm hst *source_address*

Long Syntax: SNMP.017 received a set–request packet from host *source_address*

Description: This message is generated for each SNMP packet received from a remote host of the type set–request.

SNMP.018

Level: U–TRACE

Short Syntax: SNMP.018 pkt frm hst : *source_address* caused err typ toobig

Long Syntax: SNMP.018 packet from host *source_address* resulted in a pkt with error status: toobig

Description: This message indicates that a packet was sent out with the error status as too big as a result of the SNMP variable in question not fitting the packet size.

Action: Increase the packet–size.

SNMP.019

Level: U–TRACE

Short Syntax: SNMP.019 pkt frm hst : *source_address* caused err typ nosuchnam

Long Syntax: SNMP.019 packet from host *source_address* resulted in a pkt with error status: nosuchnam

Description: This message indicates that a packet was sent out with the error status as noSuchName as a result of the SNMP variable in question not existing in the gateway or not in the view associated with the specified community or the operation is a set on a read–only variable.

Action: Ensure that the requested variable exists in the gateway (also possibly the particular instance of the variable), it is in the requested view, the community has the correct access type and the requested variable is writeable if it is a set operation.

SNMP.020

Level: U–TRACE

Short Syntax: SNMP.020 pkt frm hst : *source_address* caused err typ bad– value

Long Syntax: SNMP.020 packet from host *source_address* resulted in a pkt with error status: badvalue

Description: This message indicates that a packet was sent out with the error status as badvalue as a result of trying to set a variable with a wrong value specified in the SET request.

Action: Ensure that the SET request from the remote host specifies a value consistent with the ASN1 type of the value that it is attempting to set.

Panic "nmnostor"

Short Syntax: SNMP: no storage for MIB

Description: No storage was available to add an entry to the MIB.

Panic "nmitype"

Short Syntax: SNMP: interface type not defined for net

Description: The structure that defines an interface does not define a value for the MIB–II ifType variable.

Action: Contact customer service for a new load. Do not try and enable SNMP with this load.

Panic "snmpudperr"

Short Syntax: snmp udp port not avail

Description: Another application registered previously with snmp's UDP port.

Action: Contact customer service.

SPF

Open Shortest Path First Protocol

This chapter describes Open Shortest Path First (OSPF) Protocol messages. OSPF is a routing protocol for the IP protocol. For information about message content and how to use the message, refer to the Introduction.

SPF.001

Level: UE–ERROR

Short Syntax: SPF.001 Bad length pkt, from *IP_source*, type *OSPF_packet_type*

Long Syntax: SPF.001 Bad length packet, from *IP_source*, type *OSPF_packet_type*

Description: An OSPF packet has been received. The OSPF length field indicates a longer packet than indicated by the IP header length field. The packet is discarded.

SPF.002

Level: UE–ERROR

Short Syntax: SPF.002 Bad pkt checksum, from *IP_source*, type *OSPF_packet_type*

Long Syntax: SPF.002 Bad packet checksum, from *IP_source*, type *OSPF_packet_type*

Description: An OSPF packet has been received. The packet has an invalid OSPF checksum. The packet is discarded.

SPF.003

Level: UE–ERROR

Short Syntax: SPF.003 Bad OSPF version, from *IP_source*, type *OSPF_packet_type*

Long Syntax: SPF.003 Bad OSPF version, from *IP_source*, type *OSPF_packet_type*

Description: An OSPF packet has been received. The version field in the OSPF header is not equal to 1. The packet is discarded.

SPF.004

Level: UE–ERROR

***Short Syntax*:** SPF.004 Duplicate Router ID, from *IP_source*, type *OSPF_packet_type*

***Long Syntax*:** SPF.004 Duplicate Router ID, from *IP_source*, type *OSPF_packet_type*

Description: An OSPF packet has been received. The router ID specified in the OSPF header is equal to the router's own ID. Either two interfaces are attached to the same network (OK) or there is a conflict in the assignment of OSPF router IDs (serious). The packet is discarded.

SPF.005

Level: UE–ERROR

***Short Syntax*:** SPF.005 No matching ifc for pkt from *IP_source*, type *OSPF_packet_type*

***Long Syntax*:** SPF.005 No matching SPF–interface for packet from *IP_source*, type *OSPF_packet_type*

Description: An OSPF packet has been received. Either the IP destination specified in the packet is not acceptable, or the parameters in the OSPF header (like area ID) do not match the parameters configured for the receiving interface. The packet is discarded.

SPF.006

Level: UE–ERROR

***Short Syntax*:** SPF.006 Authentication failure, from *IP_source*, type *OSPF_packet_type*

***Long Syntax*:** SPF.006 Packet authentication failure, from *IP_source*, type *OSPF_packet_type*

Description: An OSPF packet has been received which fails to authenticate. The packet is discarded.

SPF.007

Level: UE–ERROR

***Short Syntax*:** SPF.007 No matching nbr for pkt from *IP_source*, type *OSPF_packet_type*

***Long Syntax*:** SPF.007 No matching OSPF neighbor for packet from *IP_source*, type *OSPF_packet_type*

Description: An OSPF packet has been received. The packet is not a hello packet, and does not match any existing OSPF neighbor. The packet is discarded.

SPF.008

Level: UE–ERROR

Short Syntax: SPF.008 Bad pkt type from *IP_source*, type *OSPF_packet_type*

Long Syntax: SPF.008 Bad packet type received from *IP_source*, type *OSPF_packet_type*

Description: An OSPF packet has been received. The OSPF packet type field is invalid. The packet is discarded.

SPF.009

Level: UI–ERROR

Short Syntax: SPF.009 No buffer for mcast to *IP_destination*

Long Syntax: SPF.009 No buffer for multicast packet to *IP_destination*

Description: An attempt was made to send a multicast packet on a nonbroadcast network by expanding the packet on the link level. This expansion failed due to insufficient buffer resources.

SPF.010

Level: P–TRACE

Short Syntax: SPF.010 Received packet type *OSPF_packet_type* from *IP_source*

Long Syntax: SPF.010 Received packet type *OSPF_packet_type* from *IP_source*

Description: An OSPF packet of the specified type was received.

SPF.011

Level: U–TRACE

Short Syntax: SPF.011 Sending unicast type *OSPF_packet_type* dst *IP_destination*

Long Syntax: SPF.011 Sending unicast type *OSPF_packet_type* dst *IP_destination*

Description: Unicast OSPF packet of specified type has been sent to the specified IP destination.

SPF.012

Level: P–TRACE

Short Syntax: SPF.012 Sending mcast type *OSPF_packet_type*, dst *IP_destination* net *network*

Long Syntax: SPF.012 Sending multicast, type *OSPF_packet_type*, destination *IP_destination* net *network*

Description: Multicast OSPF packet of specified type sent out specified interface.

SPF.013

Level: U–INFO

Short Syntax: SPF.013 Rxmitting type *OSPF_packet_type, IP_source –> IP_destination*

Long Syntax: SPF.013 Retransmitting packet, type *OSPF_packet_type, IP_source –> IP_destination*

Description: Unicast OSPF packet of specified type is being retransmitted.

SPF.014

Level: UI–ERROR

Short Syntax: SPF.014 No FSM match, ifc *interface_IP_address,* state *interface_state,* event *interface_event*

Long Syntax: SPF.014 No FSM match, interface *interface_IP_address,* state *interface_state,* event *interface_event*

Description: The specified event occurred while an interface was in the specified state. This occurrence was not covered by the interface Finite State Machine. The event is ignored.

SPF.015

Level: U–INFO

Short Syntax: SPF.015 State change, ifc *interface_IP_address,* new state new_*interface_state,* event *interface_event*

Long Syntax: SPF.015 State change, interface *interface_IP_address,* new state new_*interface_state,* event *interface_event*

Description: The specified event occurred on the specified interface, causing its state to transition.

SPF.016

Level: UE–ERROR

Short Syntax: SPF.016 No match for hlo (virtual link) from *IP_source*

Long Syntax: SPF.016 No match for hello received on virtual link, from *IP_source*

Description: A hello packet was received that could only match a virtual link, yet that link is not configured. The packet is discarded.

SPF.017

Level: UE–ERROR

Short Syntax: SPF.017 Network mask mismatch with *IP_source*

Long Syntax: SPF.017 Network mask mismatch in hello from *IP_source*

Description: Hello packet received from neighbor. Neighbor disagrees with this router concerning the network mask of their common network. The packet is discarded.

SPF.018

Level: UE–ERROR

Short Syntax: SPF.018 Hello interval mismatch with *IP_source*

Long Syntax: SPF.018 Hello interval mismatch in hello from *IP_source*

Description: Hello packet received from neighbor. Neighbor disagrees with this router concerning the hello interval to be used on the common network. The packet is discarded.

SPF.019

Level: UE–ERROR

Short Syntax: SPF.019 Dead interval mismatch with *IP_source*

Long Syntax: SPF.019 Dead interval mismatch in hello from *IP_source*

Description: Hello packet received from neighbor. Neighbor disagrees with this router concerning the ‘‘dead router interval’’ to be used on the common network. The packet is discarded.

SPF.020

Level: UI–ERROR

Short Syntax: SPF.020 No FSM match, nbr *neighbor_IP_address*, state *neighbor_state*, event *neighbor_event*

Long Syntax: SPF.020 No FSM match, neighbor *neighbor_IP_address*, state *neighbor_state*, event *neighbor_event*

Description: The specified event has been generated for the specified neighbor, which is currently in the specified state. This was not anticipated by the neighbor Finite State Machine. The event is ignored.

SPF.021

Level: U–INFO

Short Syntax: SPF.021 State change, nbr *neighbor_IP_address*, new state *neighbor_state*, event *neighbor_event*

Long Syntax: SPF.021 State change, neighbor *neighbor_IP_address*, new state *neighbor_state*, event *neighbor_event*

Description: The specified event has been generated, causing the specified neighbor to transfer to a new state.

SPF.022

Level: UI–ERROR

Short Syntax: SPF.022 Outstanding DD pkt not avail for nbr *neighbor_IP_address*

Long Syntax: SPF.022 Outstanding Database Description packet not avail for neighbor *neighbor_IP_address*

Description: An attempt was made to retransmit a Database Description packet to the specified neighbor, but the packet could not be found. Retransmission is aborted.

SPF.023

Level: UI–ERROR

Short Syntax: SPF.023 Unable to get pkt, to *IP_destination*, ifc *interface_IP_address*

Long Syntax: SPF.023 Unable to get packet to send to *IP_destination*, out interface *interface_IP_address*

Description: An attempt was made to send an OSPF packet to the specified destination. The specified interface has been aborted due to lack of buffers.

SPF.024

Level: UE–ERROR

Short Syntax: SPF.024 Bad length LS adv from *neighbor_IP_address*

Long Syntax: SPF.024 Bad length Link state advertisement received from *neighbor_IP_address*

Description: A link state advertisement has been received from the specified neighbor, and the advertisement's length field indicates that the entire advertisement is NOT fully contained in the received Link State Update Packet. The partial advertisement is discarded.

SPF.025

Level: UE–ERROR

Short Syntax: SPF.025 from *neighbor_IP_address*, adv. *cksum* fl: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.025 from *neighbor_IP_address*, LS advertisement checksum fails: LS type *LS_type* id *advertisement_ID*

Description: A link state advertisement has been received. The advertisement is identified by its LS type and two–part originating ID (see OSPF specification section 12.1). The checksum field contained in the advertisement is invalid. The advertisement is ignored.

SPF.026

Level: UE–ERROR

Short Syntax: SPF.026 from *neighbor_IP_address*, bad type, adv: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.026 from *neighbor_IP_address*, bad LS type, advertisement: typ *LS_type* id *advertisement_ID*

Description: A link state advertisement has been received. The advertisement's LS type field is invalid. The advertisement is ignored.

SPF.027

Level: UE–ERROR

Short Syntax: SPF.027 from *neighbor_IP_address*, ext adv on VL: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.027 from *neighbor_IP_address*, AS external link adv. on Virtual Link: typ *LS_type* id *advertisement_ID*

Description: A link state advertisement has been received. It was received over a virtual link, yet its LS type is equal to AS external link. The advertisement is ignored.

SPF.028

Level: U–INFO

Short Syntax: SPF.028 from *neighbor_IP_address*, old adv: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.028 from *neighbor_IP_address*, old LS advertisement: typ *LS_type* id *advertisement_ID*

Description: A link state advertisement has been received. The advertisement is older than the current database copy. The received advertisement is ignored.

SPF.029

Level: U–INFO

Short Syntax: SPF.029 from *neighbor_IP_address*, self update: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.029 from *neighbor_IP_address*, self update: typ *LS_type* id *advertisement_ID*

Description: A link state advertisement has been received. The advertisement was originated by the router itself, yet is newer than the database copy. This indicates that it originated before the router was last started. This causes the router to advance the LS sequence number and originate a new instantiation of the advertisement.

SPF.030

Level: U–INFO

Short Syntax: SPF.030 from *neighbor_IP_address*, new adv: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.030 from *neighbor_IP_address*, new LS advertisement: typ *LS_type* id *advertisement_ID*

Description: A link state advertisement has been received. The advertisement is newer than the current database copy. This advertisement is flooded out all other interfaces, and installed in the routing database.

SPF.031

Level: U–INFO

Short Syntax: SPF.031 from *neighbor_IP_address*, Old ack for adv: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.031 from *neighbor_IP_address*, Old acknowledgement for advertisement: typ *LS_type* id *advertisement_ID*

Description: An unexpected link state acknowledgement has been received. The acknowledgement, however, is for a previous instantiation of the link state advertisement.

SPF.032

Level: U–INFO

Short Syntax: SPF.032 Bad ack from *neighbor_IP_address* for adv: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.032 Bad acknowledgment from *neighbor_IP_address* for advertisement: typ *LS_type* id *advertisement_ID*

Description: An unexpected link state acknowledgement has been received. The acknowledgement however is for the current instantiation of the link state advertisement.

SPF.033

Level: U–INFO

Short Syntax: SPF.033 LS update retransmission to *neighbor_IP_address*

Long Syntax: SPF.033 LS update retransmission to neighbor *neighbor_IP_address*

Description: A Link State Update packet containing retransmitted link state advertisements has been unicast to the specified neighbor. This probably indicates packet loss during the flooding procedure.

SPF.034

Level: U–INFO

Short Syntax: SPF.034 LS ack sent direct to *neighbor_IP_address*

Long Syntax: SPF.034 LS acknowledgement sent directly to neighbor *neighbor_IP_address*

Description: A Link State Acknowledgement packet has been sent directly to the specified neighbor. This is in response to duplicate link state advertisements received from the neighbor. This probably indicates packet loss during the flooding procedure.

SPF.035

Level: U–INFO

Short Syntax: SPF.035 Flushing advertisement: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.035 Flushing advertisement: typ *LS_type* id *advertisement_ID*

Description: A link state advertisement contained in the link state database has not been refreshed for 2 hours. The advertisement is deleted from the database. This probably indicates that the originator of the advertisement is unreachable. See section 14 of the OSPF specification.

SPF.036

Level: U–INFO

Short Syntax: SPF.036 Originating adv: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.036 Originating LS advertisement: typ *LS_type* id *advertisement_ID*

Description: A link state advertisement is being (re)originated by the router. This can be due to topological change, or the necessity to refresh.

SPF.037

Level: U–INFO

Short Syntax: SPF.037 new route to destination, type *route_type* cost *route_cost*

Long Syntax: SPF.037 New route to destination destination, type *route_type* cost *route_cost*

Description: The SPF routing table build process has detected a new best route to specified destination, having the specified cost.

SPF.038

Level: P–TRACE

Short Syntax: SPF.038 NBMA hello sent to dest *neighbor_IP_address*

Long Syntax: SPF.038 NBMA hello sent to IP destination *neighbor_IP_address*

Description: An OSPF hello has been sent to the specified IP destination. This has been done over a non–broadcast, multi–access interface.

SPF.039

Level: U–INFO

Short Syntax: SPF.039 The OSPF routing protocol is en/disabled

Long Syntax: SPF.039 The OSPF routing protocol is en/disabled

Description: Printed on router startup. Indicates operational status of the SPF protocol.

SPF.040

Level: U–INFO

Short Syntax: SPF.040 SPF Interface *interface_IP_address* is not an IP address, Interface not installed

Long Syntax: SPF.040 SPF Interface *interface_IP_address* is not an IP address, Interface not installed

Description: Printed on router startup when an OSPF interface address is configured, yet this address has not also been configured in the IP console. OSPF interface is not installed.

SPF.041

Level: U–INFO

Short Syntax: SPF.041 Non–Broadcast net *interface_IP_address* is not an SPF interface

Long Syntax: SPF.041 Non–Broadcast net *interface_IP_address* is not an SPF interface

Description: Printed on router startup when OSPF non–broadcast parameters have been configured for a non–existent OSPF interface. These configuration parameters are ignored.

SPF.042

Level: P–TRACE

Short Syntax: SPF.042 protocol (disabled): *IP_source –> IP_destination*, type *OSPF_packet_type* discarded

Long Syntax: SPF.042 protocol (disabled): *IP_source –> IP_destination*, type *OSPF_packet_type* discarded

Description: The OSPF protocol is disabled, yet an OSPF protocol packet has been received. The packet is discarded.

SPF.043

Level: U–INFO

***Short Syntax*:** SPF.043 Duplicate LS ack received from *neighbor_IP_address*

***Long Syntax*:** SPF.043 Duplicate LS acknowledgment received from neighbor *neighbor_IP_address*

Description: Unexpected link state acknowledgements have been received from the specified neighbor. This probably indicates packet loss during the flooding procedure.

SPF.044

Level: UE–ERROR

***Short Syntax*:** SPF.044 from *neighbor_IP_address*, bad age field, adv (*LS_type*,*advertisement_ID*)

***Long Syntax*:** SPF.044 from *neighbor_IP_address*, bad age field, advertisement: typ *LS_type* id *advertisement_ID*

Description: A link state advertisement has been received. The advertisement's LS age field is invalid. The advertisement is ignored.

SPF.045

Level: U–INFO

***Short Syntax*:** SPF.045 non–existent area *proposed_transit_area*, VL discarded

***Long Syntax*:** SPF.045 Transit area *proposed_transit_area* not configured, virtual link discarded

Description: A virtual link has been configured to have a certain transit area, yet that area has not been defined. The virtual link is ignored.

SPF.046

Level: U–INFO

***Short Syntax*:** SPF.046 No backbone configured, VLs discarded

***Long Syntax*:** SPF.046 Backbone area is not configured, all virtual links discarded

Description: Virtual links cannot be used unless a backbone area is configured.

SPF.047

Level: U–INFO

***Short Syntax*:** SPF.047 destination now unreachable

***Long Syntax*:** SPF.047 Destination destination now unreachable

Description: The destination has been found to be unreachable during the routing table build process.

SPF.048

Level: UE–ERROR

Short Syntax: SPF.048 AS ext adv limit exceeded; adv ignored

Long Syntax: SPF.048 Limit of AS external advertisements exceeded; advertisement discarded

Description: The estimated number of advertisements has been exceeded. New AS external advertisements are ignored in order to put a limit on router heap usage.

SPF.049

Level: UE–ERROR

Short Syntax: SPF.049 AS ext adv limit exceeded; origination deferred

Long Syntax: SPF.049 Limit of AS external advertisements exceeded; origination deferred

Description: The estimated number of advertisements has been exceeded. The origination of new AS external advertisements is deferred in order to put a limit on router heap usage.

SPF.050

Level: U–INFO

Short Syntax: SPF.050 from *neighbor_IP_address*, MaxAge: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.050 from *neighbor_IP_address*, received unexpected Max– Age: typ *LS_type* id *advertisement_ID*

Description: A link state advertisement has been received. Its age is MaxAge, and there is no current instantiation of the advertisement in the router's database. The advertisement is acknowledged and then discarded without flooding.

SPF.051

Level: UE–ERROR

Short Syntax: SPF.051 bad adv/ovflo: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.051 error in advertisement or routing overflow: typ *LS_type* id *advertisement_ID*

Description: A link state advertisement has been received. The advertisement contains an error, or cannot be added to the database due to routing table overflow. In any case, the advertisement is discarded.

SPF.052

Level: UE–ERROR

Short Syntax: SPF.052 Stub area mismatch with *IP_source*

Long Syntax: SPF.052 Stub area mismatch in hello from *IP_source*

Description: Hello packet received from neighbor. Neighbor disagrees with this router concerning the attached area's ability to process AS external link advertisements. Hello packet is ignored.

SPF.053

Level: UE–ERROR

Short Syntax: SPF.053 from *neighbor_IP_address*, recvd in stub area, adv (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.053 from *neighbor_IP_address*, type 5 LSA in stub area, adv: typ *LS_type* id *advertisement_ID*

Description: A type 5 link state advertisement has been received. The advertisement is being flooded through a stub area, and is therefore ignored.

SPF.054

Level: C–INFO

Short Syntax: SPF.054 Dijkstra calculation performed: *Number_areas* area(s)

Long Syntax: SPF.054 Dijkstra calculation performed, on *Number_areas* area(s)

Description: As a result of a topology change, the routing table has been recalculated, starting with the Dijkstra calculation.

SPF.055

Level: U–INFO

Short Syntax: SPF.055 Network LSA w/ old Adv Rtr: (*LS_type*,*advertisement_ID*)

Long Syntax: SPF.055 Network LSA with old Advertising Router: (*LS_type*,*advertisement_ID*)

Description: A network links advertisement having one of our addresses as Link State ID, but whose Advertising Router is not our Router ID, has been received. These advertisements are flushed, as they are assumed to be out–of–date.

SPF.056

Level: U–INFO

Short Syntax: SPF.056 Reparsing Network LSA: *Link_State_ID*

Long Syntax: SPF.056 Reparsing Network LSA: *Link_State_ID*

Description: A network link is being reparsed, owing to the fact that there are multiple network–LSAs in the network with the same Link State ID. This indicates that a router has changed OSPF Router IDs, and has originated the same router–LSA before and after the change. This is a normal, but rare, event.

SPF.057

Level: UI–ERROR

Short Syntax: SPF.057 Send unicast type *OSPF_packet_type* dst *IP_destination* fld, rsn *reason_code*, net *network*

Long Syntax: SPF.057 Sending unicast type *OSPF_packet_type* dst *IP_destination* failed, reason *reason_code*, network *network*

Description: Sending of a unicast OSPF packet of specified type failed to the specified IP destination. The *reason_code* is the internal error code for the failure.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network_name.*

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

SPF.058

Level:	UI–ERROR
Short Syntax:	SPF.058 Send multicast type *OSPF_packet_type* dst *IP_destination* fld, rsn *reason_code*, net *network*
Long Syntax:	SPF.058 Sending multicast type *OSPF_packet_type* dst *IP_destination* failed, reason *reason_code*, network *network*
Description:	Sending of a multicast OSPF packet of specified type failed to the specified IP destination. The *reason_code* is the internal error code for the failure.
Cause:	Miscellaneous handler error. (Reason code 1.)
Action:	Check for error messages from handler for *network_name.*
Cause:	Output queue overflow, or other flow control. (Reason code 2.)
Action:	Alleviate congestion.
Cause:	Network down. (Reason code 3.)
Action:	See why handler thinks network is down.
Cause:	Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)
Action:	Check configuration.
Cause:	Host down. (Reason code 5.)
Action:	See why handler thinks host is down.

SPF.059

Level:	UI–ERROR
Short Syntax:	SPF.059 Rxmit type *OSPF_packet_type* fld, *IP_source –> IP_destination*, rsn *reason_code*, net *network*
Long Syntax:	SPF.059 Retransmitting packet failed, type *OSPF_packet_type*, *IP_source –> IP_destination*, reason *reason_code*, network *network*
Description:	Retransmission of unicast OSPF packet of specified type failed. The *reason_code* is the internal error code for the failure.
Cause:	Miscellaneous handler error. (Reason code 1.)
Action:	Check for error messages from handler for *network_name.*
Cause:	Output queue overflow, or other flow control. (Reason code 2.)
Action:	Alleviate congestion.
Cause:	Network down. (Reason code 3.)
Action:	See why handler thinks network is down.
Cause:	Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)
Action:	Check configuration.
Cause:	Host down. (Reason code 5.)
Action:	See why handler thinks host is down.

SPF.060

Level: UI–ERROR

Short Syntax: SPF.060 NBMA hello disc to dest *neighbor_IP_address*, rsn *reason_code*, net *network*

Long Syntax: SPF.060 NBMA hello disc to IP destination *neighbor_IP_address*, reason *reason_code*, network *network*

Description: An OSPF hello has was discarded when attempting to send to the specified IP destination. This was attempted over a non–broadcast, multi–access interface. The *reason_code* is the internal error code for the failure.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network_name.*

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

SPF.061

Level: UI–ERROR

Short Syntax: SPF.061 Disabling OSPF because Integrated ISIS is enabled

Long Syntax: SPF.061 Disabling OSPF because Integrated ISIS is enabled

Description: OSPF cannot be enabled if Integrated ISIS is enabled because these protocols do not currently coordinate access to the IP routing table.

Cause: Both OSPF and Integrated ISIS are enabled in the SRAM configuration.

Action: Disable either OSPF or Integrated ISIS.

SRLY

SDLC Relay

This chapter describes SDLC Relay messages. (This ELS subsystem was entitled SDLC in releases before R15.0. The subsystem and event numbers remain the same, only the name is different.)For information about message content and how to use the message, refer to the Introduction.

SRLY.001

Level: UI–ERROR

Short Syntax: SRLY.001 invld cnfgrton ip addr cnfgd on nt *networkID*

Long Syntax: SRLY.001 Invalid router configuration because an IP address has been configured on network *networkID*

Description: IP addresses are not allowed to be configured on the SDLC relay interfaces.

SRLY.002

Level: UI–ERROR

Short Syntax: SRLY.002 unsptd intf nt *networkID*

Long Syntax: SRLY.002 unsupported interface on network *networkID*

Description: An unsupported network interface had been configured on the SDLC relay group.

SRLY.003

Level: C–INFO

Short Syntax: SRLY.003 SDLC relay intf init strt nt *networkID*

Long Syntax: SRLY.003 SDLC relay initialization started on network *networkID*

Description: The SDLC relay forwarder began initialization on the relay interface.

SRLY.004

Level: C–INFO

Short Syntax: SRLY.004 SDLC relay intf init cmpl nt *networkID*

Long Syntax: SRLY.004 SDLC relay initialization completed on network *networkID*

Description: The SDLC Relay forwarder completed initialization on the relay interface.

SRLY.005

Level: UI–ERROR

Short Syntax: SRLY.005 disc scndry–>prmry pkt addr *SRLY_addr*H net congestd on nt *networkID*

Long Syntax: SRLY.005 Discard SDLC frame with sdlc address *SRLY_addr*H heading to primary station due to network congestion on network *networkID*

Description: A SDLC frame had been discarded out a network interface due to congestion.

Cause: Bursty traffic may be causing outbound frame congestion or internal software inconsistencies exists.

SRLY.006

Level: C–TRACE

Short Syntax: SRLY.006 added prmry–>scndry pkt addr *SRLY_address*H on nt *networkID* to sdlc qu

Long Syntax: SRLY.006 Added packet received on primary side with SDLC address *SRLY_address*H on network *networkID* onto the sdlc queue.

Description: This message is generated whenever the forwarder receives a SDLC relay frame from a primary port (port directly or indirectly attached to a primary station) destined for a secondary port (port directly or indirectly attached to a secondary station).

SRLY.007

Level: C–TRACE

Short Syntax: SRLY.007 added scndry–>prmry pkt addr *SRLY_address*H on nt *networkID* to sdlc qu

Long Syntax: SRLY.007 Added packet received on secondary side with SDLC address *SRLY_address*H on network *networkID* onto the sdlc queue.

Description: This message is generated whenever the forwarder receives a SDLC relay frame from a secondary port (port directly or indirectly attached to a secondary station) destined for a primary port (port directly or indirectly attached to a primary station).

SRLY.008

Level: CE–ERROR

Short Syntax: SRLY.008 frm disc grp *group_num* not dfned nt *networkID*

Long Syntax: SRLY.008 A SDLC relay frame discarded due to group *group_num* defined in the frame received from the network *networkID* not being defined in the router.

Description: A SDLC relay frame has been discarded due to the group number of the frame not being defined for that router.

SRLY.009

Level: CE–ERROR

Short Syntax: SRLY.009 frm disc grp *group_num* dsbld nt *networkID*

Long Syntax: SRLY.009 A SDLC relay frame discarded due to group *group_num* being disabled for frame coming in from the network *networkID*.

Description: A SDLC relay frame has been discarded due to the group not being enabled.

SRLY.010

Level: CE–ERROR

Short Syntax: SRLY.010 frm with sdlc addr *SRLY_addr*H grp *group_num* disc src prmry port dsbld

Long Syntax: SRLY.010 A SDLC relay frame with sdlc address *SRLY_addr*H discarded due to the source primary port of group *group_num* being disabled.

Description: A SDLC relay frame has been discarded due to the source port(where the frame was coming from) being disabled.

SRLY.011

Level: CI–ERROR

Short Syntax: SRLY.011 disc rcved frm from prmry but prt dclrd as sndry for grp *group_num*

Long Syntax: SRLY.011 A SDLC relay frame discarded due to the port being misconfigured in the group *group_num*.

Description: A SDLC relay frame has been discarded due to the SDLC relay ports being inconsistently configured. The router on one side has the port configured as a primary, while the router on the other side has the same port configured as a secondary.

SRLY.012

Level: CI–ERROR

Short Syntax: SRLY.012 frm disc src prt sdlc addr *SRLY_addr*H not found in grp *group_num*

Long Syntax: SRLY.012 A SDLC relay frame discarded due to the src port with sdlc address *SRLY_addr*H specified in the frame not being found in group *group_num*.

Description: A SDLC relay frame has been discarded due to the src port with the sdlc address specified in the frame not being found in the group specified. This is a result of user misconfiguration of the group.

SRLY.013

Level: CI–ERROR

Short Syntax: SRLY.013 frm with sdlc addr *SRLY_addr*H grp *group_num* disc dst prmry port dsbld

Long Syntax: SRLY.013 A SDLC relay frame with sdlc address *SRLY_addr*H discarded due to the destination primary port of group *group_num* being disabled.

Description: A SDLC relay frame has been discarded due to the destination port (where the frame was heading to) being disabled.

SRLY.014

Level: CI–ERROR

Short Syntax: SRLY.014 frm disc prt dst sdlc addr *SRLY_addr*H not fnd in grp *group_num*

Long Syntax: SRLY.014 A SDLC relay frame discarded due to the destination port sdlc address *SRLY_addr*H specified in the packet not being found in group *group_num*.

Description: A SDLC relay frame has been discarded due to the destination port with the sdlc address specified in the frame not being found in the group specified. This is a result of user misconfiguration of the group. The specific port with sdlc address %d was not added to the group.

SRLY.015

Level: CI–ERROR

Short Syntax: SRLY.015 frm with dst sdlc addr *SRLY_addr*H disc rly dwn or rly dsbld nt *networkID*

Long Syntax: SRLY.015 SDLC frame with dst sdlc addr *SRLY_addr*H discarded due to relay down condition on network *networkID*

Description: A SDLC frame had been discarded due to the SDLC relay failing to forward out a network interface which had been in a down state. This message will be printed if the network is down or if IP is not not currently enabled; if IP is not enabled, no SDLC relay can take place, so the frame is simply discarded.

SRLY.016

Level: CI–ERROR

Short Syntax: SRLY.016 dsc scndry–>prmry frm sdlc addr *SRLY_addr*H rjd rsn = *reason* on nt *networkID*

Long Syntax: SRLY.016 discard net rejected sdlc frame address *SRLY_addr*H heading for primary stationwith reject reason = *reason* on network *networkID*

Description: A SDLC relay frame has rejected by the network interface and discarded.

SRLY.017

Level: P–TRACE

Short Syntax: SRLY.017 disc frm *src_SRLY_addr*H –> *dst_SRLY_addr*H nt *networkID*

Long Syntax: SRLY.017 discarded frame with source addr *src_SRLY_addr*H and destination addr *dst_SRLY_addr*H on network *networkID*

Description: A frame had been discarded due to SDLC relay not configured on interface noted.

Cause: The null or fake forwarder is configured on the interface, all received SDLC relay frames are discarded.

SRLY.018

Level: C–INFO

Short Syntax: SRLY.018 frwrd SRLY frm scndry–>prmry sdlc addr *SRLY_addr*H nt *networkID*

Long Syntax: SRLY.018 forwarded SDLC Relay frame from secondary station destined for primary station with frame sdlc address *SRLY_addr*H on network *networkID*

Description: A frame travelling in the direction of secondary–>primary station has been forwarded out onto the interface noted.

SRLY.019

Level: C–INFO

Short Syntax: SRLY.019 frwrd SRLY frm prmry–>scndry sdlc addr *SRLY_addr*H nt *networkID*

Long Syntax: SRLY.019 forwarded SDLC Relay frame from primary station destined for secondary station with frame sdlc address *SRLY_addr*H on network *networkID*

Description: A frame travelling in the direction of primary–>secondary station has been forwarded out onto the interface noted.

SRLY.020

Level: UI–ERROR

Short Syntax: SRLY.020 dsc frm from nt *networkID* IP not enbled

Long Syntax: SRLY.020 discard sdlc frame from network *networkID* because IP is not enabled on router

Description: A SDLC relay frame has discarded because in order for SDLC relay to work, IP has to be enabled on the router. The user must add at least one IP address to at least one of its non–SDLC relay interfaces.

SRLY.021

Level: CI–ERROR

Short Syntax: SRLY.021 frm not fwrd dst IP addr *ip_address* mscnfgrd grp *group_num*

Long Syntax: SRLY.021 Frame not forwarded because the destination ip addresses *ip_address* for group *group_num* is one of the ip addresses configured on the source router.

Description: This message is generated when the forwarder must discard a packet because the destination ip address configured for the group is one of the ip addresses configured on the source router.

SRLY.022

Level: CI–ERROR

Short Syntax: SRLY.022 disc frm grp *group_num* cnfg bad

Long Syntax: SRLY.022 Frame discarded because group *group_num* configuration is bad.

Description: This message is generated when the forwarder must discard a packet because the group configuration among the routers participating in SDLC relay are inconsistent with each other. Check to make sure the primary and secondary attributes of the ports in groups are consistent.

SRLY.023

Level: C–INFO

Short Syntax: SRLY.023 IP dest *ip_address* unrchble

Long Syntax: SRLY.023 The IP destination *ip_address* is unreachable.

Description: This message is generated when the encapsulated SDLC frame is lost due to the IP destination address specified in the frame being unreachable. The software will try to use the next IP address configured to resend the frame. If there are no more addresses, the software will drop the packet. The user should try to delete the IP address from the IP address list using the command DELETE IP–ADDRESS command.

SRLY.024

Level: CI–ERROR

Short Syntax: SRLY.024 disc prmry–>scndry pkt addr *SRLY_addr*H net congestd on nt *networkID*

Long Syntax: SRLY.024 Discard SDLC frame with sdlc address *SRLY_addr*H heading to secondary station due to network congestion on network *networkID*

Description: A SDLC frame had been discarded out a network interface due to congestion.

Cause: Bursty traffic maybe causing outbound frame congestion or internal software inconsistencies exists.

SRLY.025

Level: CI–ERROR

Short Syntax: SRLY.025 frm with sdlc addr *SRLY_addr*H grp *group_num* disc src scndry port dsbld

Long Syntax: SRLY.025 A SDLC relay frame with sdlc address *SRLY_addr*H discarded due to the source secondary port of group *group_num* being disabled.

Description: A SDLC relay frame has been discarded due to the source port(where the frame was coming from) being disabled.

SRLY.026

Level: CI–ERROR

Short Syntax: SRLY.026 frm with sdlc addr *SRLY_addr*H grp *group_num* disc dst scndry port dsbld

Long Syntax: SRLY.026 A SDLC relay frame with sdlc address *SRLY_addr*H discarded due to the destination secondary port of group *group_num* being disabled.

Description: A SDLC relay frame has been discarded due to the destination port (where the frame was heading to) being disabled.

SRLY.027

Level: CI–ERROR

Short Syntax: SRLY.027 dsc prmry–>scndry frm sdlc addr *SRLY_addr*H rjd rsn = reason on nt *networkID*

Long Syntax: SRLY.027 discard net rejected sdlc frame address *SRLY_addr*H heading for secondary stationwith reject reason = reason on network *networkID*

Description: A SDLC relay frame has rejected by the network interface and discarded.

SRLY.028

Level: CI–ERROR

Short Syntax: SRLY.028 dsc frm grp *group_addr* no *IPaddr* cnfgrd

Long Syntax: SRLY.028 discard frame no ip address configured for group *group_addr*

Description: A SDLC relay frame destined for a far router has been discarded because no IP address has been configured for the remote port. Panic ‘‘SRLYimem’’

Short Syntax: SRLY mem alloc fld

Description: The SRLY forwarder failed to allocate sufficient memory to complete initialization.

Action: Contact customer service. Panic ‘‘sdlcudperr’’

Short Syntax: sdlc udp port not avail

Description: Another application registered previously with srly’s UDP port.

Action: Contact customer service.

SRT

Source Routing Transparent

This chapter describes Source Routing Transparent Bridge messages. For information about message content and how to use the message, refer to the Introduction.

SRT.001

Level: UI–ERROR

Short Syntax: SRT.001 No buf to dup broadcast frame *source_mac–>dest_mac* to port *port*, nt *network*

Long Syntax: SRT.001 No buffer available to duplicate frame from *source_mac* to *dest_mac* on to port *port*, network *network*

Description: No buffer available to copy a frame in order to send a bridged frame on multiple interfaces. Bridged packets are sent on multiple interfaces either for multicast destination addresses, or in the case of certain static entries. No copy of this frame will be sent on the specified port and network.

Cause: Severe packet buffer shortage.

Action: Check memory statistics in GWCON to verify packet buffer level.

Cause: Traffic peak using all available buffers.

Action: This is the problem if this message occurs infrequently.

SRT.002

Level: UI–ERROR

Short Syntax: SRT.002 Err *error_code* setting promsic mode on nt *network*

Long Syntax: SRT.002 Error code *error_code* trying to set promiscuous mode on network *network*

Description: The Spanning Tree Protocol requested setting this network into Learning state, but the command to the device failed. The *error_code* is a device–specific error code that may indicate what the error is.

Cause: Hardware failure or software bug.

Action: Contact customer service.

SRT.003

Level: UI–ERROR

Short Syntax: SRT.003 Err *error_code* add stat ent on nt *network*

Long Syntax: SRT.003 Error code *error_code* trying to add static entries on network *network*

Description: An attempt to add a set of static entries to the internal database of a bridging interface having internal filtering failed. The *error_code* is a device–specific error code that may indicate what the error is.

Cause: Hardware failure or software bug.

Action: Contact customer service.

SRT.004

Level: UI–ERROR

Short Syntax: SRT.004 No buf for *command_name* cmd to net *network*

Long Syntax: SRT.004 No buffer available for *command_name* command to network *network*

Description: No buffer was available to send a command to the device. The possible command names are “D_CNFGSRB” (configure source–routing bridging), “SRT_ON” (promiscuous on), “SRT_INFORM” (learn capabilities of device), “SRT_SET_AGE” (set age for filtering database in device), “SRT_DECR_AGE” (do ageing pass on filtering database in device), “SRT_ADD_ENTRY” (add static entry), “SRT_DEL_ENTRY” (delete entry, from console), “SRT_SEARCH_ENTRY” (search for particular entry, from console), and “SRT_LIST_ENTRY” (list contents of learning database in card). For commands “D_CNFGSRB” and “SRT_ON” the result will be that the interface may remain in the wrong state. A failure on “SRT_INFORM” could cause serious problems. For other commands the results will be less serious.

Cause: Severe packet buffer shortage.

Action: Check memory statistics in GWCON to verify packet buffer level.

Cause: Traffic peak using all available buffers.

Action: This is the problem if this message occurs infrequently.

SRT.005

Level: UI–ERROR

Short Syntax: SRT.005 *source_mac–>dest_mac* send fld, rsn *reason_code*, port *port* nt *network*

Long Syntax: SRT.005 Sending Frame from *source_mac* to *dest_mac* failed, reason *reason_code*, on port *port* network *network*

Description: The sending of a packet being forwarded failed. The reason is the internal error code for the failure.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network_name.*

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

SRT.006

Level: CI–ERROR

Short Syntax: SRT.006 Input q ovf *source_mac–>dest_mac*, dropped, nt *network*

Long Syntax: SRT.006 Input queue overflow on frame from *source_mac* to *dest_mac*, packet dropped from network *network*

Description: The input queue for frames to be forwarded is too long, and this frame has been dropped to attempt to alleviate the congestion.

Cause: Bursty traffic may be causing congestion.

Action: Wait for burst to subside.

Cause: Too much traffic for forwarder to forward.

Action: Reconfigure network. Increase speed of router.

Cause: Inadequate buffer resources.

Action: Examine memory statistics in GWCON.

SRT.007

Level: CI–ERROR

*Short Syntax***:** SRT.007 BPDU q ovf frm *source_mac*, dropped, nt *network*

*Long Syntax***:** SRT.007 Bridge Protocol Data Unit input queue overflow on frame from *source_mac*, dropped from network *network*

Description: The input queue for Spanning Tree Protocol Bridge Protocol Data Units is too long, and this frame has been dropped to attempt to alleviate the congestion.

Cause: Source node streaming BDPU frames.

Action: Correct behavior of source node.

Cause: Too much traffic for forwarder to forward.

Action: Reconfigure network. Increase speed of router.

Cause: Inadequate buffer resources.

Action: Examine memory statistics in GWCON.

SRT.008

Level: CE–ERROR

*Short Syntax***:** SRT.008 *source_mac–>dest_mac* too big (*reformatted_length* > *output_maximum*) for port *port* nt *network*, dropped

*Long Syntax***:** SRT.008 Frame from *source_mac* to *dest_mac* is too big (reformatted length *reformatted_length* bytes > output maximum size *output_maximum* bytes) for port *port* network *network*, dropped

Description: The specified frame is too large to send on this outgoing port and network. The reformatted_length is the size of the frame including MAC headers after any mapping of data link headers.

Cause: Host on network with large maximum frame size sending to host on network with smaller maximum frame size.

Action: Reconfigure sending host to not send such large frames. If frame is of a routable protocol supporting fragmentation (such as IP or ISO) or maximum frame size determination (DNA or XNS), convert to using routing instead of bridging.

Cause: Host on network with large maximum frame size sending to host via an intervening network with smaller maximum frame size.

Action: Reconfigure network to use networks with large maximum frame size (such as FDDI or 802.5) as the backbone networks. Reconfigure port costs in Spanning Tree Protocol to favor spanning trees via networks with large maximum frame sizes.

SRT.009

Level: UE–ERROR

Short Syntax: SRT.009 *source_mac–>dest_mac* drp, nt *network* down

Long Syntax: SRT.009 Frame from *source_mac* to *dest_mac* dropped, input network *network* is down

Description: A frame has been received for bridging on a network that is down. It will be ignored.

Cause: A BDPU has been sent to the unicast address of the router on this interface.

Action: Correct action of sending node.

Cause: Internal state inconsistency.

SRT.010

Level: P–TRACE

Short Syntax: SRT.010 *source_mac–>dest_mac* drp, src add flt, port *port* nt *network*

Long Syntax: SRT.010 Frame from *source_mac* to *dest_mac* dropped, source address filtered, port *port* network *network*

Description: A MAC frame has been received by the hardware, but is being dropped because the source MAC address is being administratively filtered by the bridge. The frame will be dropped.

Cause: Receipt of frame whose source MAC address matches the source filter.

SRT.011

Level: U–TRACE

Short Syntax: SRT.011 *source_mac–>dest_mac* dropped, input port *port* nt *network* not forwarding

Long Syntax: SRT.011 Frame from *source_mac* to *dest_mac* dropped, input port *port* network *network* not in forwarding state

Description: A MAC frame was received on a port that is still only in “learning” state. Frames are only bridged when the input port is in “forwarding” state. While the port is still in “learning” state, they are only processed to learn the source addresses for the filtering database. The frame will not be bridged.

Cause: Normal part of transition to “forwarding” state.

SRT.012

Level: U–INFO

Short Syntax: SRT.012 *source_mac–>dest_mac* dropped, output port *port* nt *network* not forwarding

Long Syntax: SRT.012 Frame from *source_mac* to *dest_mac* dropped, output port *port* network *network* not in forwarding state

Description: A MAC frame was being bridged, but the destination port was not in "forwarding" state. It will not be sent on that port.

Cause: Output port still in "learning" state.

Action: None needed, port will transition to "forwarding".

Cause: Static entry in filtering database points to port that is not in "forwarding" state.

SRT.013

Level: P–TRACE

Short Syntax: SRT.013 *source_mac–>dest_mac* drp, dst same LAN, port *port* nt *network*

Long Syntax: SRT.013 Frame from *source_mac* to *dest_mac* dropped, destination on same LAN, port *port* network *network*

Description: A MAC frame has been received whose destination address is known to be on the same side of the bridge as the packet came from. It is dropped by the filtering logic since it does not need to be bridged.

Cause: Normal local traffic on network.

SRT.014

Level: CI–ERROR

Short Syntax: SRT.014 *source_mac–>dest_mac* drp, dst port *port* not enabled, nt *network*

Long Syntax: SRT.014 Frame from *source_mac* to *dest_mac* dropped, destination port *port* not enabled, network *network*

Description: A frame being bridged was destined for a port which is not running transparent bridging, or not in "forwarding" state for transparent bridging.

Cause: Static entry in filtering database points to port that is not in "forwarding" state.

SRT.015

Level: P–TRACE

Short Syntax: SRT.015 *source_mac–>dest_mac* brdg port *port* nt *network* to port *port* nt *network*

Long Syntax: SRT.015 Frame from *source_mac* to *dest_mac* bridged from port number port network *network* to port number port network *network*

Description: A frame is being bridged between these two interfaces. The destination address was known, so it was sent only to the correct destination network.

SRT.016

Level: P–TRACE

Short Syntax: SRT.016 *source_mac–>dest_mac* brdg–all port *port* nt *network* to port *port* nt *network*

Long Syntax: SRT.016 Frame from *source_mac* to *dest_mac* bridged to all ports from port number port network *network* to port number port network *network*

Description: A frame is being transparently bridged to all active transparent bridging ports. This happens when the frame destination is a multicast, when the frame destination is not in the learning database, or when required by static entries in the learning database. There will be one message for each port the frame is sent on.

SRT.017

Level: U–INFO

Short Syntax: SRT.017 Enabling SRT on port *port* nt *network*

Long Syntax: SRT.017 Enabling SRT on port *port* network *network*

Description: The SRT forwarder is starting the process of enabling bridging on the specified interface. This starts when the interface comes up from a self-test.

SRT.018

Level: C–INFO

Short Syntax: SRT.018 SRT startup complete on port *port* nt *network*

Long Syntax: SRT.018 SRT startup complete on port *port* network *network*

Description: The SRT forwarder has completed the process of enabling bridging on the specified interface. It will now enter “blocking” state.

SRT.019

Level: UI–ERROR

Short Syntax: SRT.019 Unsupp ifc typ *type_name*, nt *network*

Long Syntax: SRT.019 Unsupported interface type *type_name*, network *network*

Description: The SRT forwarder had been enabled on a type of interface it does not support.

Cause: Enabling SRT on an interface which does not support SRT, such as ProNET–10.

SRT.020

Level: UI–ERROR

Short Syntax: SRT.020 Can't autocfg brdg addr, lowest port *port* nt *network* no MAC addr

Long Syntax: SRT.020 Cannot autoconfigure the bridge address, the lowest numbered port *port* network *network* has no MAC address

Description: The user has configured the bridge to autoconfigure the bridge address based on the MAC address of the lowest number port. However, the lowest numbered port is of a type that does not have a MAC address, such as a serial line.

Action: Assign address to bridge by using SRT config> command "SET BRIDGE".

SRT.021

Level: U–TRACE

Short Syntax: SRT.021 Bridge *source_mac–>dest_mac*, no fwd, nt *network*

Long Syntax: SRT.021 Bridge frame from *source_mac* to *dest_mac*, no forwarder, network *network*

Description: Bridge frame received, but there is no bridging available in this load. The frame will be ignored.

Cause: Receiving a frame to 802.2 destination SAP 42.

SRT.022

Level: UI–ERROR

Short Syntax: SRT.022 Bridge config with no ports, disabling

Long Syntax: SRT.022 Bridge configured with no ports on it, disabling the bridge

Description: The bridge has been enabled, but there are no ports configured on that bridge. The bridge will be left disabled. It takes at least two ports to be a bridge.

Action: Add ports in SRT config> console.

SRT.023

Level: UI–ERROR

Short Syntax: SRT.023 port *port* config on nonexist network number *network_number*

Long Syntax: SRT.023 port *port* configured on nonexistent *network* number *network_number*

Description: The port has been configured to use a network that has not been configured with the Config> ADD DEVICE command. This port of the bridge will be disabled.

Cause: Inconsistency between router device configuration and bridge configuration.

Action: Correct the network number in the bridge configuration, or add the network in the device configuration.

SRT.024

Level: UI–ERROR

Short Syntax: SRT.024 *existent_port_count* ports is

Long Syntax: SRT.024 *existent_port_count* existent ports is less than 2, disabling bridge

Description: Less than two (valid) ports have been configured on the bridge. There must be at least two ports.

Cause: Less than two ports configured.

Action: Add more ports, or don't try and use bridging.

Cause: Too many ports on non–configured devices.

Action: Resolve configuration conflicts between bridging ports and devices.

SRT.025

Level: UI–ERROR

Short Syntax: SRT.025 No mem for filt db (req *requested_size*, min *minimum_size*), disabl

Long Syntax: SRT.025 No memory for filtering databse (desired size *requested_size* bytes, absolute minimum size *minimum_size* bytes), disabling bridge

Description: There is not enough free memory to allocate even a minimal size filtering database. The bridge will be disabled. The bridge starts by trying to allocate requestd_size bytes, and then tries with progressively smaller sizes down to *minimum_size*. The minimum size is enough only for the registered and static entries.

Cause: Severe shortage of memory.

Action: Reduce routing table sizes in other protocols, use system with less protocols, expand memory in router.

SRT.026

Level: C–INFO

Short Syntax: SRT.026 *source_mac==dest_mac*, drop, port *port* nt *network*

Long Syntax: SRT.026 Frame from *source_mac* to *dest_mac*, source same as destination, dropping, from port *port* network *network*

Description: Frames to and from the same address are not bridged by this bridge.

SRT.027

Level: P–TRACE

Short Syntax: SRT.027 Chg state *old_state* to *new_state*, port *port* nt *network*

Long Syntax: SRT.027 Changing port state from *old_state* to *new_state* for port *port*, network *network*

Description: The Spanning Tree Protocol has requested this state change for this port in the SRT bridge. The *old_state* and *new_state* are one of: FORWARDING (Spanning Tree Protocol Forwarding state), LEARNING (Spanning Tree Protocol Learning state), LISTENING (Spanning Tree Protocol Listening state), BLOCKED (Spanning Tree Protocol Blocking state), CONFIGURING (configuration of port device pending), POSTCONFIGURING (configuration of port device done), PRECONFIGURING (port enabled, configuration of port device to start), and DISABLED (port disabled).

SRT.028

Level: UI_ERROR

Short Syntax: SRT.028 No room for PERM *mac_address* in filt database, disabling

Long Syntax: SRT.028 No room for permanent address *mac_address* in filtering database, disabling bridge

Description: There is no room for the permanent entry in the filtering database. The bridge will be disabled.

Cause: Filtering database size too small.

Action: Make filtering database larger.

Cause: Too many permanent entries.

Action: Configure less permanent entries.

SRT.029

Level: UI_ERROR

Short Syntax: SRT.029 No memm for PERM *mac_address*, disabling

Long Syntax: SRT.029 No memory for permanent address *mac_address*, disabling bridge

Description: There is no room for the permanent entry in an auxiliary database. The bridge will be disabled.

Cause: Too little free memory.

Action: Make routing databases smaller.

Action: Increase memory size.

Cause: Too many permanent entries.

Action: Configure less permanent entries.

SRT.030

Level: UI–ERROR

Short Syntax: SRT.030 *command* Cmd fld to net *network*

Long Syntax: SRT.030 *command* command failed to network *network*

Description: A command to a network device failed. The possible command names are "SRT_ON" (promiscuous on), "SRT_OFF" (promiscuous off), "SRT_INFORM" (learn capabilities of device), "SRT_ADD_ENTRY" (add static entry in device), "SRT_SET_AGE" (set age for filtering database in device), and "SRT_DECR_AGE" (do ageing pass on filtering database in device). For commands "SRT_ON" and "SRT_OFF" the result will be that the interface may remain in the wrong state. A failure on "SRT_INFORM" could cause serious problems. For other commands the results will be less serious.

Cause: Hardware failure or software bug.

Action: Contact customer service.

SRT.031

Level: UI–ERROR

Short Syntax: SRT.031 No buf to dup *routing_type* frame *source_mac–>dest_mac* to port *port*, nt *network*

Long Syntax: SRT.031 No buffer available to duplicate *routing_type* frame from *source_mac* to *dest_mac* on to port *port*, network *network*

Description: No buffer available to copy a frame in order to send an All Routes Explorer (ARE) or Spanning Tree Explorer (STE) *routing_type* frame on multiple interfaces. ARE frames are sent on all interfaces which are part of the SRT spanning tree, STE frames are sent on all interfaces running source-routing. No copy of this frame will be sent on the specified port and network.

Cause: Severe packet buffer shortage.

Action: Check memory statistics in GWCON to verify packet buffer level.

Cause: Traffic peak using all available buffers.

Action: This is the problem if this message occurs infrequently.

SRT.032

Level: UI–ERROR

Short Syntax: SRT.032 SR *source_mac–>dest_mac* send fld, rsn *reason_code*, port *port* nt *network*

Long Syntax: SRT.032 Sending source routed frame from *source_mac* to *dest_mac* failed, reason *reason_code*, on port *port* network *network*

Description: The sending of a source routed frame being forwarded failed. The *reason_code* is the internal error code for the failure.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for *network_name.*

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

SRT.033

Level: C–TRACE

Short Syntax: SRT.033 *routing_type* dup RD drop *source_mac–>dest_mac* from port *port*, nt *network*

Long Syntax: SRT.033 *routing_type* with duplicate Route Descriptor from *source_mac* to *dest_mac* from port *port*, network *network*

Description: A source–routed frame having a All Routes Descriptor (ARE) or Spanning Tree Explorer (STE) *routing_type* in the RIF has a duplicate Routing Descriptor in the RIF. The frame will be dropped. This is a normal occurrence for ARE frames when there are any duplicate paths in the source routing domain. For STE frames, this indicates that there is an interface that is part of the source–routing spanning tree that should not be.

Cause: Receiving an ARE/STE from a segment is has already been on.

Action: None needed for ARE, this is normal. For STE, one may want to correct it's "spanning tree," but this is not essential.

SRT.034

Level: UE–ERROR

Short Syntax: SRT.034 SRF dup LOUT (RIF *RIF*) drop *source_mac–>dest_mac* from port *port*, nt *network*

Long Syntax: SRT.034 SRF with duplicate LOUT (RIF *RIF*) from *source_mac* to *dest_mac* from port *port*, network *network*

Description: A source–routed frame of Specifically–routed frame (SRF) type has a duplicate LOUT (outgoing LAN ID). This is illegal, and the frame will be dropped.

Cause: Station sending frame with invalid RIF that would go through the same bridge more than once, thus looping forever.

Action: Find out why station is using this RIF. Either it is using a hand–configured one that is wrong, or there is a bug in the discovery algorithm.

SRT.035

Level: UE–ERROR

Short Syntax: SRT.035 ARE max RD (RIF *RIF*) drop *source_mac–>dest_mac* from port *port*, nt *network*

Long Syntax: SRT.035 All Routes Explorer exceeds maximum Route Descriptors (RIF *RIF*) from *source_mac* to *dest_mac* from port *port*, network *network*

Description: An All Routes Explorer (ARE) source–routed frame has more Route Descriptors than this bridge is configured to allow for ARE frames. The frame will be dropped.

Cause: Upstream bridge has an ARE RD limit inconsistent with this bridge.

Action: Reconfigure all bridges in source–routing domain to have consistent ARE RD limit.

Cause: Network has too many hops for configured ARE RD limit.

Action: Reconfigure all bridges in source–routing domain to have ARE RD limit consistent with the diameter of the domain.

SRT.036

Level: UE–ERROR

Short Syntax: SRT.036 STE max RD (RIF *RIF*) drop *source_mac–>dest_mac* from port *port*, nt *network*

Long Syntax: SRT.036 Spanning Tree Explorer exceeds maximum Route Descriptors (RIF *RIF*) from *source_mac* to *dest_mac* from port *port*, network *network*

Description: A Spanning Tree Explorer (STE) source–routed frame has more Route Descriptors than this bridge is configured to allow for STE frames. The frame will be dropped.

Cause: Upstream bridge has an STE RD limit inconsistent with this bridge.

Action: Reconfigure all bridges in source–routing domain to have consistent STE RD limit.

Cause: Network has too many hops for configured STE RD limit.

Action: Reconfigure all bridges in source–routing domain to have STE RD limit consistent with the diameter of the domain.

SRT.037

Level: CE–ERROR

Short Syntax: SRT.037 SRF unk LOUT (RIF *RIF*) drop *source_mac–>dest_mac* from port *port*, nt *network*

Long Syntax: SRT.037 SRF with unknown LOUT (RIF *RIF*) from *source_mac* to *dest_mac* from port *port*, network *network*

Description: A source–routed frame of Specifically–routed frame (SRF) type has an outgoing LOUT (LAN ID Out) that does not match that of any active source–routing interface in the router. It will be dropped.

Cause: End station using RIF that was discovered before an interface went down in the router.

Action: None should be needed, the session on the station will fail, and it will re–initiate route discovery.

Cause: More than one bridge on the incoming segment with the same bridge number, and this LOUT matches in it.

Action: Reconfigure for legal configuration. All Bridge Numbers must be unique on a given segment.

Cause: End station using completely invalid RIF.

Action: Find out why station is using this RIF.

SRT.038

Level: P–TRACE

Short Syntax: SRT.038 ARE rcv (RIF *RIF*) *source_mac–>dest_mac* from port *port*, nt *network*

Long Syntax: SRT.038 All Routes Explorer received (RIF *RIF*) from *source_mac* to *dest_mac* from port *port*, network *network*

Description: An All Routes Explorer frame has been received on the specified port.

SRT.039

Level: P–TRACE

Short Syntax: SRT.039 ARE sent (RIF *RIF*) *source_mac–>dest_mac* to port *port*, nt *network*

Long Syntax: SRT.039 All Routes Explorer sent (RIF *RIF*) from *source_mac* to *dest_mac* to port *port*, network *network*

Description: An All Routes Explorer frame has been sent on the specified port.

SRT.040

Level: P–TRACE

Short Syntax: SRT.040 STE rcv (RIF *RIF*) *source_mac–>dest_mac* from port *port*, nt *network*

Long Syntax: SRT.040 Spanning Tree Explorer received (RIF *RIF*) from *source_mac* to *dest_mac* from port *port*, network *network*

Description: A Spanning Tree Explorer frame has been received on the specified port.

SRT.041

Level: P–TRACE

Short Syntax: SRT.041 STE sent (RIF *RIF*) *source_mac–>dest_mac* to port *port*, nt *network*

Long Syntax: SRT.041 Spanning Tree Explorer sent (RIF *RIF*) from *source_mac* to *dest_mac* to port *port*, network *network*

Description: A Spanning Tree Explorer frame has been sent on the specified port.

SRT.042

Level: U–INFO

Short Syntax: SRT.042 *routing_type* LF lowered (*old_LF* to *new_LF*) *source_mac–>dest_mac* from port *port*, nt *network*

Long Syntax: SRT.042 *routing_type* Largest Frame size lowered (from *old_LF* bytes to *new_LF* bytes) from *source_mac* to *dest_mac* from port *port*, network *network*

Description: A source–routing explorer (ARE or STE in *routing_type*) has had the Largest Frame (LF) field lowered in its RIF. This happens whenever a frame is received from a segment with a smaller maximum frame size than the one presently encoded in the LF bits. This is a normal part of the spanning tree protocol to determine the maximum frame size on all routes.

Cause: It is somewhat abnormal to see this happen on received frames, and indicates that the endnodes or other bridges on this segment have different frame sizes configured. However, it is a perfectly legal configuration.

Action: Make frame size configurations consistent on a given segment.

SRT.043

Level: C–INFO

Short Syntax: SRT.043 *routing_type* LF lowered (*old_LF* to *new_LF*) *source_mac–>dest_mac* to port *port*, nt *network*

Long Syntax: SRT.043 *routing_type* Largest Frame size lowered (from *old_LF* bytes to *new_LF* bytes) from *source_mac* to *dest_mac* to port *port*, network *network*

Description: A source–routing explorer (ARE or STE in *routing_type*) has had the Largest Frame (LF) field lowered in its RIF. This happens whenever a frame is sent to a segment with a smaller maximum frame size than the one presently encoded in the LF bits. This is a normal part of the spanning tree protocol to determine the maximum frame size on all routes.

SRT.044

Level: P–TRACE

Short Syntax: SRT.044 SRF rcv (RIF *RIF*) *source_mac–>dest_mac* from port *port*, nt *network*

Long Syntax: SRT.044 Specifically–routed frame received (RIF *RIF*) from *source_mac* to *dest_mac* from port *port*, network *network*

Description: A Specifically–routed frame has been received on the specified port.

SRT.045

Level: P–TRACE

Short Syntax: SRT.045 Send SRF (RIF *RIF*) *source_mac–>dest_mac* to port *port*, nt *network*

Long Syntax: SRT.045 Sending Specifically–routed frame (RIF *RIF*) from *source_mac* to *dest_mac* to port *port*, network *network*

Description: A Specifically–routed frame is being sent on the specified port.

SRT.046

Level: UI–ERROR

Short Syntax: SRT.046 *routing_type* rcv *source_mac–>dest_mac* from disabl port *port*, nt *network*, disc

Long Syntax: SRT.046 *routing_type* frame received from *source_mac* to *dest_mac* on disabled port *port*, network *network*, discarded

Description: A source–routed frame has been received on the specified port, but that port is not configured for bridging. The *routing_type* is one of SRF (Specifically–routed frame), STE (Spanning Tree Explorer), or ARE (All Routes Explorer). This really should not happen on more than a transient basis, because ports that are not enabled for bridging should not be queueing packets to the source–routing forwarder.

SRT.047

Level: UI–ERROR

Short Syntax: SRT.047 *routing_type* rcv *source_mac–>dest_mac* from non–SR port *port*, nt *network*, disc

Long Syntax: SRT.047 *routing_type* frame received from *source_mac* to *dest_mac* on non–source–routing port *port*, network *network*, discarded

Description: A source–routed frame has been received on the specified port, but that port is not configured for source–routing bridging. The *routing_type* is one of SRF (Specifically-routed frame), STE (Spanning Tree Explorer), or ARE (All Routes Explorer). This really should not happen on more than a transient basis, because ports that are not enabled for bridging should not be queueing packets to the source-routing forwarder.

SRT.048

Level: P–TRACE

Short Syntax: SRT.048 STE dropped (RIF *RIF*) *source_mac–>dest_mac* from blk port *port*, nt *network*

Long Syntax: SRT.048 Spanning Tree Explorer dropped (RIF *RIF*) from *source_mac* to *dest_mac* from blocked port *port*, network *network*

Description: A Spanning Tree Explorer (STE) frame was dropped, and not forwarded, because the incoming port is not part of the spanning tree, or has been configured not to forward STE frames.

Cause: Normal for STE frames, this is the difference between them and ARE frames.

SRT.049

Level: P–TRACE

Short Syntax: SRT.049 STE not sent (RIF *RIF*) *source_mac–>dest_mac* to blk port *port*, nt *network*

Long Syntax: SRT.049 Spanning Tree Explorer not sent (RIF *RIF*) from *source_mac* to *dest_mac* to blocked port *port*, network *network*

Description: A Spanning Tree Explorer (STE) frame was not sent on the specified port because it is not part of the spanning tree, or has been configured not to forward STE frames.

Cause: Normal for STE frames, this is the difference between them and ARE frames.

SRT.050

Level: UI–ERROR

Short Syntax: SRT.050 err *error_string* ena SR on nt *network*

Long Syntax: SRT.050 Got *error_string* error trying to enable source-routing on network *network*

Description: The bridge tried to enable source–routing bridging on this interface, but the interface refused the configuration command. Source–routing will be left disabled on this interface.

Cause: Either bad commands were passed to the interface, or there is a bug in the interface firmware.

Action: Contact customer service.

SRT.051

Level: UE–ERROR

Short Syntax: SRT.051 SRF *source_mac–>dest_mac* too big (*reformatted_length* > *output_maximum*) for port *port* nt *network*, dropped

Long Syntax: SRT.051 Specifically–routed frame from *source_mac* to *dest_mac* is too big (reformatted length *reformatted_length* > output maximum size *output_maximum*) for port *port* network *network*, dropped

Description: The specified Specifically–routed (source–routed) frame is too large to send on this outgoing port and network. The *reformatted_length* is the size of the frame including MAC headers after any mapping of data link headers.

Cause: Host not honoring LF bit values from its returned explorer frames.

Action: Fix host.

SRT.052

Level: UE–ERROR

Short Syntax: SRT.052 *routing_type source_mac–>dest_mac* too big (*reformatted_length* > *output_maximum*) for port *port* nt *network*, dropped

Long Syntax: SRT.052 *routing_type* frame from *source_mac* to *dest_mac* is too big (reformatted length *reformatted_length* > output maximum size *output_maximum*) for port *port* network *network*, dropped

Description: The source–routed explorer (ARE or STE *routing_type*) frame is too large to send on this outgoing port and network. The *reformatted_length* is the size of the frame including MAC headers after any mapping of data link headers.

Cause: The sending host is putting too much data in its explorer frames. These should normally be short, since it should not be making any assumptions about the maximum frame size available.

Action: Correct behavior of sending host.

SRT.053

Level: UI–ERROR

Short Syntax: SRT.053 *routing_type* inv RIF len *RIF_length*, *source_mac*– >*dest_mac* port *port*, nt *network*, disc

Long Syntax: SRT.053 *routing_type* with invalid RIF lenth *RIF_length* from *source_mac* to *dest_mac* from port *port*, network *network*, discarded

Description: A source–routing frame was received with an invalid RIF length encoded in the Length bits of the RIF. The *routing_type* is one of SRF (Specifically–routed frame), STE (Spanning Tree Explorer), or ARE (All Routes Explorer).

Cause: Received frame with RIF length less than 2 or not a multiple of 2 in length.

Action: Correct software in sending node.

SRT.054

Level: UI–ERROR

Short Syntax: SRT.054 No mem for hash tab (req *requested_size*), disabl

Long Syntax: SRT.054 No memory for hash table (desired size *requested_size* bytes), disabling bridge

Description: There is not enough free memory to allocate the hash table for the filtering database. The bridge will be disabled.

Cause: Severe shortage of memory.

Action: Reduce routing table sizes in other protocols, use system with less protocols, expand memory in router.

SRT.055

Level: UI–ERROR

Short Syntax: SRT.055 No mem for conv hash tab (req *requested_size*), disabl

Long Syntax: SRT.055 No memory for conversion hash table (desired size *requested_size* bytes), disabling bridge

Description: There is not enough free memory to allocate the hash table for the conversion database. The bridge will be disabled.

Cause: Severe shortage of memory.

Action: Reduce routing table sizes in other protocols, use system with less protocols, expand memory in router.

SRT.056

Level: CI–ERROR

Short Syntax: SRT.056 Input SR q ovf *source_mac–>dest_mac*, dropped, nt *network*

Long Syntax: SRT.056 Input source–routing queue overflow on frame from *source_mac* to *dest_mac*, packet dropped from network *network*

Description: The input queue for source–routed frames to be forwarded is too long, and this frame has been dropped to attempt to alleviate the congestion.

Cause: Bursty traffic may be causing congestion.

Action: Wait for burst to subside.

Cause: Too much traffic for forwarder to forward.

Action: Reconfigure network. Increase speed of router.

Cause: Inadequate buffer resources.

Action: Examine memory statistics in GWCON.

SRT.057

Level: P–TRACE

Short Syntax: SRT.057 *source_mac–>dest_mac* brdg port *port* nt *network* to port *port* nt *network*

Long Syntax: SRT.057 Frame from *source_mac* to *dest_mac* bridged from port number port network *network* to port number port network *network*

Description: A frame is being bridged between these two interfaces. The destination address was known, so it was sent only to the correct destination network.

SRT.058

Level: CE–ERROR

Short Syntax: SRT.058 TB–>SR *source_mac–>dest_mac* too big (*reformatted_length* > *output_maximum*) for port *port* nt *network*, drop

Long Syntax: SRT.058 Transparent frame converted to source–routed frame from *source_mac* to *dest_mac* is too big (reformatted length *reformatted_length* bytes > output maximum size *output_maximum* bytes) for port *port* network *network*, dropped

Description: The specified transparent bridge frame is too large to send as a source–routed frame on this outgoing port and network. The *reformatted_length* is the size of the frame including MAC headers and RIF after any mapping of data link headers.

Cause: Host on network with large maximum frame size sending to host on network with smaller maximum frame size.

Action: Reconfigure sending host to not send such large frames. If frame is of a routable protocol supporting fragmentation (such as IP or ISO) or maximum frame size determination (DNA or XNS), convert to using routing instead of bridging.

Cause: Host on network with large maximum frame size sending to host via an intervening network with smaller maximum frame size.

Action: Reconfigure network to use networks with large maximum frame size (such as FDDI or 802.5) as the backbone networks. Reconfigure port costs in Spanning Tree Protocol to favor spanning trees via networks with large maximum frame sizes.

SRT.059

Level: P–TRACE

Short Syntax: SRT.059 TB–>SR *source_mac–>dest_mac* (RIF *RIF*) brdg port *port* nt *network* to port *port* nt *network*

Long Syntax: SRT.059 Transparent frame converted to source–routed frame from *source_mac* to *dest_mac* (RIF *RIF*) bridged from port number port network *network* to port number port network *network*

Description: A frame is being conversion bridged between these two interfaces. The destination address and RIF were known, so it was sent only to the correct destination network.

SRT.060

Level: P–TRACE

Short Syntax: SRT.060 TB–>SR *source_mac–>dest_mac* (RIF *RIF*) brdg–all port *port* nt *network* to port *port* nt *network*

Long Syntax: SRT.060 Transparent frame converted to source–routed frame from *source_mac* to *dest_mac* (RIF *RIF*) bridged to all ports from port number port network *network* to port number port network *network*

Description: A frame is being conversion bridged to all active source-routing ports. This happens when the frame destination is a multicast or when the frame destination is not in the source–routing learning database. There will be one message for each port the frame is sent on.

SRT.061

Level: UE–ERROR

Short Syntax: SRT.061 SRF rcv *source_mac–>dest_mac* (RIF *RIF*) to disabl port *port*, nt *network*, disc

Long Syntax: SRT.061 Specifically routed frame frame received from *source_mac* to *dest_mac* (RIF *RIF*) to disabled port *port*, network *network*, discarded

Description: A Specifically Routed frame has been received whose RIF would send it on the specified port, but that port is not configured for bridging.

Cause: End station using invalid RIF. This can happen when the end station acquires a RIF, and caches it, but in the interim the bridge has been reconfigured and restarted.

SRT.062

Level: CE–ERROR

Short Syntax: SRT.062 Warning:SR–>TB *source_mac–>dest_mac* too big (*reformatted_length* > *output_maximum*) from port *port* nt *network*

Long Syntax: SRT.062 Source–routed frame converted to transparent frame from *source_mac* to *dest_mac* is too big (reformatted length *reformatted_length* bytes > output maximum size *output_maximum* bytes) from port *port* network *network*, may get dropped.

Description: The specified source–routed frame is larger than that is allowed by LF–BIT configuration for the transparent bridge domain. After mapping to the MAC headers of the outgoing port, the packet may get dropped if it exceeds the MSDU limit of the port.

Cause: Source–routing host not honoring maximum frame size that was determined in source–routing threading process.

Action: Correct behavior of host.

Cause: Host on network with large maximum frame size sending to host on network with smaller maximum frame size.

Action: Reconfigure sending host to not send such large frames. If frame is of a routable protocol supporting fragmentation (such as IP or ISO) or maximum frame size determination (DNA or XNS), convert to using routing instead of bridging.

Cause: Host on network with large maximum frame size sending to host via an intervening network with smaller maximum frame size.

Action: Reconfigure network to use networks with large maximum frame size (such as FDDI or 802.5) as the backbone networks. Reconfigure port costs in Spanning Tree Protocol to favor spanning trees via networks with large maximum frame sizes.

SRT.063

Level: UI–ERROR

Short Syntax: SRT.063 No buf to dup *routing_type* frame *source_mac*– >*dest_mac* for SR–>TB from port *port* nt *network*

Long Syntax: SRT.063 No buffer available to duplicate *routing_type* frame from *source_mac* to *dest_mac* for source–routing to transparent bridging conversion from port *port* network *network*

Description: No buffer available to copy a frame in order to send Routes Explorer (ARE) or Spanning Tree Explorer (STE) *routing_type* frame out as a transparent bridged frame in the transparent bridging domain. No copy of this frame will be sent into the transparent bridge domain.

Cause: Severe packet buffer shortage.

Action: Check memory statistics in GWCON to verify packet buffer level.

Cause: Traffic peak using all available buffers.

Action: This is the problem if this message occurs infrequently.

SRT.064

Level: UI–ERROR

Short Syntax: SRT.064 No mem for conv db (req *requested_size*), disabl

Long Syntax: SRT.064 No memory for conversion databse (desired size *requested_size* bytes), disabling bridge

Description: There is not enough free memory to allocate even a minimal size conversion database. The bridge will be disabled.

Cause: Severe shortage of memory.

Action: Reduce routing table sizes in other protocols, use system with less protocols, expand memory in router.

SRT.065

Level: UI–ERROR

Short Syntax: SRT.065 Can’t add stat ent *MAC_address* on nt *network*

Long Syntax: SRT.065 Can not add static entrie for address *MAC_address* on network *network*

Description: An attempt to add a particular static entry to the internal database of a bridging interface having internal filtering failed.

Cause: Hardware failure or software bug.

Action: Contact customer service.

SRT.066

Level: UI–ERROR

Short Syntax: SRT.066 Can't ena TB on nt *network*

Long Syntax: SRT.066 Can not enable transparent bridging on network *network*

Description: The bridge has been configured to enable transparent bridging on an IEEE 802.5 Token–Ring network that does not have the hardware to support transparent bridging. Transparent bridging will not be enabled on this interface.

Cause: Misconfiguration.

Action: Correct configuration.

SRT.067

Level: UI–ERROR

Short Syntax: SRT.067 SRF *source_mac–>dest_mac* (RIF *RIF*) fwd to disabl port *port*, nt *network*, disc

Long Syntax: SRT.067 Specifically routed frame frame from *source_mac* to *dest_mac* (RIF *RIF*) forwarded to disabled port *port*, network *network*, discarded

Description: A Specifically Routed frame has been sent on a port, but that port is not configured for bridging. This should never happen, since prior checks should prevent calling this code if the port is not configured for bridging.

SRT.068

Level: UI–ERROR

Short Syntax: SRT.068 Eth type table full for *ethernet_type*

Long Syntax: SRT.068 Ethernet type table full for Ethernet type *ethernet_type*

Description: There is no space in the Ethernet type registration table for the specified *ethernet_type*. This happens when there are too many hash collisions, and there are not enough overflow buckets.

Cause: Too many added Ethernet type filters.

Action: Do not use as many Ethernet type filters.

SRT.069

Level: UI–ERROR

Short Syntax: SRT.069 SNAP type table full for PID protocol

Long Syntax: SRT.069 Subnetwork Access Protocol table full for Protocol Identifier type protocol

Description: There is no space in the SNAP PID registration table for the specified protocol. This happens when there are too many hash collisions, and there are not enough overflow buckets.

Cause: Too many added SNAP PID filters.

Action: Do not use as many SNAP PID filters.

SRT.070

Level: P–TRACE

Short Syntax: SRT.070 *source_mac–>dest_mac* drp, dst add flt, port *port* nt *network*

Long Syntax: SRT.070 Frame from *source_mac* to *dest_mac* dropped, destination address filtered, port *port* network *network*

Description: A MAC frame has been received by the hardware, but is being dropped because the destination MAC address is being administratively filtered by the bridge. The frame will be dropped.

Cause: Receipt of frame whose destination MAC address matches the exclusive filter.

SRT.071

Level: UI–ERROR

Short Syntax: SRT.071 SR not supp on port *port*, net *network*

Long Syntax: SRT.071 Source Routing not supported on port *port*, net *network*

Description: Source Routing is configured on the port which is attached to an underlying network which inherently does not support source routing type of functionalities. Such networks are Ethernet and FDDI. Bridge disables source routing on the port.

Cause: User misconfiguration.

SRT.072

Level: UI–ERROR

Short Syntax: SRT.072 Conversion enabled, but not licensed, disabling

Long Syntax: SRT.072 Conversion bridging (Adaptive or SR–TB) enabled, but not licensed, disabling

Description: Conversion bridging has been enabled, but that feature was not purchased as part of this software load. The conversion bridging feature will not be enabled.

Cause: Enabling feature that was not purchased.

Action: Buy software with feature.

SRT.073

Level: UI–ERROR

Short Syntax: SRT.073 SRB enabled, but not licensed, disabling

Long Syntax: SRT.073 Source–routing bridging enabled, but not licensed, disabling

Description: Source–routing bridging has been enabled, but that feature was not purchased as part of this software load. The source–routing bridging feature will not be enabled.

Cause: Enabling feature that was not purchased.

Action: Buy software with feature.

SRT.074

Level: UI–ERROR

Short Syntax: SRT.074 SRB enabled on int *network*, but not licensed, disabling

Long Syntax: SRT.074 Source–routing bridging enabled on int *network*, but not licensed, disabling

Description: Source–routing bridging has been enabled, but that feature was not purchased as part of this software load. The source–routing bridging feature will not be enabled.

Cause: Enabling feature that was not purchased.

Action: Buy software with feature.

SRT.075

Level: UI–ERROR

Short Syntax: SRT.075 STB enabled on int *network*, but not licensed, disabling

Long Syntax: SRT.075 Spanning tree (transparent) bridging enabled on int *network*, but not licensed, disabling

Description: Spanning tree (transparent) bridging has been enabled, but that feature was not purchased as part of this software load. The spanning tree (transparent) bridging feature will not be enabled.

Cause: Enabling feature that was not purchased.

Action: Buy software with feature.

SRT.076

Level: UI–ERROR

Short Syntax: SRT.076 no mem to alloc NB flt

Long Syntax: SRT.076 No memory to allocate a NETBIOS Filter

Description: At least one configured NETBIOS Filter will not be enabled, because there is not enough memory.

Cause: Insufficient free memory.

Action: Increase memory size.

SRT.077

Level: U–INFO

Short Syntax: SRT.077 input_output NB flt lst, port *port_number*, dlted

Long Syntax: SRT.077 input_output NETBIOS filter list, for port *port_number*, deleted by user. Filter will not be enabled

Description: The user deleted a filter list, which was part of an already configured filter. The filter will not be enabled.

Cause: User configuration error.

Action: Reconfigure the filter list that was deleted.

SRT.078

Level: U–INFO

Short Syntax: SRT.078 input_output NB flt configd for port *port_number*, port doesnt exist

Long Syntax: SRT.078 input_output NETBIOS filter for port *port_number* is configured, but that port number is not configured

Description: The user configured a NETBIOS filter for a particular port, but that port number is not configured.

Cause: User configuration error.

Action: Either reconfigure the NETBIOS filter for the correct port number, or add to the SRT configuration the port number that was configured in the NETBIOS filter.

SRT.079

Level: C–TRACE

Short Syntax: SRT.079 NB outp pkt fltd *source_mac–>dest_mac*, prt port, nt *network*

Long Syntax: SRT.079 NETBIOS Output Packet Filtered – *source_mac– >dest_mac* , port *port*, network *network*

Description: A NETBIOS packet has matched the criteria specified in a NETBIOS Filter configuration record. The packet is dropped.

SRT.080

Level: UI–ERROR

Short Syntax: SRT.080 no mem to alloc NB cnsl info

Long Syntax: SRT.080 No memory to allocate information for NETBIOS Filter console display

Description: The part of the router that handles NETBIOS console display cannot allocate enough memory to do the complete display. Some part of the NETBIOS console display will not be shown from the T 5 process.

Cause: Insufficient free memory.

Action: Increase memory size.

SRT.081

Level: P–TRACE

Short Syntax: SRT.081 NB STE converted to SRF (RIF *RIF*) *source_mac– >dest_mac* from port *port*

Long Syntax: SRT.081 NETBIOS STE converted to SRF (RIF *RIF*) *source_mac– >dest_mac* from port *port*

Description: A NETBIOS STE converted to SRF by NETBIOS Name Caching

SRT.082

Level: P–TRACE

Short Syntax: SRT.082 NB STE not converted, RIF too long

Long Syntax: SRT.082 NETBIO STE not converted, RIF too long

Description: NETBIO STE not converted, RIF too long

SRT.083

Level: P–TRACE

Short Syntax: SRT.083 NB find–name STE filtered (RIF *RIF*) *source_mac– >dest_mac* from port *port*

Long Syntax: SRT.083 NETBIOS find–name STE filtered (RIF *RIF*) *source_mac–>dest_mac* from port *port*

Description: A NETBIOS find–name STE has been filtered

Panic "SRTimem"

Short Syntax: SRT: memory allocation failed

Description: The SRT forwarder failed to allocate sufficient memory to hold its most fundamental tables.

Cause: Insufficient free memory.

Action: Making databases for other protocols smaller.

Action: Increase memory size.

Fatal "srtiisrt"

Short Syntax: SRT: Invalid i_srt on input

Description: The i_srt flag passed from the handler to forwarder has an invalid value.

Cause: Software bug.

Action: Take a crash dump and contact customer service.

Fatal "srtuimed"

Short Syntax: SRT: unknown input media

Description: The input net type is not one of the ones understood by the SRT bridge (802.3/Ethernet, FDDI, or 802.5).

Cause: Software bug.

Action: Take a crash dump and contact customer service.

STP

Spanning Tree Protocol

This chapter describes Spanning Tree Protocol (STP) messages. The Spanning Tree Protocol is used by the SRT Bridge to form a loop–free topology. For information about message content and how to use the message, refer to the Introduction

STP.001

Level: C–TRACE

Short Syntax: STP.001 Cfg BPDU rcv frm *source_address bridge_type– bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.001 Configuration BPDU received frm *source_address* on *bridge_type–bridge_instance* port *bridge_port*, network *network*

Description: A configuration BPDU has been received from the specified MAC address.

Cause: Another bridge on the same network as this bridge on this port.

STP.002

Level: C–TRACE

Short Syntax: STP.002 Tcn BPDU rcv frm *source_address bridge_type– bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.002 Topology change notification BPDU received frm *source_address* on *bridge_type–bridge_instance* port *bridge_port*, network *network*

Description: A topology change notification BPDU has been received from the specified MAC address.

Cause: Topology change has been detected at or downstream of the sending bridge.

Action: None needed, the message should stop when the topology change is acknowledged by the root bridge.

STP.003

Level: UE–ERROR

Short Syntax: STP.003 Ukn BPDU type *BDPU_type* rcv frm *source_address* *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.003 Unkown BPDU type *BDPU_type* received frm *source_address* on *bridge_type–bridge_instance* port *bridge_port*, network *network*

Description: An BPDU with an undefined value in the BPDU Type field was received from the specified host. It will be ignored.

Cause: Programming error at remote bridge.

Action: Correct remote node.

Cause: Data corruption in received packet.

Action: Eliminate source of data corruption.

STP.004

Level: UE–ERROR

Short Syntax: STP.004 BPDU bd ID *Protocol_Identifier* frm *source_address* *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.004 BPDU bad protocol identifier *Protocol_Identifier* frm *source_address* on *bridge_type–bridge_instance* port *bridge_port*, network *network*

Description: A configuration BPDU has been received with a Protocol Identifier that is not 0000. It will be ignored.

Cause: Programming error at remote bridge.

Action: Correct remote node.

Cause: Data corruption in received packet.

Action: Eliminate source of data corruption.

STP.005

Level: UE–ERROR

Short Syntax: STP.005 BPDU bd ver *Protocol_Version_Identifier* frm *source_address* *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.005 BPDU bad Version *Protocol_Version_Identifier* frm *source_address* on *bridge_type–bridge_instance* port *bridge_port*, network *network*

Description: A configuration BPDU has been received with a Protocol Version Identifier that is not 00. It will be ignored.

Cause: Programming error at remote bridge.

Action: Correct remote node.

Cause: Data corruption in received packet.

Action: Eliminate source of data corruption.

STP.006

Level: UE–ERROR

Short Syntax: STP.006 Cfg BPDU trunc (length byt) frm *source_address* *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.006 Configuration BPDU truncated (*length* bytes) frm *source_address* on *bridge_type–bridge_instance* port *bridge_port*, network *network*

Description: A configuration BPDU has been received which is less than 35 bytes in length. It will be ignored.

Cause: Programming error at remote bridge.

Action: Correct remote node.

Cause: Data corruption in received packet.

Action: Eliminate source of data corruption.

STP.007

Level: UE–ERROR

Short Syntax: STP.007 Cfg BPDU unk flg *flags* frm *source_address* *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.007 Configuration BPDU unknown flags *flags* frm *source_address* on *bridge_type–bridge_instance* port *bridge_port*, network *network*

Description: A configuration BPDU has been received which has undefined bits set in the flags field. It will be ignored.

Cause: Programming error at remote bridge.

Action: Correct remote node.

Cause: Data corruption in received packet.

Action: Eliminate source of data corruption.

STP.008

Level: UE–ERROR

Short Syntax: STP.008 Tcn BPDU trunc (*length* byt) frm *source_address* *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.008 Topology change notification BPDU truncated (*length* bytes) frm *source_address* on *bridge_type–bridge_instance* port *bridge_port*, network *network*

Description: A topology change notification BPDU has been received that is less than 4 bytes in length. It will be ignored.

Cause: Programming error at remote bridge.

Action: Correct remote node.

Cause: Data corruption in received packet.

Action: Eliminate source of data corruption.

STP.009

Level: UI–ERROR

Short Syntax: STP.009 No buf for BPDU *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.009 No buffer to send BDPU on *bridge_type– bridge_instance* port *bridge_port*, network *network*

Description: No packet buffer was available to construct and send a BDPU on the specified port.

Cause: Severe packet buffer shortage.

Action: Check memory statistics in GWCON to verify packet buffer level.

Cause: Traffic peak using all available buffers.

Action: This is the problem if this message occurs infrequently.

STP.010

Level: P–TRACE

Short Syntax: STP.010 Sndg cfg BPDU *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.010 Sending Configuration BPDU on *bridge_type– bridge_instance* port *bridge_port* network *network*

Description: A Configuration BPDU will be sent on the specified port. This is done normally on a periodic basis as part of the spanning tree protocol. The flags field in this BPDU is zero, e.g., neither the Topology Change or the Topology Change Acknowledgement bits are set.

STP.011

Level: P–TRACE

Short Syntax: STP.011 Sndg Cfg BPDU flgs *TC TCA bridge_type– bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.011 Sending Configuration BPDU with flags *TC TCA* on *bridge_type–bridge_instance* port *bridge_port*, network *network*

Description: A Configuration BPDU will be sent on the specified port. This is done normally on a periodic basis as part of the spanning tree protocol. TC will be displayed if the Topology Change bit is set in the Flags byte of the BPDU, TCA will be displayed if the Topology Change Acknowledge bit is set in the flags byte.

Cause: The Topology Change flag is set if this bridge is the root and it knows that there is a topology change in process. Also, non–root bridges propagate this bit received in incoming Configuration BPDUs.

Action: None needed, this flag will be set only for the sum of the current maximum age and current forward delay parameters (as propagate by the root bridge).

Cause: The Topology Change Acknowledge flag is set if this bridge has received a Topology Change Notification BPDU, and this port is the Designated Bridge on its LAN.

Action: None needed, this flag will only be sent on one BDPU.

STP.012

Level: P–TRACE

Short Syntax: STP.012 Sndg tcn BPDU *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.012 Sending Topology Change Notification BPDU on *bridge_type–bridge_instance* port *bridge_port* network *network*

Description: A Topology Change Notification BPDU will be sent on the specified port. These are sent on the root port of non–root ports when they detect a topology change in the spanning tree.

Cause: A bridge, or an interface on a bridge, has gone up or down in this spanning tree.

Action: None needed. This state persists only until a topology change acknowledgement is received, or a timeout that indicates that the old root bridge is no longer reachable.

STP.013

Level: UI–ERROR

Short Syntax: STP.013 BPDU snd fld, rsn *reason_code*, *bridge_type– bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.013 BPDU send failed for reason code *reason_code* on *bridge_type–bridge_instance* port *bridge_port* network *network*

Description: The attempt to queue a BPDU for transmission on the specified port failed.

Cause: Miscellaneous handler error. (Reason code 1.)

Action: Check for error messages from handler for network.

Cause: Output queue overflow, or other flow control. (Reason code 2.)

Action: Alleviate congestion.

Cause: Network down. (Reason code 3.)

Action: See why handler thinks network is down.

Cause: Dropped by handler to avoid looping, or bad broadcast. (Reason code 4.)

Action: Check configuration.

Cause: Host down. (Reason code 5.)

Action: See why handler thinks host is down.

STP.014

Level: U–INFO

Short Syntax: STP.014 Blocking *bridge_type–bridge_instance* port *bridge_port*, nt *network*, det topol chg

Long Syntax: STP.014 Blocking *bridge_type–bridge_instance* port *bridge_port*, network *network*, detecting topology change

Description: This port has just been placed in Blocking state. This is a change in the topology, so this bridge detects a topology change. This will in turn cause topology change notifications to be sent.

Cause: A bridge, or an interface on a bridge, has gone up or down in this spanning tree.

Action: None needed. This is normal when there are changes.

STP.015

Level: U–INFO

Short Syntax: STP.015 Topol chg detected *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.015 Topology change detected on *bridge_type– bridge_instance* port *bridge_port*, network *network*

Description: A topology change notification has been received on this port, and this port is the designated port on its LAN. This causes the protocol to enter topology change notification state. The topology change will be acknowledged towards the sender, and propagated towards the root.

Cause: A bridge, or an interface on a bridge, has gone up or down in this spanning tree.

Action: None needed. This is normal when there are changes.

STP.016

Level: U–INFO

Short Syntax: STP.016 Select as root *bridge_type–bridge_instance*, det topol chg

Long Syntax: STP.016 Selected as root on *bridge_type–bridge_instance*, detecting topology change

Description: This bridge has just selected itself as the root of the spanning tree when it previously had not been. This causes the bridge to enter topology change notification state.

Cause: A bridge, or an interface on a bridge, has gone up or down in this spanning tree.

Action: None needed. This is normal when there are changes.

Cause: This is the first bridge up, thus it is the root of the tree.

STP.017

Level: C–INFO

Short Syntax: STP.017 Tply chg ackd *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.017 Topology change acknowledged on *bridge_type– bridge_instance* port *bridge_port*, network *network*

Description: A topology change acknowledgement has been detected on the specified port. This port is the root port of the bridge.

Cause: Bridge on same LAN as our root port has set topology change acknowledgement flag in outgoing Configuration BDPU. This was in response to a topology change notification that this bridge originated or propagated.

Action: None needed. This is the normal conclusion of topology change notification.

STP.018

Level: C–INFO

Short Syntax: STP.018 Acking tply chg *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.018 Acknowledging topology change on *bridge_typebridge_instance* port *bridge_port*, network *network*

Description: A topology change notification is being acknowledged on the specified port. This is done when a topology change notification is received on a port that is the designated port for that LAN.

Cause: Change on bridge topology downstream of this bridge.

Action: None needed. This is a normal port of reconfiguration of the spanning tree.

STP.019

Level: C–TRACE

Short Syntax: STP.019 Tplgy chg notif timer expired *bridge_type– bridge_instance*

Long Syntax: STP.019 Topology Change Notification timer expired on *bridge_type–bridge_instance*

Description: The Topology Change timer expired. This bridge will cease sending topology change notification BPDU's on its root port.

Cause: This timer expires when the bridge has been in Topology Change Notification state for the bridge hello timer period.

Action: None needed, this is the normal conclusion of this state.

STP.020

Level: C–TRACE

Short Syntax: STP.020 Tplgy chg timer expired *bridge_type–bridge_instance*

Long Syntax: STP.020 Topology Change timer expired on *bridge_type– bridge_instance*

Description: The Topology Change timer expired. This bridge, which is the root, will cease sending the Topology Change in its Configuration BPDUs.

Cause: This happens when this root bridge has been in Topology Change state for the sum of current maximum age and current forward delay parameters.

Action: None needed, this is the normal conclusion of this state.

STP.021

Level: U–INFO

Short Syntax: STP.021 Msg age tmr exp *bridge_type–bridge_instance* port *bridge_port*, nt *network*, try Root

Long Syntax: STP.021 Message age timer expired on *bridge_type– bridge_instance* port *bridge_port*, network *network*, will try and become root

Description: The message age timer has expired on this port. The bridge will attempt to become the root. It will become the designated port on that LAN.

Cause: No Configuration BPDU's being received on this interface. Either there are no bridges on this LAN, or they are down.

STP.022

Level: C–TRACE

Short Syntax: STP.022 Hello tmr exp *bridge_type–bridge_instance*

Long Syntax: STP.022 Hello timer expired on *bridge_type–bridge_instance*

Description: The hello timer has expired on this port. Configuration BPDUs will be sent on all ports.

STP.023

Level: C–TRACE

Short Syntax: STP.023 Stop msg age tmr *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.023 Stopping message age timer for *bridge_type– bridge_instance* port *bridge_port*, network *network*

Description: Stopping the message age timer on this port because is it the designated port on its LAN.

STP.024

Level: U–INFO

Short Syntax: STP.024 Not root *bridge_type–bridge_instance*, stop hello tmr

Long Syntax: STP.024 Not root anymore on *bridge_type–bridge_instance*, stopping hello timer

Description: This bridge has just decided that it is no longer the root bridge of the spanning tree. The hello timer will also be cancelled.

STP.025

Level: C–INFO

Short Syntax: STP.025 Stop tplgy chg age tmr *bridge_type–bridge_instance*

Long Syntax: STP.025 Stopping topology change timer for *bridge_type– bridge_instance*

Description: Stopping the topology change timer because this bridge is no longer the root.

STP.026

Level: U–INFO

Short Syntax: STP.026 Root *bridge_type–bridge_instance*, strt hello tmr

Long Syntax: STP.026 Selected as root on *bridge_type–bridge_instance*, starting hello timer

Description: This bridge has just decided that it is the root bridge of the spanning tree. The hello timer will be started.

STP.027

Level: C–TRACE

Short Syntax: STP.027 Strt msg age tmr *bridge_type–bridge_instance* port *bridge_port*, nt *network*

Long Syntax: STP.027 Starting message age timer for *bridge_type– bridge_instance* port *bridge_port*, network *network*

Description: Starting the message age timer on this port.

STP.028

Level: U–INFO

Short Syntax: STP.028 Attmpt root *bridge_type–bridge_instance*, strt hello tmr

Long Syntax: STP.028 Attempting to become root on *bridge_type– bridge_instance*, starting hello timer

Description: This bridge is attempting to become the root bridge of the spanning tree. The hello timer will be started.

STP.029

Level: UI–ERROR

Short Syntax: STP.029 Cfg BPDU frm *source_address* ign *bridge_type– bridge_instance*, inact port *bridge_port*, nt *network*

Long Syntax: STP.029 Configuration BPDU from *source_address* on *bridge_type–bridge_instance* ignored, inactive port *bridge_port*, network *network*

Description: A configuration BPDU has been received from the specified MAC address, but the port is not participating in the spanning tree protocol.

STP.030

Level: UI–ERROR

Short Syntax: STP.030 Tcn BPDU frm *source_address* ign *bridge_type– bridge_instance*, inact port *bridge_port*, nt *network*

Long Syntax: STP.030 Topology change notification BPDU from *source_address* on *bridge_type–bridge_instance* ignored, inactive port *bridge_port*, network *network*

Description: A topology change notification BPDU has been received from the specified MAC address, but the port is not participating in the spanning tree protocol.

STP.031

Level: C–INFO

Short Syntax: STP.031 *bridge_type–bridge_instance* desig port *bridge_port*, nt *network*

Long Syntax: STP.031 *bridge_type–bridge_instance* becoming designated port *bridge_port*, network *network*

Description: This bridge is declaring itelf the designated port on the LAN connected to this port.

Fatal “stpubpdu”

Short Syntax: Attempt to send unknown BPDU type

Description: The code attempted to send an unknown type of BPDU.

Cause: Possible software bug.

Action: Get crash dump, contact customer service.

TCP

Transmission Control Protocol

This chapter describes Transmission Control Protocol (TCP) messages. TCP is part of the IP protocol. For information about message content and how to use the message, refer to the Introduction.

TCP.001

Level: UI–ERROR

Short Syntax: TCP.001 pkt *cksum* fld pkt = *tcp_checksum* calc = *tcp_checksum*

Long Syntax: TCP.001 packet checksum failed received packet checksum is *tcp_checksum* and calculated checksum is *tcp_checksum*

Description: Checksum failed because received packet checksum is not equal to the calculated checksum

TCP.002

Level: UI–ERROR

Short Syntax: TCP.002 rcvd pkt *source_ip_address –> destination_ip_address* dst prt *tcp_port* no cnn

Long Syntax: TCP.002 received packet *source_ip_address –> destination_ip_address* with destination port *tcp_port* has no tcp connection

Description: TCP has received a packet with an invalid tcp port number.

TCP.003

Level: C–INFO

Short Syntax: TCP.003 Act opn sccfl dst prt *tcp_port*

Long Syntax: TCP.003 TCP Active open successful for port number *tcp_port*

Description: Active open was successful and we are notifying application of the open.

TCP.004

Level: UI–ERROR

Short Syntax: TCP.004 rcvd invld SYN in wndw *source_ip_address –> destination_ip_address* dst prt *tcp_port* kill cnn

Long Syntax: TCP.004 received invalid SYN packet *source_ip_address –> destination_ip_address* with destination port *tcp_port*, kill connection

Description: TCP has received an illegal SYN packet, so kill the connection.

TCP.005

Level: UI–ERROR

Short Syntax: TCP.005 rcvd old SYN *source_ip_address –> destination_ip_address* dst prt *tcp_port* snd ACK

Long Syntax: TCP.005 received old duplicate SYN packet *source_ip_address –> destination_ip_address* with destination port *tcp_port*, send ACK packet in response

Description: TCP has received an old duplicate SYN, so send ACK with received sequence number; this forces the other side to do a RST.

TCP.006

Level: UI–ERROR

Short Syntax: TCP.006 rcvd out of wndow seg *source_ip_address –> destination_ip_address* dst prt *tcp_port* snd ACK

Long Syntax: TCP.006 received an out of window segment *source_ip_address –> destination_ip_address* with destination port *tcp_port*, send a valid ACK

Description: TCP has received an out of window segment; send ACK in response.

TCP.007

Level: UI–ERROR

Short Syntax: TCP.007 drp seg *source_ip_address –> destination_ip_address* dst prt *tcp_port* rsn *reject_code* snd ACK

Long Syntax: TCP.007 dropped segment *source_ip_address –> destination_ip_address* with destination port *tcp_port*, reason *reject_code*, send a valid ACK in response

Description: TCP has rejected a segment. Reject codes are as follows: Reject codes: 1 – Seg len = 0, Rcv win > 0, seqnum = winend 2 – Seg len = 0, Rcv win = 0, seqnum != tcb_ack 3 – Seg len > 0, Rcv win > 0, inend = winend 4 – Seg len > 0, Rcv win = 0. Note: we only ACK if the segment received was a non RST segment.

TCP.008

Level: UI–ERROR

Short Syntax: TCP.008 rcvd old seg dst prt *tcp_port* seq num *seq_num* snd ACK

Long Syntax: TCP.008 received old duplicate packet with destination port *tcp_port*, sequence number *seq_num*, send ACK in response

Description: TCP has received an old segment that has already been consumed by the application, so send ACK in response.

TCP.009

Level: C–INFO

Short Syntax: TCP.009 state LISTEN: rcvd RST dst prt *tcp_port* seq num *seq_num*

Long Syntax: TCP.009 while in LISTEN state, received RST with destination port *tcp_port*, sequence number *seq_num*; drop segment

Description: TCP has received a RST while in LISTEN state; just ignore packet.

TCP.010

Level: C–INFO

Short Syntax: TCP.010 state SYN_RCVD|SYN_SNT: rcvd RST dst prt *tcp_port* seq num *seq_num*, rtrn to LISTEN

Long Syntax: TCP.010 while in SYN_RECEIVED or SYN_SENT states, received RST with destination port *tcp_port*, sequence number *seq_num*; drop segment and return to LISTEN state

Description: TCP has received a RST while in SYN_RECEIVED or SYN_SENT states; drop packet and return to LISTEN state.

TCP.011

Level: C–INFO

Short Syntax: TCP.011 rcvd RST dst prt *tcp_port* seq num *seq_num*, abort

Long Syntax: TCP.011 received RST with destination port *tcp_port*, sequence number *seq_num*; drop segment and abort connection

Description: TCP has received a RST; abort connection.

TCP.012

Level: UI–ERROR

Short Syntax: TCP.012 drop seg dst prt *tcp_port* seq num *seq_num* no ACK present

Long Syntax: TCP.012 drop segment with destination port *tcp_port*, sequence number *seq_num* because no ACK is present

Description: TCP has stopped processing the packet because there is no ACK present in the packet.

TCP.013

Level: UI–ERROR

Short Syntax: TCP.013 drop seg dst prt *tcp_port* seq num *seq_num* ack num *ack_num* rcv invld ACK

Long Syntax: TCP.013 drop segment with destination port *tcp_port*, sequence number *seq_num*, acknowledge number *ack_num*, received invalid ACK

Description: Stop processing the segment because it contains acknowledgement for data not yet sent.

TCP.014

Level: C–INFO

Short Syntax: TCP.014 state ESTAB: rcvd FIN dst prt *tcp_port* seq num *seq_num*

Long Syntax: TCP.014 while in ESTABLISHED state, received FIN with destination port *tcp_port*, sequence number *seq_num*

Description: TCP has received a FIN while in ESTABLISHED state; when all data has been received, send FIN|ACK.

TCP.015

Level: C–INFO

Short Syntax: TCP.015 rcvd PSH dst prt *tcp_port* seq num *seq_num*

Long Syntax: TCP.015 received a segment with the PSH bit set with destination port *tcp_port*, sequence number *seq_num*

Description: TCP has received a segment with PSH bit set

TCP.016

Level: C–INFO

Short Syntax: TCP.016 state SYNRCVD: rcvd vld seg dst prt *tcp_port* seq num *seq_num*, enter ESTAB

Long Syntax: TCP.016 while in SYNRCVD state, received valid segment with destination port *tcp_port*, sequence number *seq_num*, so enter ESTABLISHED state

Description: TCP has received a valid segment while in SYNRCVD state; enter ESTABLISHED state and notify application of the open.

TCP.017

Level: UI–ERROR

Short Syntax: TCP.017 rcvd FIN while in LISTEN dst prt *tcp_port* seq num *seq_num*, snd RST

Long Syntax: TCP.017 received FIN segment while in the LISTEN state, destination port *tcp_port*, sequence number *seq_num*, snd RST

Description: TCP has received a FIN while in the LISTEN state, so we send RST to the other side.

TCP.018

Level: C–INFO

Short Syntax: TCP.018 rcvd out of order seg dst prt *tcp_port* seq num *seq_num*, add hole at end *seq_num* to *seq_num*

Long Syntax: TCP.018 received an out of order segment with destination port *tcp_port*, sequence number *seq_num*; hole created at end of receive buffer seq num *seq_num* to *seq_num*

Description: TCP has received an out of order packet; this creates a hole in the receive buffer.

TCP.019

Level: C–INFO

Short Syntax: TCP.019 rcvd out of order seg dst prt *tcp_port* seq num *seq_num*, add hole at end *seq_num* to *seq_num*

Long Syntax: TCP.019 received an out of order segment with destination port *tcp_port*, sequence number *seq_num*; hole created at end of receive buffer seq num *seq_num* to *seq_num*

Description: TCP has received an out of order packet; this creates a hole in the receive buffer.

TCP.020

Level: C–INFO

Short Syntax: TCP.020 rcvd seg dst prt *tcp_port* seq num *seq_num*, prtally fill bgnng hole *seq_num* to *seq_num*

Long Syntax: TCP.020 received segment with destination port *tcp_port*, sequence number *seq_num*; partially fills the beginning of a hole *seq_num* to *seq_num*

Description: TCP has received a packet that partially fills the beginning of a hole.

TCP.021

Level: C–INFO

Short Syntax: TCP.021 rcvd seg dst prt *tcp_port* seq num *seq_num*, prtally fill end hole *seq_num* to *seq_num*

Long Syntax: TCP.021 received segment with destination port *tcp_port*, sequence number *seq_num*; partially fills the end of a hole *seq_num* to *seq_num*

Description: TCP has received a packet that partially fills the end of a hole.

TCP.022

Level: C–INFO

Short Syntax: TCP.022 rcvd seg dst prt *tcp_port* seq num *seq_num*, rmv hole *seq_num* to *seq_num*

Long Syntax: TCP.022 received segment with destination port *tcp_port*, sequence number *seq_num*; completely fills a hole, removing hole *seq_num* to *seq_num*

Description: TCP has received a packet that completely fills a hole.

TCP.023

Level: UI–ERROR

Short Syntax: TCP.023 drp seg dst prt *tcp_port* seq num *seq_num*, too big for rcv buff

Long Syntax: TCP.023 drop segment with destination port *tcp_port*, sequence number *seq_num*; segment too big for receive buffer

Description: TCP has received a packet that is too big to fit into the remaining space in the receive buffer.

TCP.024

Level: UI–ERROR

Short Syntax: TCP.024 prcss FIN in invld state

Long Syntax: TCP.024 process a received FIN; current state is not SYNRCVD|ESTAB, so do nothing

Description: TCP processing FIN while not in SYNRCVD|ESTAB state.

TCP.025

Level: C–INFO

Short Syntax: TCP.025 prcss FIN in ESTAB|SYNRCVD state frgn hst *ip_address* lcl hst *ip_address* dprt *dst_port* sprt *src_port*

Long Syntax: TCP.025 process a received FIN; current state is SYNRCVD|ESTAB, foreign host *ip_address* local host *ip_address* destination port *dst_port* source port *src_port*

Description: TCP processing FIN while in SYNRCVD|ESTAB state.

TCP.026

Level: C–INFO

Short Syntax: TCP.026 app rcv tmout

Long Syntax: TCP.026 application posted receive timeout has fired

Description: Application posts a read specifying a timeout value. If not all the requested data has been received within a timeout period, a timer fires, and whatever is in the receive buffer is given to the application.

TCP.027

Level: UI–ERROR

Short Syntax: TCP.027 frgn prt illgl close of wndw frgn hst *ip_address* lcl hst *ip_address* dprt *dst_port* sprt *src_port*

Long Syntax: TCP.027 foreign port closed the advertised window illegally foreign host *ip_address* local host *ip_address* destination port *dst_port* source port *src_port*

Description: The other side has been deaf and mute, and the the foreign window seems to have been closed illegally; send a RST.

TCP.028

Level: C–INFO

Short Syntax: TCP.028 state trnstn to SYNRCVD

Long Syntax: TCP.028 state of TCP connection transitioned to SYN–RECEIVED state

Description: State of the connection has transitioned to SYN–RECEIVED state as a result of either an active open or a passive open.

TCP.029

Level: C–INFO

Short Syntax: TCP.029 state trnstn to ESTAB

Long Syntax: TCP.029 state of TCP connection transitioned to ESTABLISHED state

Description: State of the connection has transitioned to ESTABLISHED state as a result of either an active open or a passive open.

TCP.030

Level: P–TRACE

Short Syntax: TCP.030 rcvd TCP pkt *source_ip_address –> destination_ip_address* dst prt *tcp_port*

Long Syntax: TCP.030 received packet *source_ip_address –> destination_ip_address* with destination port *tcp_port*

Description: TCP has received a packet.

TCP.031

Level: P–TRACE

***Short Syntax*:** TCP.031 seq num *seq_num* to *seq_num* given to app.

***Long Syntax*:** TCP.031 data with sequence number *seq_num* through to *seq_num* given to application

Description: Valid data in receive buffer has been handed to the application for further processing.

TCP.032

Level: C–INFO

***Short Syntax*:** TCP.032 excssv num rtries

***Long Syntax*:** TCP.032 excessive number of retries has occurred

Description: We have retransmitted a frame an excessive number of times. If the application has closed the connection already, just abort. Else, notify the application that there is a problem.

TCP.033

Level: P–TRACE

***Short Syntax*:** TCP.033 snd ctrl seg seq num *seq_num* ack num *ack_num* wndw *window*

***Long Syntax*:** TCP.033 send control segment with sequence number *seq_num* and acknowledge number *ack_num* window *window*

Description: Send a control segment to either ack a segment or send special control segments like FIN or RST.

TCP.034

Level: C–INFO

***Short Syntax*:** TCP.034 rxmt seq num *seq_num* to *seq_num*

***Long Syntax*:** TCP.034 retransmit data with sequence number *seq_num* through to *seq_num*

Description: We have failed to receive a valid ACK for transmitted data, so retransmit the data.

TCP.035

Level: P–TRACE

***Short Syntax*:** TCP.035 xmt seq num *seq_num* to *seq_num*

***Long Syntax*:** TCP.035 transmit data with sequence number *seq_num* through to *seq_num*

Description: Transmit data.

TCP.036

Level: UI–ERROR

Short Syntax: TCP.036 illgl optn rcvd in SYN seg

Long Syntax: TCP.036 illegal option received in SYN segment

Description: An unsupported option is present in the options field of a SYN packet.

TCP.037

Level: C–INFO

Short Syntax: TCP.037 zero wndw probe seq num *seq_num*

Long Syntax: TCP.037 zero window probe segment with sequence number *seq_num* sent

Description: The other side has advertised a zero window in the last segment received, so we have to send a zero window probe.

TCP.038

Level: UI–ERROR

Short Syntax: TCP.038 rjct seg dst prt *tcp_port* seq num *seq_num* bad ACK in SYNRCVD, snd RST

Long Syntax: TCP.038 reject segment with destination port *tcp_port* and sequence number *seq_num*, bad ACK in segment while in SYNRCVD state

Description: Reject the segment, and send a RST to the other side for receiving a segment with the incorrect acknowledgement while in the SYNRCVD state. Until a correct acknowledgement is received, we cannot progress into the ESTABLISHED state.

TCP.039

Level: UI–ERROR

Short Syntax: TCP.039 rcvd ACK seg with dst prt *tcp_port* seq num *seq_num* in LISTEN, snd RST

Long Syntax: TCP.039 received ACK segment with destination port *tcp_port*, sequence number *seq_num* while in the LISTEN state, send RST

Description: TCP has received an ACK while in the LISTEN state; this does not make any sense because we have not yet sent any data, so nothing should be ACKed. As a result, we send a RST.

TCP.040

Level: UI–ERROR

Short Syntax: TCP.040 TCP snd rst to hst *source_ip_address*

Long Syntax: TCP.040 TCP sending RESET to host *source_ip_address*

Description: TCP is sending a RESET segment to the other side.

TCP.041

Level: C–INFO

Short Syntax: TCP.041 TCP cnn clsd frgn hst *foreign_ip_address* lcl hst *local_ip_address*

Long Syntax: TCP.041 TCP connection closed, foreign host *foreign_ip_address*, local hst *local_ip_address*

Description: TCP connection is closed – notifying the application.

TCP.042

Level: C–INFO

Short Syntax: TCP.042 Frng TCB for frgn hst *foreign_ip_address* lcl hst *local_ip_address*

Long Syntax: TCP.042 Freeing TCB block for connection between *foreign_ip_address* and *local_ip_address*

Description: Freeing the TCB block associated with the TCP connection that has closed.

TCP.043

Level: C–INFO

Short Syntax: TCP.043 Frng TCB for frgn hst *foreign_ip_address* lcl hst *local_ip_address*

Long Syntax: TCP.043 Freeing TCB block for connection between *foreign_ip_address* and *local_ip_address*

Description: Freeing the TCB block associated with the TCP connection that has closed.

TCP.044

Level: C–INFO

Short Syntax: TCP.044 Idle tmr fires frgn hst *foreign_ip_address* lcl hst *local_ip_address*

Long Syntax: TCP.044 Idle timer fires for connection between *foreign_ip_address* and *local_ip_address*

Description: Idle timer fires for TCP connection.

TCP.045

Level: C–INFO

Short Syntax: TCP.045 Rxmt tmr fires frgn hst *foreign_ip_address* lcl hst *local_ip_address*

Long Syntax: TCP.045 Retransmit timer fires for connection between *foreign_ip_address* and *local_ip_address*

Description: Retransmit timer fires for TCP connection.

TCP.046

Level: C–INFO

Short Syntax: TCP.046 State trnstn frm ESTAB to FINWAIT *source_ip_address –> destination_ip_address* dst prt *tcp_src_port* src prt *tcp_dst_port*

Long Syntax: TCP.046 State transitioned from ESTABLISHED to FINWAIT *source_ip_address –> destination_ip_address* dst prt *tcp_src_port* src prt *tcp_dst_port*

Description: State of tcp connection has transitioned from ESTABLISHED to FINWAIT – send FIN, and now waiting for FIN–ACK to arrive.

TCP.047

Level: C–INFO

Short Syntax: TCP.047 State trnstn to CLOSED *source_ip_address –> destination_ip_address* dst prt *tcp_src_port* src prt *tcp_dst_port*

Long Syntax: TCP.047 State transitioned to CLOSED *source_ip_address –> destination_ip_address* dst prt *tcp_src_port* src prt *tcp_dst_port*

Description: State of tcp connection has transitioned to CLOSED.

TCP.048

Level: C–INFO

Short Syntax: TCP.048 Rcvd data after CLOSE issued and zero wndw, snd RST *source_ip_address –> destination_ip_address* dst prt *tcp_src_port* src prt *tcp_dst_port*

Long Syntax: TCP.048 Received data after CLOSE was issued, and window is zero, send RESET *source_ip_address –> destination_ip_address* dst prt *tcp_src_port* src prt *tcp_dst_port*

Description: TCP connection is CLOSING due to application requesting a CLOSE. After the window shrinks to zero, discard any packets received. This is based on the half–duplex TCP close sequence.

TCP.049

Level: C–INFO

Short Syntax: TCP.049 Rcvd NACK

Long Syntax: TCP.049 Received NACK

Description: The other side has send an old ACK with zero data length – we treat this as a NACK.

TCP.050

Level: C–INFO

Short Syntax: TCP.050 Rcvd ACK for Keep Alive

Long Syntax: TCP.050 Received Acknowledge for the keep alive packet sent

Description: The other side has acknowledged the keep alive packet. The keep alive packet is sent if keep alive is enabled on this tcp connection, and the connection has been idle.

TCP.051

Level: C–INFO

Short Syntax: TCP.051 Lcl wndw zero

Long Syntax: TCP.051 Local window zero

Description: The local window advertised is zero. The application is not draining the tcp receive buffer fast enough.

TCP.052

Level: C–INFO

Short Syntax: TCP.052 snd FIN seq *seq_num*, ack *ack_num*

Long Syntax: TCP.052 send FIN sequence number *seq_num*, acknowledge number *ack_num*

Description: The TCP connection is closing, and we sent a FIN.

TCP.053

Level: C–INFO

Short Syntax: TCP.053 get buf fld – cannot snd pkt

Long Syntax: TCP.053 get buf failed – cannot send packet

Description: The router is running out of iorbs, getbuf failed, so we cannot send a packet.

TCP.054

Level: C–INFO

Short Syntax: TCP.054 xmit buf too large (*requested_amount* clipped to *clipped_amount*

Long Syntax: TCP.054 transmit buffer too large for listen/open (*requested_amount*), clipped to (*clipped_amount*)

Description: The transmit buffer size requested for a TCP connection is too large to be allocated by the system. TCP has selected in its place the largest chunk size available in the system.

TCP.055

Level: C–INFO

Short Syntax: TCP.055 recv buf too large (*requested_amount* clipped to *clipped_amount*

Long Syntax: TCP.055 receive buffer too large for listen/open (*requested_amount*), clipped to (*clipped_amount*)

Description: The receive buffer size requested for a TCP connection is too large to be allocated by the system. TCP has selected in its place the largest chunk size available in the system.

TFTP

Trivial File Transfer Protocol

This chapter describes Trivial File Transfer Protocol (TFTP) messages. The TFTP protocol is part of the IP protocol. For information about message content and how to use the message, refer to the Introduction.

TFTP.001

Level: UI–ERROR

Short Syntax: TFTP.001 xfer max exceeded

Long Syntax: TFTP.001 simultaneous transfer maximum exceeded

Description: There is a maximum number of simultaneous TFTP transfers supported; a request (either local or remote) was made while this maximum number of TFTP transfers were already in progress.

TFTP.002

Level: UI–ERROR

Short Syntax: TFTP.002 unknwn rqst opcode: *opcode*

Long Syntax: TFTP.002 unknown TFTP request opcode: *opcode*

Description: Unknown TFTP request opcode was received.

TFTP.003

Level: UI–ERROR

Short Syntax: TFTP.003 accs viol fn: *filename_requested*

Long Syntax: TFTP.003 access violation filename: *filename_requested*

Description: A TFTP file transfer request (either local or remote) failed because of a TFTP access control violation.

TFTP.004

Level: UI–ERROR

***Short Syntax*:** TFTP.004 no UDP port avail

***Long Syntax*:** TFTP.004 no UDP port available

Description: A TFTP file transfer request (either local or remote) failed because no UDP port was available.

TFTP.005

Level: UI–ERROR

***Short Syntax*:** TFTP.005 no bfr avail

***Long Syntax*:** TFTP.005 no buffer available

Description: A TFTP request failed for lack of buffers.

TFTP.006

Level: CI–ERROR

***Short Syntax*:** TFTP.006 2nd srvr regd

***Long Syntax*:** TFTP.006 second TFTP server registered

Description: Only one TFTP server can be active at any one time; a second server has been registered by software and the previous server has been deactivate.

TFTP.007

Level: UE–ERROR

***Short Syntax*:** TFTP.007 unexp data pkt rcv

***Long Syntax*:** TFTP.007 unexpected TFTP data packet received

Description: A TFTP packet on an inactive connection was received.

TFTP.008

Level: UE–ERROR

***Short Syntax*:** TFTP.008 unexp xfer term: *reason_code*, tid *transfer_id*

***Long Syntax*:** TFTP.008 TFTP transfer unexpected termination: *reason_code*, transfer id *transfer_id*

Description: A TFTP transfer has terminated prematurely; reason code provided.

TFTP.009

Level: C–INFO

Short Syntax: TFTP.009 normal xfer cmpltd, tid *transfer_id*

Long Syntax: TFTP.009 TFTP transfer completed normally, transfer id *transfer_id*

Description: A TFTP transfer has completed normally.

TFTP.010

Level: CE–ERROR

Short Syntax: TFTP.010 sorc appren avrtd, blk *block* exp *expected_block* tid *transfer_id*

Long Syntax: TFTP.010 sorcerer's apprentice bug avoided, block *block* expected *expected_block* transfer id *transfer_id*

Description: The fix to a bug called the sorcerer's apprentice is to not retransmit old TFTP data packets in response to out–of–sequence TFTP acks; this has just occurred. The block number of the ack received and of the ack expected are displayed.

TFTP.011

Level: UE–ERROR

Short Syntax: TFTP.011 xfer timeout, tid *transfer_id*

Long Syntax: TFTP.011 TFTP transfer network timeout, transfer id *transfer_id*

Description: TFTP transfer failed due to timeout on the network.

TFTP.012

Level: U–INFO

Short Syntax: TFTP.012 ack pkt retrns, blk *block* tid *transfer_id*

Long Syntax: TFTP.012 TFTP ack packet retransmission, block number *block* transfer id *transfer_id*

Description: A TFTP ack packet was retransmitted in response to an out–of–sequence data packet received.

TFTP.013

Level: U–INFO

Short Syntax: TFTP.013 data pkt retrns, blk *block* tid *transfer_id*

Long Syntax: TFTP.013 TFTP data packet retransmission, block number *block* transfer id *transfer_id*

Description: A TFTP packet was retransmitted on expiration of a timer.

TFTP.014

Level: C–INFO
Short Syntax: TFTP.014 rmt *type* req accptd, tid *transfer_id*
Long Syntax: TFTP.014 remote TFTP *type* request accepted, transfer id *transfer_id*
Description: A remote TFTP transfer request has been accepted.

TFTP.015

Level: C–INFO
Short Syntax: TFTP.015 data pkt sent, blk *block* tid *transfer_id*
Long Syntax: TFTP.015 data packet sent, block number *block* transfer id *transfer_id*
Description: A TFTP data packet has been sent.

TFTP.016

Level: C–INFO
Short Syntax: TFTP.016 ack pkt sent, blk *block* tid *transfer_id*
Long Syntax: TFTP.016 ack packet sent, block number *block* transfer id *transfer_id*
Description: A TFTP ack packet has been sent.

TFTP.017

Level: U–INFO
Short Syntax: TFTP.017 req pkt retrns, tid *transfer_id*
Long Syntax: TFTP.017 request packet retransmitted, transfer id *transfer_id*
Description: A TFTP request packet has been retransmitted

TFTP.018

Level: UE–ERROR
Short Syntax: TFTP.018 remt req rej'd: *reasonoptional_details*
Long Syntax: TFTP.018 remote request rejected: *reasonoptional_details*
Description: A remote TFTP request was rejected for the reason shown. An optional second parameter provides further details.

TFTP.019

Level: C–INFO
Short Syntax: TFTP.019 *type* req sent, tid *transfer_id*
Long Syntax: TFTP.019 locally originated *type* request sent, transfer id *transfer_id*
Description: A locally originated TFTP request has been sent.

TFTP.020

Level: C–INFO

***Short Syntax*:** TFTP.020 xfer abrted by usr

***Long Syntax*:** TFTP.020 locally originated TFTP transfer aborted at the console

Description: Locally originated TFTP transfer was aborted at the console.

TFTP.021

Level: C–INFO

***Short Syntax*:** TFTP.021 ack pkt rcvd blk *block* tid *trans_id*

***Long Syntax*:** TFTP.021 ack packet received, block *block* transfer id *trans_id*

Description: A TFTP ack packet has been received.

TFTP.022

Level: C–INFO

***Short Syntax*:** TFTP.022 data pkt rcvd blk *block* tid *trans_id*

***Long Syntax*:** TFTP.022 data packet received, block *block* transfer id *trans_id*

Description: A TFTP data packet has been received.

TFTP.023

Level: C–INFO

***Short Syntax*:** TFTP.023 unexp err pkt rcvd code *errcodecolon_and_openquoteerrmsgclosequote*

***Long Syntax*:** TFTP.023 unexpected error packet received, code *errcodecolon_and_openquoteerrmsgclosequote*

Description: A unexpected TFTP error packet has been received.

TFTP.024

Level: UE–ERROR

***Short Syntax*:** TFTP.024 lcl dev err *errmsg*

***Long Syntax*:** TFTP.024 local device error, *errmsg*

Description: Error accessing one of the local device. Errmsg describes the actual device and the type of error.

TKR

Token-ring

This chapter describes IEEE 802.5 Token-ring network interface messages. For information about message content and how to use the message, refer to the Introduction.

TKR.001

Level: U–INFO

Short Syntax: TKR.001 unexp type frm *LLC_control* fm *source_MAC* ssap *source_SAP* dsap *dest_SAP* nt *network ID*

Long Syntax: TKR.001 Unexpected type frame *LLC_control* from *source_MAC*, ssap *source_SAP*, dsap *dest_SAP*, net *network ID*

Description: This message is generated when an unexpected 802.2 LLC frame type is received. Type may be I (information transfer) or S (supervisory). The frame was addressed to the router.

Cause: Host attempting to make 802.2 type 2 connection to router.

TKR.002

Level: P–TRACE

Short Syntax: TKR.002 unexp *type* brd frm *LLC_control* fm *source_MAC* ssap *source_SAP* dsap *dest_SAP* nt *network ID*

Long Syntax: TKR.002 Unexpected *type* broadcast frame *LLC_control* from *source_MAC*, ssap *source_SAP*, dsap *dest_SAP*, net *network ID*

Description: This message is generated when an unexpected 802.2 LLC frame type is received. Type may be I (information transfer) or S (supervisory). The frame was a broadcast.

Cause: Host attempting to make 802.2 type 2 connection to router.

TKR.003

Level: U–INFO

***Short Syntax*:** TKR.003 unkn SNAP mfr cd *number* fm *source_MAC* nt *network ID*

***Long Syntax*:** TKR.003 Unknown SNAP manufacturer code *number* from *source_MAC* net *network ID*

Description: This message is generated when a frame with an unknown organization code (not 000000) in the SNAP header is received. The frame was addressed to the router.

Cause: Host sending packets for unknown proprietary protocol using SNAP.

TKR.004

Level: P–TRACE

***Short Syntax*:** TKR.004 unkn SNAP mfr code *number* fm *source_MAC* nt *network ID*

***Long Syntax*:** TKR.004 Unknown SNAP manufacturer code *number* from *source_MAC* net *network ID*

Description: This message is generated when a frame with an unknown organization code (not 000000) in the SNAP header is received. The frame was a broadcast.

Cause: Host sending packets for unknown proprietary protocol using SNAP.

TKR.005

Level: U–INFO

***Short Syntax*:** TKR.005 unkn SNAP type *type_code* fm *source_MAC* nt *network ID*

***Long Syntax*:** TKR.005 Unknown SNAP type *type_code* from *source_MAC* net *network ID*

Description: This message is generated when a frame with an unknown SNAP type (within organization code 000000) is received. The frame was addressed to the router.

Cause: Host sending packets for unknown Ethernet type using SNAP.

TKR.006

Level: P–TRACE

***Short Syntax*:** TKR.006 unkn SNAP type *type_code* fm *source_MAC* nt *network ID*

***Long Syntax*:** TKR.006 Unknown SNAP type *type_code* from *source_MAC* net *network ID*

Description: This message is generated when a frame with an unknown SNAP type (within organization code 000000) is received. The frame was a broadcast.

Cause: Host sending packets for unknown Ethernet type using SNAP.

TKR.007

Level: U–INFO

Short Syntax: TKR.007 unkn SAP *sap_number* fm *source_MAC* nt *network ID*

Long Syntax: TKR.007 Unknown SAP *sap_number* from *source_MAC* net *network ID*

Description: This message is generated when a frame with an unknown destination SAP is received. The message was addressed to the router.

Cause: Host sending packets for unknown protocol identifier (SAP).

TKR.008

Level: U–INFO

Short Syntax: TKR.008 unkn SAP *sap_number* fm *source_MAC* nt *network ID*

Long Syntax: TKR.008 Unknown SAP *sap_number* from *source_MAC* net *network ID*

Description: This message is generated when a frame with an unknown destination SAP is received. The message was a broadcast.

Cause: Host sending packets for unknown protocol identifier (SAP).

TKR.009

Level: U–INFO

Short Syntax: TKR.009 unexp U frm *LLC_control* fm *source_MAC* ssap *source_SAP* dsap *dest_SAP* nt *network ID*

Long Syntax: TKR.009 Unexpected U frame *LLC_control* from *source_MAC*, ssap *source_SAP*, dsap *dest_SAP*, net *network ID*

Description: This message is generated when an unexpected 802.2 LLC U (unnumbered) frame type is received. (Only UI, XID, and TEST are supported.)The frame was addressed to the router.

TKR.010

Level: P–TRACE

Short Syntax: TKR.010 unexp U frm *LLC_control* fm *source_MAC* ssap *source_SAP* dsap *dest_SAP* nt *network ID*

Long Syntax: TKR.010 Unexpected U frame *LLC_control* from *source_MAC*, ssap *source_SAP*, dsap *dest_SAP*, net *network ID*

Description: This message is generated when an unexpected 802.2 LLC U (unnumbered) frame type is received. (Only UI, XID, and TEST are supported.)The frame was a broadcast.

TKR.011

Level: U–TRACE

*Short Syntax***:** TKR.011 add new RIF to *MAC_address* (*RIF header*) nt *network ID*

*Long Syntax***:** TKR.011 Added new RIF to *MAC_address* (*RIF header*), net *%3* interface *network ID*/

Description: This message is generated when a new RIF is added to the 802.5 MAC address to RIF translation cache. The first 32 bits of the *RIF header* are displayed 16 bits at a time.

TKR.012

Level: C–TRACE

*Short Syntax***:** TKR.012 xtra RIF to *MAC_address* dscd nt *network ID*

*Long Syntax***:** TKR.012 Extraneous RIF to *MAC_address* discarded, net *network ID*

Description: This message is generated when additional RIF responses are received for a request which has already been satisfied.

Cause: Redundant source routes to destination.

Action: None. This is a normal event when there are source routing bridges in parallel.

TKR.013

Deleted: Message deleted at Release 8.3.

TKR.014

Level: UI–ERROR

*Short Syntax***:** TKR.014 *selftest_phase* fld *error_condition* nt *network*

*Long Syntax***:** TKR.014 *selftest_phase* failed: *error_condition*, network *network*

Description: The self-test for the 802.5 Token–Ring card has reported an error during self-test. The phases are “Initial test”, “Board reset”, “Configuration”, “Open”, “Open: Lobe media test”, “Open: Physical insertion”, “Open: Address verification”, “Open: Roll call poll”, “Open: Request parameters”, “Packet output”, and “Packet receive”. See message TKR–45 for IBM Token–Ring self-test failures.

Cause: In the “Initial test” phase, the error is “Buffer unavail”. This indicates that there is a severe packet buffer shortage in the router.

Action: Increase memory size, or decrease size of routing tables.

Cause: In the “Board reset” phase, the error can be one of: “Initial test error”, “Adaptor ROM CRC error”, “Adaptor RAM error”, “Instruction Test error”, “Context/Interrupt Test error”, “Protocol Handler Hardware Err”, or “System Interface Register Err”. Any of these indicate internal problems within the adapter chipset.

Action: Probable hardware failure of interface. Replace.

Cause: In the “Configuration” phase, the error can be one of: “Invalid init block”, “Invalid options”, “Invalid receive burst”, “Invalid transmit burst”, “Invalid DMA abort threshhold”, “Invalid SCB”, “Invalid SSB”, “DIO Parity”, “DMA timeout”, “DMA parity”, “DMA bus error”, “DMA data error”, or “Adaptor check”. These can indicate a hardware problem within the chipset, or a software problem.

Action: Probable hardware failure of interface. Replace.

Cause: In the “Open” phase, the error can be one of: “Node address error”, “List size error”, “Buffer size error”, “Expansion RAM error”, “Transmit buffer count”, or “Invalid open option”. These can indicate a hardware problem within the chipset, or a software problem.

Action: Probable hardware failure of interface. Replace.

Cause: In the “Open: Lobe media test”, “Open: Physical insertion”,”Open: Address verification”, “Open: Participation in ring poll”, and “Open: Request initialization” phases, the error can be one of: “Function failure”, “Signal loss”, “Timeout”, “Ring failure”, “Ring beaconing”, “Duplicate node Address”, “Request initialization”, “Remove received”, or “IMPL force received”. These are indications of failures in the process of the MAC algorithms for joining the ring. The problem is probably in the ring or the cabling, not the interface.

Action: Investigate network problems in 802.5 ring that the interface is attempting to connect to.

Cause: In the “Packet output” phase, the error is “Unknown”. The self-test packet that was sent by the node to itself did not have the address recognized bit set upon the completion of transmission.

Action: Investigate network problems, possible hardware problem.

Cause: In the “Packet input” phase, the error is “Unknown”. The self-test packet that was sent by the node to itself was not received within half a second.

Action: Investigate network problems, possible hardware problem.

TKR.015

Level: UI–ERROR

***Short Syntax*:** TKR.015 dwn sts cls nt *network*

***Long Syntax*:** TKR.015 Down, ring status close indication, network *network*

Description: The interface has automatically removed itself form the ring due to some serious error condition. This may be one of “Lobe wire fault”, “Auto–removal error”, or “Remove received”. The interface will attempt to join the ring again, and may come up again.

Cause: There is a hardware problem with the ring or the interface. The exact cause is not logged, but these errors are counted, and the counters in the +interface command should indicate what the problem is.

Action: Look at the interface counters. “Lobe wire fault” indicates a problem with the network. “Auto–removal error” indicates internal problems with the interface. “Remove received” indicates that a network management station has instructed this station to leave the ring.

TKR.016

Level: UI–ERROR

***Short Syntax*:** TKR.016 dwn adap chk *adapter_check_code* nt *network*

***Long Syntax*:** TKR.016 Down, adapter check *adapter_check_code*, network *network*

Description: The interface has been brought down because of an adaptor status check. The interface will not be self-tested, and will not come back up automatically. The *adapter_check_code* indicates which error occurred.

Cause: The adapter has detected a severe unrecoverable internal failure.

Action: If the problem persists, have the interface replaced.

TKR.017

Level: UI–ERROR

***Short Syntax*:** TKR.017 pkt sz *configured_size* too big for 4 Mbps, limting to *maximum_size*, nt *network*

***Long Syntax*:** TKR.017 Packet size *configured_size* too big for 4 Megabit/Second, limiting to *maximum_size*, network *network*

Description: The user has set the packet size for the 802.5 network larger than is allowed for a 4 Megabit/second network. The 8144, 11407, and 17800 byte sizes are only legal on a 16 Megabit/second network.

TKR.018

Level: UI–ERROR

Short Syntax: TKR.018 16 Mbps not supp on dev, net *network*

Long Syntax: TKR.018 16 Megabits/second speed not supported on device, network *network*

Description: The user has set the network speed to 16 Megabits/second, but the interface in the router does not have the capability to operate at the 16 Megabits/second speed. The network will be operated at the 4 Megabits/second speed.

TKR.019

Level: UE–ERROR

Short Syntax: TKR.019 runt pkt (*length*) frm *source_address*, net *network*

Long Syntax: TKR.019 runt packet (*length* bytes) from node *source_address*, network *network*

Description: A packet has been received which is too short to contain the MAC and LLC headers.

Cause: External error.

TKR.020

Level: UE–ERROR

Short Syntax: TKR.020 DN bd ln *actual_length claimed_length source_MAC_address –> destination_MAC_address* nt *network*

Long Syntax: TKR.020 DECnet packet received with a bad length actual *actual_length* claimed *claimed_length* from *source_MAC_address* to *destination_MAC_address* network *network*

Description: A DECnet packet was received with a length field that was larger than the actual length of the packet.

TKR.021

Level: P–TRACE

Short Syntax: TKR.021 LOOP rcv *source_MAC_address –> destination_MAC_address*, nt *network*

Long Syntax: TKR.021 Loopback Protocol frame received from *source_MAC_address* to *destination_MAC_address*, network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet was received.

TKR.022

Level: UE–ERROR

Short Syntax: TKR.022 LOOP odd skp *count*,*source_MAC_address –> destination_MAC_address*, nt *network*

Long Syntax: TKR.022 Loopback Protocol, odd skipCount *count* from *source_MAC_address* to *destination_MAC_address*, network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet had an odd skipCount in the packet. It will be discarded.

Cause: Programming error on remote node.

TKR.023

Level: UE–ERROR

Short Syntax: TKR.023 LOOP bd skp *count*,*source_MAC_address –> destination_MAC_address*, nt *network*

Long Syntax: TKR.023 Loopback Protocol, bad skipCount *count* from *source_MAC_address* to *destination_MAC_address*, network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet had a skipCount in the packet that points to beyond the end of the packet. It will be discarded.

Cause: Programming error on remote node.

TKR.024

Level: P–TRACE

Short Syntax: TKR.024 LOOP func *function* not forw,*source_MAC_address –> destination_MAC_address*, nt *network*

Long Syntax: TKR.024 Loopback Protocol, function *function* not Forward Data from *source_MAC_address* to *destination_MAC_address*, network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet did not have a function code of forward (2). It will be discarded.

Cause: Function code was reply (1), because we were the ultimate destination of this packet.

Action: None.

TKR.025

Level: UE–ERROR

Short Syntax: TKR.025 LOOP mc fwd dst *forward_MAC_address*, *source_MAC_address –> destination_MAC_address*, nt *network*

Long Syntax: TKR.025 Loopback Protocol, multicast forward address *forward_MAC_address* from *source_MAC_address* to *destination_MAC_address*, network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet has a forward address that is a multicast. It will be discarded.

Cause: Programming error in remote node.

TKR.026

Level: P–TRACE

Short Syntax: TKR.026 LOOP fwd *source_MAC_address –> forward_MAC_address*, nt *network*

Long Syntax: TKR.026 Loopback Protocol, forwarding from *source_MAC_address* to *forward_MAC_address*, network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet is being forwarded to the specified next hop.

TKR.027

Level: UI–ERROR

Short Syntax: TKR.027 LOOP fwd to *forward_Ethernet_address* dsc, rsn *code*, nt *network*

Long Syntax: TKR.027 Loopback protocol, forward to *forward_Ethernet_address* discarded, for reason *code*, network *network*

Description: An Ethernet Loopback Protocol (Configuration Testing Protocol) packet could not be forwarded to the specified address, for the reason specified by code.

TKR.028

Level: UI–ERROR

Short Syntax: TKR.028 rif table corruption for nt *network*

Long Syntax: TKR.028 rif related functions failed because of rif table corruption on network *network*

Description: The rif table is corrupted.

TKR.029

Level: P_TRACE

Short Syntax: TKR.029 rif entry is being removed entry *hardware_address protocol_type* nt *network*

Long Syntax: TKR.029 rif aging function is removing entry *hardware_address protocol_type* network *network*

Description: The rif entry aging function is removing an entry from the rif table.

TKR.030

Level: UI_ERROR

Short Syntax: TKR.030 MAC frm typ *mac_frametype* unex from *hardware_address* nt *network*

Long Syntax: TKR.030 MAC frame type *mac_frametype* unexpected from *hardware_address* network *network*

Description: The handler received a frame with an unexpected frame type.

TKR.031

Level: P_TRACE

Short Syntax: TKR.031 Main rcd on nt *network*

Long Syntax: TKR.031 Maintenance packet received on net *network*

Description: The handler received a maintenance packet.

TKR.032

Level: P_TRACE

Short Syntax: TKR.032 test frm *mac_address*, src sap *source_sap*, nt *network*

Long Syntax: TKR.032 test packet from *mac_address*, source sap *source_sap*, net *network*

Description: The handler received a test message.

TKR.033

Level: P_TRACE

Short Syntax: TKR.033 xid frm *mac_address*, sap *source_sap*, nt *network*

Long Syntax: TKR.033 xid packet received from *mac_address*, source sap *source_sap*, net *network*

Description: The handler received an xid message.

TKR.034

Level: UI_ERROR

Short Syntax: TKR.034 unable to allocate buffer in handler

Long Syntax: TKR.034 unable to allocate buffer in handler

Description: The handler was unable to allocate a buffer.

Cause: The free buffer pool is getting low or there was a temporary shortage of free buffers. The handler will attempt to recover, but this situation could be a sign of an eventual meltdown. If large numbers of these errors are reported, be advised that there is probably a major configuration problem.

TKR.035

Level: U–TRACE

Short Syntax: TKR.035 new RIF (*RIF*) for *MAC_address* nt *network ID*

Long Syntax: TKR.035 new RIF (*RIF*) for *MAC_address* net *network ID*

Description: This message is generated when a new RIF is added to the 802.5 MAC address to RIF translation cache.

TKR.036

Level: ALWAYS

Short Syntax: TKR.036 can’t set 2nd grp addr *MAC_address*

Long Syntax: TKR.036 can’t set 2nd group address *MAC_address*

Description: The Token–Ring hardware can only support one group address. A second address is being attempted to be installed.

TKR.037

Level: ALWAYS

Short Syntax: TKR.037 net *network ID*, Unkn SRT Cmd Completion code – *SRT_Completion*. Being restarted

Long Syntax: TKR.037 network *network ID*, Has Received an Unknown SRT Command Completion code –*SRT_Completion* . Interface being restarted

Description: The Token–Ring board has returned an unexpected SRT Completion Code. This will cause the interface to enter self-test.

TKR.038

Level: ALWAYS

Short Syntax: TKR.038 net *network ID*, Cmnd to TKR failed – invld param(s). Being restarted

Long Syntax: TKR.038 network *network ID*, Command to Token Ring Adapter failed – invalid parameter(s). Interface being restarted

Description: The Token–Ring board has returned an a illegal parameter status code indicating that one or more of theparameters passed to it were invalid. This will cause the interface to re–initialize.

TKR.039

Level: ALWAYS

Short Syntax: TKR.039 net *network ID*, Unkn TKR Cmd Completion code – *Completion_Code*. Being restarted

Long Syntax: TKR.039 network *network ID*, Unknown Command Completion code –*Completion_Code* . Interface being restarted

Description: The Token–Ring board has returned an unexpected Completion Code. This will cause the interface to re–initialize.

TKR.040

Level: ALWAYS

Short Syntax: TKR.040 net *network ID*, Invld Command Command rcvd in tm_ioctl. Being restarted

Long Syntax: TKR.040 network *network ID*,Invalid CommandCommand received by tm_ioctl from handler. Interface being restarted

Description: The tm_ioctl routine has received an invalid command from the device handler. This will cause the interface to reinitialize.

TKR.041

Level: ALWAYS

Short Syntax: TKR.041 net *network ID*, Invld Interrupt rcvd *Interrupt* from TKR adapter. Being restarted

Long Syntax: TKR.041 network *network ID*, Invalid Interrupt *Interrupt* received from the TKR adapter. Interface being restarted

Description: The interrupt service routine has received an invalid interrupt from the adapter card. This will cause the interface to re–initialize.

TKR.042

Level: ALWAYS

Short Syntax: TKR.042 net *network ID*, Invld Interrupt rcvd *Interrupt* from TKR adapter. Being restarted

Long Syntax: TKR.042 network *network ID*, Invalid Interrupt *Interrupt* received from the TKR adapter. Interface being restarted

Description: The interrupt service routine has received an invalid interrupt from the adapter card. This will cause the interface to re–initialize.

TKR.043

Level: UE–ERROR

Short Syntax: TKR.043 drop IPX pkt w/*encap_seen* encaps – using *encap_used* encaps on int *intnum*

Long Syntax: TKR.043 dropped IPX pkt with encaps *encap_seen* using *encap_used* on interface *intnum*

Description: This message is generated when an IPX packet is received with an encapsulation other than that which has been selected for this interface

Cause: Normal for networks using multiple encapsulations on a single wire.

Action: None needed.

TKR.044

Level: UE–ERROR

Short Syntax: TKR.044 odd RIF len frm *MAC_address*; pkt drpd nt *network ID*

Long Syntax: TKR.044 odd RIF length from *MAC_address*; packet dropped on net *network ID*

Description: The length byte in the *RIF header* was odd, which is illegal. The packet was dropped.

TKR.045

Level: UI–ERROR

*Short Syntax***:** TKR.045 *selftest_phase* fld *error_condition* nt *network*

*Long Syntax***:** TKR.045 *selftest_phase* failed: *error_condition*, network *network*

Description: The self-test for the IBM 802.5 Token–Ring has reported an error during self-test. This message can often serve as a useful quick primitive diagnostic tool for the Token–Ring hardware. The phases are “reset”, “load loader (part 1)”, “load loader (part 2)”, “download microcode”, “check downloaded microcode”, “Configuration”, “Read interesting pointers”, “open: lobe media test”, “open: physical insertion”, “open: address verification”, “open: participation in ring poll”, “open: request initialization”, “Set bridge params”, “Set STE wanted”, “Packet output”, “Packet receive”, “SRT Config”, “Set func/group address”, “Unknown Test”.

Cause: open: lobe media test: function failure.

Action: This is a basic cable problem. Check the cable. Check that router configuration has the correct media cable setting, that is, UTP or STP.

Cause: open: physical insertion fld ring beaconing. The Token–Ring is beaconing. This is usually due to one station having a misconfigured speed.

Action: Check that router configuration has the correct speed setting, that is, 4 Mbps or 16 Mbps. Check that all the stations in your ring are set to the same speed. Check for physical breaks in the Token–Ring.

Cause: open: address verification fld duplicate node address. The MAC address for this interface is a duplicate on the ring.

Action: Check that router configuration has the correct MAC address for this interface. Verify the other stations on your ring for a duplicate address.

Cause: Any of the “reset”, “load loader (part 1)”, “load loader (part 2)”, “download microcode”, “check downloaded microcode” phases.

Action: Probable hardware failure of interface. Replace.

Cause: In the “Configuration” phase, the error can be one of: “initial test error”, “microcode crc error”, “adapter ram error”, “instruction test error”, “context/interrupt test error”, “protocol handler hardware err”, “system interface register err”, “invalid parameter length”, “invalid options”, “invalid receive burst”, “invalid transmit burst”, “invalid dma abort threshhold”, “invalid dma test address”, “dio parity”, “dma timeout”, “dma parity”, “dma bus error”, “dma data error”, “adapter check”.

Action: These are the failures from the diagnostics run by the adapter. Probable hardware failure of interface. Replace if it persists.

Cause: In the “Open” phase, the error can be one of: “Node address error”, “List size error”, “Buffer size error”, “Expansion RAM error”, “Transmit buffer count error”, or “Invalid open option”.

Action: Probable hardware failure of interface. Replace.

Cause: The “open: lobe media test”, “open: physical insertion”, “open: address verification”, “open: participation in ring poll”, “open: request initialization” phases. The open operation has failed.

Action: These are fixable a lot of the time. The usual failures have already been described above. Check cable configuration and speed again. Investigate network or cabling problems, possible hardware problem.

Cause: Phases “Set bridge params”, “Set STE wanted”, “SRT Config”, “Set func/group address” are phases related to setting the token–ring for bridging, group address, functional addresses, etc.

Action: This is more likely to be a software problem since the Token–Ring is already up and running successfully.

Cause: Packet output fld unknown. The Token–Ring driver could not send a test packet. This is more likely to be a software problem, such as the buffers within the router are exhausted.

Action: Restart router if it persists.

Cause: Packet receive fld unknown. The Token–Ring driver was unable to send a test packet around the ring and receive it.

Action: Check for an unusually large amount of traffic on the ring.

Panic “tkrMacTooManyReg”

*Short Syntax***:** tkr_regMacAddrUpCall: too many registered

Description: Internal problem.

Cause: Software bug.

Action: Inform customer service.

Panic “tkrMacStsTooManyReg”

Short Syntax: tkr_regStatusUpCall: too many registered

Description: Internal problem.

Cause: Software bug.

Action: Inform customer service.

Panic “tkrMacXmitTooManyReg”

Short Syntax: tkr_regXmitpCall: too many registered

Description: Internal problem.

Cause: Software bug.

Action: Inform customer service.

UDP

User Datagram Protocol

This chapter describes User Datagram Protocol (UDP) messages. UDP is part of the IP protocol. For information about message content and how to use the message, refer to the Introduction.

UDP.001

Level: P–TRACE

Short Syntax: UDP.001 pkt *source_ip_address –> destination_ip_address* port *port_number* no srvr

Long Syntax: UDP.001 Packet from *source_ip_address* for *destination_ip_address* port *port_number*, no server

Description: This message is generated when a packet is discarded because UDP is not installed in the router. The packet was for the broadcast address.

UDP.002

Level: U–INFO

Short Syntax: UDP.002 pkt *source_ip_address –> destination_ip_address* port *port_number* no srvr

Long Syntax: UDP.002 Packet from *source_ip_address* for *destination_ip_address* port *port_number*, no server

Description: This message is generated when a packet is discarded because UDP is not installed in the router. The packet was addressed to the router.

UDP.003

Level: UE–ERROR

Short Syntax: UDP.003 dsc pkt frm *source_ip_address* bd len *length*

Long Syntax: UDP.003 Discarded packet from *source_ip_address*, bad length *length*

Description: This message is generated when a packet is discarded because it had a UDP length greater than its IP length.

UDP.004

Level: UE–ERROR

Short Syntax: UDP.004 bd cksm clc *checksum* rcv *checksum*

Long Syntax: UDP.004 Bad checksum – calculated *checksum*, received *checksum*

Description: This message is generated when a packet is discarded because it had a bad checksum.

V25B

V.25*bis* Network Interface

This chapter describes V.25*bis* Network Interface messages. These messages cover the establishment and tear–down of physical layer switched circuits via a V.25*bis* compliant modem. For information about message content and how to use the message, refer to the Introduction.

V25B.001

Level: CE–ERROR

Short Syntax: V25B.001 I_ERR (0x*status*) len(*msglen*) on rcv nt *network ID*

Long Syntax: V25B.001 Frame received with I_ERR set (status = 0x*status*) or bad length(*msglen*), on network *network ID*

Description: V.25*bis*: v25b_rx() received a buffer from the driver with the error flag set or with a length less than the minimum.

Action: Report this event to customer service.

V25B.002

Level: UE–ERROR

Short Syntax: V25B.002 Rx bad type (*type*) st *state* on nt *network ID*

Long Syntax: V25B.002 Received an unrecognized frame type (*type*) in state *state*, on network *network ID*

Description: V.25*bis*: v25b_rx() received a frame from the DCE other than a normal V.25*bis* indication in a state other than “connected”.

Action: Report this event to customer service.

V25B.003

Level: U–INFO

Short Syntax: V25B.003 Cll to "*address*" failed T = *secs.ms* secs on nt *network ID*

Long Syntax: V25B.003 Call to "*address*" failed after *secs.ms* seconds on network *network ID*

Description: A connection attempt failed. Ref V25B.016 for posssible reasons.

V25B.004

Level: UE–ERROR

Short Syntax: V25B.004 Board Down DCT flags in (0x*idctst*) out (0x*odctst*) nt *network ID*

Long Syntax: V25B.004 INIDEV of the serial interface card failed, DCT flags for input and output are 0x*idctst* and 0x*odctst* respectively for network *network ID*.

Description: The serial card isn't responding to driver initialization attempts.

Action: Test the network interface: if this does not correct the problem, restarting the router may be necessary. As a last resort, consider replacing the card. This error should be reported to customer service.

V25B.005

Level: UE–ERROR

Short Syntax: V25B.005 Unexpected state (*state1*) instead of *state2* nt *network ID*

Long Syntax: V25B.005 V25B handler state (*state1*) is different from that expected (*state2*) for internal event on network *network ID*.

Description: An event occurred in a state which is inconsistent with the design of the FSM.

Action: Report this event to customer service.

V25B.006

Level: C–INFO

Short Syntax: V25B.006 FSM st *state1* ev event –> st *state2* nt *network ID*

Long Syntax: V25B.006 FSM transition occurred: old state *state1*, event event, new state *state2* on network *network ID*.

Description: The handler received an event which triggered a state change. If this occurred as a result of a modem signal change, the preceding log message (if enabled) should indicate the new signals.

V25B.007

Level: C–INFO

Short Syntax: V25B.007 Mdm Chg 0x*modem1* –> 0x*modem2* (DSR/CTS/CD/CI) nt *network ID*

Long Syntax: V25B.007 A modem signal change was detected (0x*modem1* –> 0x*modem2* DSR/CTS/CD/CI) network *network ID*.

Description: A change in the modem signals from the DCE was detected; this may or may not precipitate an FSM transition (follows).

V25B.008

Level: UE–ERROR

Short Syntax: V25B.008 Dead DCE nt *network ID*

Long Syntax: V25B.008 DCE not responding to the handler on network *network ID.*

Description: The V.25*bis* handler attempts to raise the modem (or CU/DSU) on self-test. If it doesn't respond (by raising CTS), the handler assumes it is dead or non–compliant.

Cause: DCE not connected, powered–off, inoperable, or non–V.25*bis* compliant.

Action: Attach the cable, turn it on, fix it, or get a compliant one.

V25B.009

Level: P–TRACE

Short Syntax: V25B.009 RxD Pkt ln *msglen* nt *network ID*

Long Syntax: V25B.009 Received a frame of length (*msglen*) from network *network ID.*

Description: The V.25*bis* handler received a data frame, which it is forwarding to its client encapsulator.

V25B.010

Level: P–TRACE

Short Syntax: V25B.010 TxD Pkt ln *msglen* nt *network ID*

Long Syntax: V25B.010 Transmitted a frame of length (*msglen*) over network *network ID.*

Description: The V.25*bis* handler has transmitted a data frame on behalf of its client encapsulator.

V25B.011

Level: UE–ERROR

Short Syntax: V25B.011 Unsup Fn I/F (*function*) nt *network ID*

Long Syntax: V25B.011 The (*function*) handler/forwarder interface function is not supported by the V.25*bis* handler on network *network ID*.

Description: V.25*bis* only handles the V.25*bis* call setup on behalf of an encapsulator, so some of the normal handler functions aren't applicable: "forwarder protocol initialization", "forwarder data transmit", etc.

V25B.012

Level: UE–ERROR

Short Syntax: V25B.012 No heap on *function* nt *network ID*

Long Syntax: V25B.012 Insufficient heap memory to support this function (*function*) on network *network ID*.

Description: The V.25*bis* handler requires a certain amount of heap memory to operate, and it couldn't get it.

Cause: Either the load image, or the protocol tables are too large.

Action: Get a smaller load image, or reduce the size of the forwarder tables.

V25B.013

Level: UE–ERROR

Short Syntax: V25B.013 Bd cfg (*function*) nt *network ID*

Long Syntax: V25B.013 Incomplete configuration (*function*) for network *network ID*.

Description: The V.25*bis* handler requires a minimal configuration to work, and that information was not specified.

Action: Verify that the V25B configuration for this interface includes at least the Local Address.

V25B.014

Level: UE–ERROR

Short Syntax: V25B.014 Bd *ConnID* (0x*ConnID*)

Long Syntax: V25B.014 V.25*bis* function invoked with an invalid Connection Identifier (0x*ConnID*).

Description: The V.25*bis* handler interfaces to the encapsulators via a Connection Identifier for its connection–related functions. It has been invoked with an invalid Connection Identifier.

V25B.015

Level: U–TRACE

Short Syntax: V25B.015 Drp RxD Pkt ln *msglen* st *state* nt *network ID*

Long Syntax: V25B.015 Dropping a received Data frame of length (*msglen*) in state *state* from network *network ID*.

Description: The V.25*bis* handler received a data frame, in a state where it doesn't expect one, so it dropped it.

V25B.016

Level: U–TRACE

Short Syntax: V25B.016 *indtype* Ind rsn "*reason*" st *state* nt *network ID*

Long Syntax: V25B.016 DCE indication *indtype*, reason "*reason*" in state *state* on network *network ID*.

Description: The DCE has sent the specified indication. This may indicate that a connect attempt, initiated by the V.25*bis* handler has failed (INV or CFI) for the reason specified (see the calling unit user's manual for a description of the reason code, if any accompanies this message). Alternatively, this may just be a redundant incoming call indication (INC), which had already been signalled by the CI Circuit 125.

Cause: Call aborted: router timed out, or modem user interface command.

Action: Extend the call establishment period or don't interrupt the call.

Cause: Local DCE Busy: the user interfered through the calling unit user interface.

Action: Do not interfere.

Cause: Engaged Tone: the remote end is busy.

Action: Try again later (the router should automatically).

Cause: No Dial tone: the telephone network isn't responding.

Action: Fix the link, contact service provider.

Cause: Number not stored.

Action: Call customer service: we don't use the corresponding command.

Cause: No Answer Tone detected: remote unit did not respond with answer tone.

Action: Check called number, verify that remote unit is on–line.

Cause: Ring Tone (but no answer).

Action: Check called number, verify that remote unit is on–line.

V25B.017

Level: C–INFO

***Short Syntax*:** V25B.017 Indctn "*Message*" st *state* nt *network ID*

***Long Syntax*:** V25B.017 DCE sent "*Message*" in state *state*, on network *network ID*.

Description: The calling unit has either accepted the router's request (INC), or is connecting the call (CNX). This is a normal event – albeit perhaps not always reported by a given DCE/CU.

V25B.018

Level: UE–ERROR

***Short Syntax*:** V25B.018 Dlyd Cll ind *delaytime* minutes nt *network ID*

***Long Syntax*:** V25B.018 DCE indicates Call Delayed for *delaytime* minutes on network *network ID*.

Description: The calling unit (DCE) has indicated that it will not attempt additional outgoing calls for at least the indicated period. This is an optional feature of some DCEs in some administrations, which inhibits high frequencies of calls over a short period. Examine the previous log entries to determine why so many calls are being made.

Cause: Connections to a particular destination(s) are continually being cleared.

Action: Check the GateWay messages, to determine if the calls are being IDLE–d out (increase the idle period), or if the verification procedure is failing (check the calling number at both ends).

Cause: Non–responding remote DCE.

Action: Check the called number and verify that the remote DCE is on–line.

Cause: Busy remote.

Action: Increase the Call Retries timeout for that destination.

V25B.019

Level: UE–ERROR

***Short Syntax*:** V25B.019 No Bf Cll nt *network ID*

***Long Syntax*:** V25B.019 Buffer unavailable for connection request on network *network ID*.

Description: The handler needs a buffer to send the "connection request" to the DCE, and couldn't obtain one. The call fails. The router should re–initiate the call at a later time.

V25B.020

Level: UE–ERROR

***Short Syntax*:** V25B.020 Bd Sts CRN Tx 0x*status* nt *network ID*

***Long Syntax*:** V25B.020 Bad transmit status (0x*status*) for CRN network *network ID*.

Description: The driver reports a bad transmit status when trying to send the Call Request (CRN).

V25B.021

Level: C–INFO

***Short Syntax*:** V25B.021 Set DSS DSS nt *network ID*

***Long Syntax*:** V25B.021 Set output signalsDSS on network *network ID*

Description: The router is changing its output dataset signals in response to the preceding event. (DTR = V.24 Circuit 108/2 and RTS = V.24 Circuit 105)

V25B.022

Level: CI–ERROR

***Short Syntax*:** V25B.022 no bfr avl *action* nt *network ID*

***Long Syntax*:** V25B.022 no buffer available for *action* network *network ID*

Description: A packet buffer was not available when the hardware–specific interface code required one to perform the specified action.

V25B.023

Level: U–INFO

***Short Syntax*:** V25B.023 Slftst OK nt *network ID*

***Long Syntax*:** V25B.023 Selftest completed successfully on network *network ID*

Description: self-test of the connection between the router and the modem completed ok.

V25B.024

Level: C–INFO

***Short Syntax*:** V25B.024 Tx CRN *destination* nt *network ID*

***Long Syntax*:** V25B.024 Sending Dial (CRN) command for call to *destination* on network *network ID*

Description: The modem is in a now in a state where it can actually receive V.25*bis* commands, so we are sending it the telephone number to dial.

V25B.025

Level: C–INFO

Short Syntax: V25B.025 Clnt CR *destination* nt *network ID*

Long Syntax: V25B.025 Client connection request to *destination* on network *network ID*

Description: The client (ex: Dial Circuit or WAN Restoral) has made a connection request to the specified address.

V25B.026

Level: C–INFO

Short Syntax: V25B.026 Clnt CR blckd *destination* nt *network ID*

Long Syntax: V25B.026 Client connection request on busy interface to *destination* on network *network ID*

Description: The client (ex: Dial Circuit or Wan Restoral) is trying to initiate a connection, but the base network is busy.

V25B.027

Level: C–INFO

Short Syntax: V25B.027 Out Call *destination* cmp T=*time* nt *network ID*

Long Syntax: V25B.027 Client connection established to *destination* in *time* seconds on network *%3* interface *network ID*/

Description: The connection requested by a local client (ex: Dial Circuit or Wan Restoral) to the specified address has been established in the specified time. The operator may care to use this value to adjust the configured connect timeout.

V25B.028

Level: ALWAYS

Short Syntax: V25B.028 Bad drct Tx prot *Protoco*l, pls remap to dial circuit on nt *network ID*

Long Syntax: V25B.028 Some forwarder (*Protocol*) has attempted to transmit directly over the V.25*bis* network *network ID*

Description: Transmits over the V.25*bis* network are only supposed to be done via an associated dial circuit, which will do an appropriate encapsulation. This is caused by a mistake in the configuration of the forwarders. No forwarder should be configured to use the V.25*bis* network. To bound the number of these messages, they will be logged only a fraction of the actual events.

Cause: A forwarder (IP, IPX, etc) address was assigned to the V.25*bis* interface.

Action: Delete the address, and (probably) re–assign it to a dial circuit (which is itself mapped to the V.25*bis* network).

Cause: The bridge or other forwarder has been configured to use the V.25*bis* interface.

Action: Remove the V.25*bis* interface as a port used by the bridge or forwarder.

WRS

WAN Restoral System

This chapter describes the Wide–Area Network Restoral System messages. For information about message content and how to use the message, refer to the Introduction.

WRS.001

Level: C–INFO

Short Syntax: WRS.001 Primary net *network ID* switching to secondary net *network ID*

Long Syntax: WRS.001 Primary interface number *network ID* switching to secondary interface number *%3* interface *network ID*/

Description: The primary interface is being restored through the secondary circuit.

WRS.002

Level: C–INFO

Short Syntax: WRS.002 Primary net *network ID* restored on secondary net *network ID*

Long Syntax: WRS.002 Primary interface number *network ID* restored on secondary interface number *%3* interface *network ID*/

Description: The primary interface has been restored through the secondary circuit.

WRS.003

Level: UI–ERROR

Short Syntax: WRS.003 Primary net *network ID* can't restore on secondary net *network ID*

Long Syntax: WRS.003 Primary interface number *network ID* failed to restore on secondary interface number *%3* interface *network ID*/

Description: The primary interface has not been restored through the secondary circuit.

WRS.004

Level: C–INFO

Short Syntax: WRS.004 Secondary net *network ID* switching back to primary net *network ID*

Long Syntax: WRS.004 Secondary interface number *network ID* switching back to primary interface number *%3* interface *network ID*/

Description: The secondary interface is being restored through a secondary circuit.

WRS.005

Level: C–INFO

Short Syntax: WRS.005 Switch to sec net *network ID* aborted pri net *network ID* back on line

Long Syntax: WRS.005 Switch to secondary interface number *network ID* aborted primary interface number *%3* interface *network ID*/ back on line

Description: The switch to secondary interface has been aborted, primary came back on–line.

WRS.006

Level: C–INFO

Short Syntax: WRS.006 Switch to sec net *network ID* averted pri net *network ID* disabled

Long Syntax: WRS.006 Switch to secondary interface number *network ID* averted primary interface number *%3* interface *network ID*/ disabled

Description: The switch to secondary interface has been averted, primary interface is disabled.

WRS.007

Level: C–INFO

Short Syntax: WRS.007 Secondary net *network ID* failed resort to primary net *network ID*

Long Syntax: WRS.007 Secondary interface number *network ID* resorting back to primary interface number *%3* interface *network ID*/

Description: The secondary interface has gone down causing a switch back to the primary circuit.

WRS.008

Level: C–INFO

Short Syntax: WRS.008 Sec net *network ID* swt to AVL; pri net *network ID* bck ONL

Long Syntax: WRS.008 Secondary net number *network ID* switch to AVAILABLE; primary net number *%3* interface *network ID*/ back ONLINE

Description: The switch to secondary interface has been aborted, primary still active and on–line.

WRS.009

Level: C–TRACE

Short Syntax: WRS.009 Packet forwarded pri net *network ID* onto sec net *network ID*

Long Syntax: WRS.009 Packet forwarded from the primary interface number *network ID* onto the secondary interface number *%3 network ID*/

Description: A packet has been forwarded from the primary interface onto the secondary interface.

WRS.010

Level: C–TRACE

Short Syntax: WRS.010 Packet received on pri net *network ID* from sec net *network ID*

Long Syntax: WRS.010 Packet received on primary interface number *network ID* from secondary interface number *%3 network ID*/

Description: A packet has been received onto the primary interface from the secondary interface.

WRS.011

Level: C–TRACE

Short Syntax: WRS.011 Packet discarded on pri net *network ID* sec net *network ID* down

Long Syntax: WRS.011 Packet discarded on the primary interface number *network ID* secondary interface number *%3 network ID*/ is down

Description: A packet has been discarded from the primary interface onto the secondary interface. Secondary is down.

WRS.012

Level: C–TRACE

Short Syntax: WRS.012 Unable to forward pri net *network ID* onto sec net *network ID*

Long Syntax: WRS.012 Packet forwarded from the primary interface number *network ID* onto the secondary interface number failed*%3 network ID*/

Description: A packet cannot be forwarded from the primary interface onto the secondary interface.

WRS.013

Level: C–INFO

Short Syntax: WRS.013 Switch to sec net *network ID* aborted, sec restoral disabled

Long Syntax: WRS.013 Switch to secondary interface number *network ID* aborted secondary restoral disabled

Description: The switch to secondary interface has been aborted, secondary restoral is disabled.

WRS.014

Level: C–INFO

Short Syntax: WRS.014 Switch to sec net *network ID* aborted, sec retry exceeded

Long Syntax: WRS.014 Switch to secondary interface number *network ID* aborted secondary retries exceeded

Description: The switch to secondary interface has been aborted, secondary retry attempts have been exceeded.

WRS.015

Level: C–INFO

Short Syntax: WRS.015 Secondary test initiated net *network ID*

Long Syntax: WRS.015 Secondary test initiated on secondary interface number *network ID*

Description: A secondary interface test has been initiated.

WRS.016

Level: C–INFO

Short Syntax: WRS.016 Secondary test successful net *network ID*

Long Syntax: WRS.016 Secondary test initiated on secondary interface number *network ID* has completed successfully

Description: A secondary interface test has been completed successfully.

WRS.017

Level: C–INFO

Short Syntax: WRS.017 Secondary test unsuccessful net *network ID*

Long Syntax: WRS.017 Secondary test initiated on secondary interface number *network ID* has completed unsuccessfully

Description: A secondary interface test has been completed unsuccessfully.

WRS.018

Level: C–INFO

Short Syntax: WRS.018 Periodic sec test scheduled net *network ID*

Long Syntax: WRS.018 Periodic secondary test scheduled interface number *network ID*

Description: A periodic secondary test has been scheduled on interface.

WRS.019

Level: C–INFO

Short Syntax: WRS.019 Periodic sec test passed net *network ID*

Long Syntax: WRS.019 Periodic secondary test passed interface number *network ID*

Description: A periodic secondary test has been completed successfully on interface.

WRS.020

Level: C–INFO

Short Syntax: WRS.020 Periodic sec test failed net *network ID*

Long Syntax: WRS.020 Periodic secondary test failed interface number *network ID*

Description: A periodic secondary test has not been completed successfully on interface.

WRS.021

Level: C–INFO

Short Syntax: WRS.021 Periodic sec test aborted net *network ID*

Long Syntax: WRS.021 Periodic secondary test aborted interface number *network ID*

Description: A periodic secondary test has not been completed successfully on interface.

WRS.022

Level:	UE–ERROR
Short Syntax:	WRS.022 Protocol initialization on sec ignored, prot = *type* on nt *network ID*
Long Syntax:	WRS.022 Protocol initialization on secondary ignored, protocol = *type* on network *network ID*
Description:	Invalid protocol configured on secondary circuit.
Cause:	Software configuration out of date, contact customer service. Panic ‘‘wrsimem’’
Short Syntax:	WAN restoral initialization failed, no memory.
Description:	The WAN restoral initialization failed to allocate sufficient memory to complete initialization.
Action:	Contact customer service.

XN

Xerox Network Core

This chapter describes Xerox Network Core messages. This is the core protocol software that is shared by XNS, IPX, and DDS. For information about message content and how to use the message, refer to the Introduction.

XN.001

Level: UE–ERROR

Short Syntax: XN.001 *protocol* trunc pkt frm *source_net/source_node*, xns *length* phys *length*

Long Syntax: XN.001 *protocol* truncated packet from *source_net/source_node*; xns *length*, physical *length*

Description: This message is generated when a packet has an XNS packet length greater then the packets physical length.

Cause: Programming error in remote node, truncation by network.

XN.002

Level: UE–ERROR

Short Syntax: XN.002 *protocol* non–zero TC frm *source_net/source_node* TC

Long Syntax: XN.002 *protocol* non–zero transport control from *source_net/source_node*, TC

Description: The reserved bits in the Transport Control field of the header were not zero. An Error (checksum) packet will be sent.

XN.003

Level: UE–ERROR

Short Syntax: XN.003 *protocol* bd rtng *cksum* frm *source_net/source_node*, rcv *cksum* cmp *cksum*

Long Syntax: XN.003 *protocol* bad routing checksum from *source_net/source_node*; received *cksum*, compared *cksum*

Description: This message is generated when the checksum in a packet being forwarded does not match the calculated checksum for the packet. An Error (checksum) packet will be sent.

Cause: There is a programming error in the remote node.

Action: Correct the software in remote node.

Cause: Packet was corrupted on the network.

XN.004

Level: UE–ERROR

Short Syntax: XN.004 *protocol* hop cnt ovflo frm *source_net/source_node* to *destination_net*

Long Syntax: XN.004 *protocol* hop count overflow from *source_net/source_node* to *destination_net*

Description: This message is generated when a packet's hop count counts up past 15 and overflows. An Error (hop count) packet will be sent.

XN.005

Level: CE–ERROR

Short Syntax: XN.005 *protocol* no gwy frm *source_net/source_node* to *destination_net*

Long Syntax: XN.005 *protocol* no gateway from *source_net/source_node* to *destination_net*

Description: This message is generated when a packet cannot be forwarded because there is no gateway to the destination network. An Error (unreachable) packet will be sent.

XN.006

Level: CE–ERROR

Short Syntax: XN.006 *protocol* wstd hop frm *source_net/source_node* to *destination_net*

Long Syntax: XN.006 *protocol* wasted hop from *source_net/source_node* to *destination_net*

Description: This packet is generated when a packet is being sent out the same network interface it arrived on. This router is not the best path off that network to the destination network.

Cause: Misconfigured first–hop router for end node on network.

Action: Reconfigure node.

Cause: Routing tables are inconsistent.

XN.007

Level: UE–ERROR

Short Syntax: XN.007 *protocol* dst hst 0 frm *source_net/source_node* to *destination_net*

Long Syntax: XN.007 *protocol* destination host 0 from *source_net/source_node* to *destination_net*

Description: This message is generated when a packet is addressed to node 000000000000. This is an illegal host address. An Error (checksum) packet will be sent.

XN.008

Level: P–TRACE

Short Syntax: XN.008 *protocol source_net/source_node –> dest_net/dest_node*

Long Syntax: XN.008 *protocol* Packet received from *source_net/source_node* for *dest_net/dest_node*

Description: This message is generated when a packet is forwarded.

XN.009

Level: UE–ERROR

Short Syntax: XN.009 *protocol* pkt too lng to frwd *pkt_size* > *max_size* nt output *network ID* frm *source_net/source_node*

Long Syntax: XN.009 *protocol* packet too long to forward *pkt_size* > *max_size* net output *network ID* fromsource_net source_node/

Description: This message is generated when a forwarded packet cannot be sent out the required interface because it is too long. An Error (size) packet will be sent.

XN.010

Level: UI–ERROR

Short Syntax: XN.010 *protocol* type frm *source_net/source_node* for *dest_net/dest_node* dsc, rsn *code*

Long Syntax: XN.010 *protocol* type from *source_net/source_node* for *dest_net/dest_node* discarded for reason *code*

Description: An outgoing packet was not successfully transmitted for the reason indicated by the error code.

XN.011

Level: C–INFO

Short Syntax: XN.011 *protocol* intrfc network/node nt *network ID* up

Long Syntax: XN.011 *protocol* interface *network*/node net *network ID* up

Description: The specified interface has come up, and has been enabled for the specified XNS protocol.

XN.012

Level: U–INFO

Short Syntax: XN.012 *protocol* del nt *destination_net* rt via *gateway* nt *network ID*

Long Syntax: XN.012 *protocol* deleted net *destination_net* route via *gateway* net *network ID*

Description: The specified route has been deleted because the first hop interface for that route has gone down.

Cause: Interface down.

Action: Fix network.

XN.013

Level: UI–ERROR

Short Syntax: XN.013 *protocol* tbl ovrfl, dst *destination_net*

Long Syntax: XN.013 *protocol* Table overflow, destination *destination_net*

Description: This message is generated when a new entry cannot be made to routing table because it is already full.

Cause: Routing table too small.

Action: Increase routing table size for this protocol.

XN.014

Level: C–INFO

Short Syntax: XN.014 *protocol* echo typ *operation* to skt *socket* frm *source_net/source_node*

Long Syntax: XN.014 *protocol* Echo type *operation* to socket *socket* from *source_net/source_node*

Description: A packet of the echo type has been received with the specified operation to the specified socket.

XN.015

Level: UE–ERROR

Short Syntax: XN.015 *protocol* bd src *source_net/source_node* nt *network ID*

Long Syntax: XN.015 *protocol* bad source *source_net/source_node* net *network ID*

Description: A packet was being returned to the sender, but the senders node address was a multicast address or the illegal address 000000000000. This can happen when sending an Echo reply, an Error packet, or replying to other queries. The packet will be discarded.

XN.016

Level: UE–ERROR

Short Syntax: XN.016 *protocol* bad net 0 *source_net/source_node– >dest_net/dest_node*

Long Syntax: XN.016 *protocol* bad source network 0 from *source_net/source_node* for *dest_net/dest_node*

Description: A packet was being returned to the sender, and the source network was zero, but the destination network was non–zero. This can happen when sending an Echo reply, an Error packet, or replying to other queries. The packet will be discarded.

XN.017

Level: UI–ERROR

Short Syntax: XN.017 *protocol* no mem for err pkt

Long Syntax: XN.017 *protocol* No memory for error packet

Description: This message is generated when no memory is available to copy the offending packet into an Error packet. An Error packet will not be sent.

XN.018

Level: UE–ERROR

Short Syntax: XN.018 *protocol* short (*length*) pkt frm *source_net/source_node* (?) nt *network ID*

Long Syntax: XN.018 *protocol* short (*length*) packet from *source_net/source_node* (?) net *network ID*

Description: This message is generated when a packet has a physical length shorter than the minimum 30 byte XNS, IPX, or DDS header length. The source_net and source_node may or may not be valid packet data, depending on how severe the truncation is.

Cause: Programming error in remote node, truncation by network.

XN.019

Level: C–TRACE

Short Syntax: XN.019 *protocol* chg src net to new_source_net, pkt *source_net/source_node –> dest_net/dest_node*

Long Syntax: XN.019 *protocol* changing source network to new_source_net on packet received from *source_net/source_node* for *dest_net/dest_node*

Description: This message is generated when an IPX packet is received with a source network number of 0. The router corrects this to be the network number of the interface the packet was received on.

XN.020

Level: UE–ERROR

Short Syntax: XN.020 *protocol* bad src net 0, hop count *hop_count*, *source_net/source_node –> dest_net/dest_node*, nt *network ID*

Long Syntax: XN.020 *protocol* bas source network 0 with hop count *hop_count* on packet received from *source_net/source_node* for *dest_net/dest_node* via network *network ID*

Description: This message is generated when an IPX packet is received with a source network number of 0, and the hop count (transport control) is non–zero. The source network number will not be corrected, since it is probably not the network it was received on. The dest_node will be unable to reply.

Cause: This would indicate that the packet has already been forwarded by another router that does not correct the source network number when forwarding, or that the originating node sent the packet with a non–zero hop count.

Action: Correct programming error at remote node or router.

XN.021

Level: UE–ERROR

Short Syntax: XN.021 *protocol* inv len (*claimed_length*) frm *source_net/source_node*

Long Syntax: XN.021 *protocol* invalid length (*claimed_length* bytes) from *source_net/source_node*

Description: This message is generated when a packet has a length field in the XNS, IPX or DDS network layer header that is shorter than the 30 byte minimum packet length.

Cause: Programming error in remote node, corruption by network.

X25

X.25 Network Interface

This chapter describes X.25 Network Interface messages. These messages cover the convergence layer between the Network Layer protocols and the X.25 protocol. For information about message content and how to use the message, refer to the Introduction.

X25.001

Level: UI–ERROR

Short Syntax: X25.001 fld bff allc nt *network ID*

Long Syntax: X25.001 buffer allocation failed network index *network ID*

Description: An attempt by the X.25 network handler to allocate an internal buffer failed. The effect may not be serious, unless subsequent attempts also fail.

X25.002

Level: CE–ERROR

Short Syntax: X25.002 fld qury stat nt *network ID*

Long Syntax: X25.002 statistics query failed network index *network ID*

Description: An attempt by the X.25 network handler to query X.25 statistics from the COM–4 was unsuccessful. Typically, lack of COM–4 resources was the cause, however, is not serious.

X25.003

Level: UI–ERROR

Short Syntax: X25.003 req unkn nt *network ID*

Long Syntax: X25.003 request unknown network index *network ID*

Description: The X.25 network handler received an unknown request either via the console interface or due to a forwarder problem. The request is simply ignored.

X25.004

Level: CI–ERROR

***Short Syntax*:** X25.004 xmt ovfl dst –> *x25_destination* nt *network ID*

***Long Syntax*:** X25.004 overflow on transmit to destination –> *x25_destination* network *network ID*

Description: A forward request to the X.25 network handler resulted in an queued buffer overflow towards the network.

Action: Increase the amount of packets allowed to be queued towards the network per protocol.

X25.005

Level: CI–ERROR

***Short Syntax*:** X25.005 clls exd dst –> *x25_destination* nt *network ID*

***Long Syntax*:** X25.005 maximum calls exceeded to destination –> *x25_destination* network index *network ID*

Description: The X.25 network handler failed to open a new circuit due to exceeding maximum number of circuits per a given protocol on the interface. The effect is typical given a high bursty volume of traffic on a single interface.

Action: If condition persists, contact customer service.

X25.006

Level: UE–ERROR

***Short Syntax*:** X25.006 xmt int dwn dst –> *x25_destination* net *network ID*

***Long Syntax*:** X25.006 transmit interface is down to destination –> *x25_destination* network index *network ID*

Description: An attempt by the X.25 network handler to forward a data packet failed due to X.25 protocol being disabled. This event is only possible after the network interface had been up and then moved to the initialization state.

X25.007

Level: UE–ERROR

***Short Syntax*:** X25.007 vc frq rsts src –> *x25_source* nt *network ID*

***Long Syntax*:** X25.007 virtual circuit frequent resets source –> *x25_source* network index *network ID*

Description: The X.25 network handler is experiencing a large number of circuit resets via the network interface. This is typically the result of network instability.

Action: Consult network administrator.

X25.008

Level: UI–ERROR

Short Syntax: X25.008 prtcl unkn nt *network ID*

Long Syntax: X25.008 protocol unknown network index *network ID*

Description: The X.25 network handler received a circuit open request which was associated with a non–supported protocol.

X25.009

Level: UI–ERROR

Short Syntax: X25.009 pkt lyr dwn drng init nt *network ID*

Long Syntax: X25.009 packet layer remains down during initialization network index *network ID*

Description: The X.25 network handler cannot continue initialization due to the packet layer not yet connecting with the network.

X25.010

Level: UI–ERROR

Short Syntax: X25.010 frm lyr dwn drng init nt *network ID*

Long Syntax: X25.010 frame layer remains down during initialization network index *network ID*

Description: The X.25 network handler cannot continue initialization due to the frame layer not yet establishing the link.

X25.011

Level: UI–ERROR

Short Syntax: X25.011 phy lyr dwn drng init nt *network ID*

Long Syntax: X25.011 physical layer remains down during initialization network index *network ID*

Description: The X.25 network handler cannot continue initialization due to the physical layer not yet receiving proper signaling.

X25.012

Level: CI–ERROR

Short Syntax: X25.012 no nde addr nt *network ID*

Long Syntax: X25.012 node address not assigned network index *network ID*

Description: The X.25 network handler cannot continue initialization due to lack of X.25 node address assignment.

X25.013

Level: UI–ERROR

Short Syntax: X25.013 fwd not supprtd nt *network ID*

Long Syntax: X25.013 forwarder protocol not supported network index *network ID*

Description: The X.25 network handler received a forward request from an unsupported protocol.

X25.014

Level: CI–ERROR

Short Syntax: X25.014 prtcl not cnfg nt *network ID*

Long Syntax: X25.014 protocol forwarder not configured network index *network ID*

Description: The X.25 network handler received a protocol preinitialization which resulted in using default configuration. The protocol has not been configured.

X25.015

Level: UI–ERROR

Short Syntax: X25.015 fld vc mgr init nt *network ID*

Long Syntax: X25.015 circuit manager initialization failed network index *network ID*

Description: The X.25 network handler circuit manager failed to initialize. This should not happen.

Action: Contact customer service.

X25.016

Level: UI–ERROR

Short Syntax: X25.016 vc svr err rsp nt *network ID*

Long Syntax: X25.016 circuit manager server responded in error network index *network ID*

Description: The X.25 network handler circuit manager server issued an undefined response. This event indicates internal corruption of the database.

Action: Contact customer service.

X25.017

Level: UI–ERROR

Short Syntax: X25.017 dev int dwn drng init nt *network ID*

Long Syntax: X25.017 device driver constantly down during initialization network index *network ID*

Description: The X.25 network handler is waiting on the device driver to complete the CPU to COM–4 initialization sequence.

Action: If the situation persists, reset the COM–4 interface. Contact customer service.

X25.018

Level: UI–ERROR

Short Syntax: X25.018 xmt fld nt *network ID*

Long Syntax: X25.018 transmit towards network failed network index *network ID*

Description: An attempt by the X.25 network handler to transmit towards the network failed. Either a local CPU to COM–4 problem persists or COM–4 interface is hung.

Action: If the situation persists, reset the COM–4 interface. Contact customer service.

X25.019

Level: UI–ERROR

Short Syntax: X25.019 corpt intf cmnd nt *network ID*

Long Syntax: X25.019 corrupt network interface command network index *network ID*

Description: The X.25 network handler received a corrupt command or response from the COM–4 firmware.

Action: If the situation persists, reset the COM–4 interface. Contact customer service.

X25.020

Level: UI–ERROR

Short Syntax: X25.020 invld lcn nt *network ID*

Long Syntax: X25.020 invalid logicl channel index *network ID*

Description: The X.25 network handler detected an uninitialized logical channel.

Action: If the situation persists, reset the COM–4 interface. Contact customer service.

X25.021

Level: C–INFO

***Short Syntax*:** X25.021 cll rq dst –> *x25_destination* nt *network ID*

***Long Syntax*:** X25.021 circuit call requested destination –> *x25_destination* network index *network ID*

Description: The X.25 network handler placed a call to indicated destination, in response to a protocol forward request.

X25.022

Level: C–INFO

***Short Syntax*:** X25.022 cll ind src –> *x25_source* nt *network ID*

***Long Syntax*:** X25.022 circuit call indication received from source –> *x25_source* network index *network ID*

Description: The X.25 network handler received a call request indication from indicated source.

X25.023

Level: C–INFO

***Short Syntax*:** X25.023 clr cnf src –> *x25_source* cse *clearing_cause* diag *clearing_diagnostic* nt *network ID*

***Long Syntax*:** X25.023 circuit call clear confirmed from source –> *x25_source* cause *clearing_cause* diagnostic *clearing_diagnostic* network index *network ID*

Description: The X.25 network handler received a circuit clear confirmation from indicated source.

X25.024

Level: C–INFO

***Short Syntax*:** X25.024 pkt xmt dst –> *x25_destination* nt *network ID*

***Long Syntax*:** X25.024 packet transmitted destination –> *x25_destination* network index *network ID*

Description: The X.25 network handler transmitted a data packet to indicated destination.

X25.025

Level: C–INFO

***Short Syntax*:** X25.025 pkt rcv src –> *x25_source* nt *network ID*

***Long Syntax*:** X25.025 packet received from source –> *x25_source*network index *network ID*

Description: The X.25 network handler received a data packet from indicated source.

X25.026

Level: CI–ERROR

*Short Syntax***:** X25.026 net int dwn nt *network ID*

*Long Syntax***:** X25.026 network interface went down network index *network ID*

Description: The X.25 network handler detected the network interface moving to a down state. The handler will monitor for a brief period prior to notifying protocol forwarders of the situation.

X25.027

Level: UE–ERROR

*Short Syntax***:** X25.027 xmt int dwn net *network ID*

*Long Syntax***:** X25.027 transmit interface is down network index *network ID*

Description: An attempt by the X.25 network handler to forward a data packet failed due to X.25 protocol being disabled. This event is only possible after the network interface had been up and then moved to the initialization state.

X25.028

Level: C–INFO

*Short Syntax***:** X25.028 rset ind src –> *x25_source* cse *reset_cause* diag *reset_diagnostic* nt *network ID*

*Long Syntax***:** X25.028 circuit reset indication received, source –> *x25_source* cause *reset_cause* diagnostic *reset_diagnostic* network index *network ID*

Description: The X.25 network handler received a circuit reset indication. The source DTE address and cause and diagnostic fields are included.

X25.029

Level: C–INFO

*Short Syntax***:** X25.029 rstrt ind dst –> *x25_source* cse *restart_cause* diag *restart_diagnostic* nt *network ID*

*Long Syntax***:** X25.029 circuit restart indication received destination is –> *x25_source* cause *restart_cause* diagnostic *restart_diagnostic* network index *network ID*

Description: The X.25 network handler received a circuit level restart indication. The destination DTE address and cause and diagnostic fields are included.

X25.030

Level: C–INFO

Short Syntax: X25.030 rcv diag *diagnostic_code* nt *network ID*

Long Syntax: X25.030 received diagnostic *diagnostic_code* network index *network ID*

Description: The X.25 network handler received a diagnostic packet. The diagnostic code field is included.

X25.031

Level: C–INFO

Short Syntax: X25.031 clr rq dst –> *x25_destination* nt *network ID*

Long Syntax: X25.031 circuit clear requested to destination –> *x25_destination* network index *network ID*

Description: The X.25 network handler initiated a clear circuit request to indicated destination, in response to expiration of a period of inactivity.

X25.032

Level: C–INFO

Short Syntax: X25.032 cll cnf src –> *x25_source* nt *network ID*

Long Syntax: X25.032 circuit call confirmed from source –> *x25_source* network index *network ID*

Description: The X.25 network handler received a call confirmation from the indicated source in response to an earlier call request.

X25.033

Level: C–INFO

Short Syntax: X25.033 clr ind src –> *x25_source* cse *clearing_cause* diag *clearing_diagnostic* nt *network ID*

Long Syntax: X25.033 circuit clear indication from source –> *x25_source* cause *clearing_cause* diagnostic *clearing_diagnostic* network index *network ID*

Description: The X.25 network handler received a cleared indication from the indicated source in response to the expiration of a period of inactivity.

X25.034

Level: C–INFO

Short Syntax: X25.034 cll acpt dst –> *x25_destination* nt *network ID*

Long Syntax: X25.034 circuit call request accepted to destination –> *x25_destination*network index *network ID*

Description: The X.25 network handler accepted a call request indication from the indicated destination.

X25.035

Level: UI–ERROR

Short Syntax: X25.035 fld cll allc nt *network ID*

Long Syntax: X25.035 call resource allocation failed network index *network ID*

Description: An attempt by the X.25 network handler to allocate an internal buffer during call setup failed. The effect may not be serious unless subsequent attempts also fail.

X25.036

Level: C–INFO

Short Syntax: X25.036 clr cnf dst –> *x25_destination* nt *network ID*

Long Syntax: X25.036 circuit call clear confirmed to destination –> *x25_destination* network index *network ID*

Description: The X.25 network handler confirmed a circuit clear request to indicated DTE destination.

X25.037

Level: C–INFO

Short Syntax: X25.037 cll ot bard dst –> *x25_destination* nt *network ID*

Long Syntax: X25.037 circuit outbound call barred to destination –> *x25_destination* network index *network ID*

Description: The X.25 network handler refused a circuit open request to the indicated DTE destination. Outbound calls are barred per interface configuration.

X25.038

Level: C–INFO

Short Syntax: X25.038 cll in bard nt *network ID*

Long Syntax: X25.038 circuit call inbound barred network index *network ID*

Description: The X.25 network handler refused an inbound circuit open request. Inbound calls are barred per interface configuration.

X25.039

Level: C–INFO

Short Syntax: X25.039 IP cnvt to DDN X25 *ip_destination –> x25_destination* nt *network ID*

Long Syntax: X25.039 Added IP protocol to X25 address translation *ip_destination –> x25_destination* to ARP cache on network index *network ID*

Description: The X.25 network handler converted IP protocol address to X.25 call address and stored to ARP cache.

X25.040

Level: CI–ERROR

Short Syntax: X25.040 max clls exd on intf nt *network ID*

Long Syntax: X25.040 maximum calls exceeded through interface network index *network ID*

Description: The X.25 network handler failed to open a new circuit due to exceeding maximum number of circuits on the interface. The effect could be typical given a high bursty volume of traffic on a single interface.

Action: If condition persists, increase maximum calls allowable on the interface.

X25.041

Level: UI–ERROR

Short Syntax: X25.041 svc call collis discd nt *network ID*

Long Syntax: X25.041 switched circuit call collision discarded on network index *network ID*

Description: The X.25 network handler refused an inbound circuit open request due to call collision.

X25.042

Level: CI–ERROR

Short Syntax: X25.042 PVC cnt > max nt *network ID*

Long Syntax: X25.042 Maximum count of PVCs exceeded network index *network ID*

Description: The X.25 network handler cannot continue initialization due to an excessive number of configured PVCs.

X25.043

Level: CI–ERROR

Short Syntax: X25.043 PVC LCN rnge nt *network ID*

Long Syntax: X25.043 PVC LCN lies outside configured PVC range: network *network ID*

Description: The X.25 network handler cannot continue initialization due to a configuration conflict: the identified PVC lies outside the configured PVC range.

X25.044

Level: CI–ERROR

Short Syntax: X25.044 LCN overlap nt *network ID*

Long Syntax: X25.044 One or more logical channel ranges overlap : network *network ID*

Description: The X.25 network handler cannot continue initialization due to a configuration conflict: the configured logical channel ranges overlap. For non–zero ranges, the following inequalities must hold: LOW–PVC X25.045

Level: CI–ERROR

Short Syntax: X25.045 pkt dflt > max nt *network ID*

Long Syntax: X25.045 Packet default size greater than maximum size: network *network ID*

Description: The X.25 network handler cannot continue initialization due to a configuration conflict: configured default packet size exceeds configured maximum packet size.

X25.046

Level: UI–ERROR

Short Syntax: X25.046 call req prot not supprtd nt *network ID*

Long Syntax: X25.046 call request protocol not supported network index *network ID*

Description: The X.25 network handler received a call request indicating an unsupported protocol.

X25.047

Level: UI–ERROR

Short Syntax: X25.047 No Hdw nt *network ID*

Long Syntax: X25.047 Missing or inappropriate hardware for network index *network ID*

Description: The hardware required to support host–based X.25 is not present in the configured slot.

X25.048

Level: UI–ERROR

Short Syntax: X25.048 Mgr ch (*channel*) fsm err st *oldstate* ev *event* –> st *newstate* nt *network ID*

Long Syntax: X25.048 Manager channel (*channel*) FSM error: in state *oldstate* received event *event*, new state *newstate* network index *network ID*

Description: The packet and the virtual circuit manager layers are (temporarily) unsynchronized, probably due to a packet layer RESTART or other unusual condition.

X25.049

Level: CI–ERROR

Short Syntax: X25.049 pkt rssmbly ovrn src *x25_source* nt *network ID*

Long Syntax: X25.049 packet received an aggregate M–sequence length exceeding the router packet size: source *x25_source* network index *network ID*

Description: The X.25 network handler was attempting to re–assemble an M–sequence, and the aggregate length exceeded the maximum packet size for the router.

X25.050

Level: UI–ERROR

Short Syntax: X25.050 cll ind prot *protocol* not supprtd nt *network ID*

Long Syntax: X25.050 call indication protocol *protoco*l not supported network index *network ID*

Description: The X.25 network handler received an incoming call indicating a protocol that has not been enabled for the interface.

Panic "x25intm"

Short Syntax: X25: net intf mismatch

Description: The X.25 data structure "net" is not X.25 related.

Action: Contact customer service.

Panic "x25iprt"

Short Syntax: X25: unsuppt prt drng init

Description: The X.25 network handler detected an unsupported protocol during initialization.

Action: Contact customer service.

Panic "x25imem"

Short Syntax: X25: mem alloc fld

Description: The X.25 network handler failed to allocate sufficient memory during the initialization phase.

Action: Contact customer service.

X251

X.25 Network Interface Physical Layer

This chapter describes X.25 Network Interface messages. These messages specifically cover Physical Layer activity for Host–based X.25 configurations. For information about message content and how to use the message, refer to the Introduction.

X251.001

Level: C–INFO

Short Syntax: X251.001 Mdm sts chg: DSR/DCD/CTS *DSR/DCD/CTS* nt *network ID*

Long Syntax: X251.001 Modem status changed DSR = *DSR* DCD = *DCD* CTS = *CTS* on network *network ID*

Description: The (input) modem control signals have changed, the present state of the input signals is as specified.

X251.002

Level: C–INFO

Short Syntax: X251.002 Tx Abt nt *network ID*

Long Syntax: X251.002 Transmit Abort command network *network ID*

Description: The upper (frame) layer has requested that all outbound frames queued for transmission be aborted.

X251.003

Level: C–INFO

Short Syntax: X251.003 Srl prt up, nt *network ID*

Long Syntax: X251.003 Serial port came up successfully, on network *network ID*

Description: x25_s2 routine liked the results of the load and init.

X251.004

Level: UI–ERROR

Short Syntax: X251.004 TxCmp Rsys Schd fll nt *network ID*

Long Syntax: X251.004 Rsys ring full on Transmit complete: network *network ID*

Description: An attempt to enqueue a transmit complete notification to the frame layer of X.25 failed, due to a full internal scheduler ring. This will result in the loss of buffers.

X251.005

Level: UI–ERROR

Short Syntax: X251.005 RxCmp Rsys Schd fll nt *network ID*

Long Syntax: X251.005 Rsys ring full on Receive complete: network *network ID*

Description: An attempt to enqueue a receive complete notification to the frame layer of X.25 failed, due to a full internal scheduler ring. This will result in the loss of buffers.

X251.006

Level: CE_ERROR

Short Syntax: X251.006 RxOvr nt *network ID*

Long Syntax: X251.006 Receiver overrun: frame too long network *network ID*

Description: A frame was received with a correct CRC, but which exceeded the (configured) maximum length.

X251.007

Level: CE_ERROR

Short Syntax: X251.007 RxErr st *status* nt *network ID*

Long Syntax: X251.007 Receiver error: Erroneous frame (driver status *status*) received on network *network ID*

Description: A frame was received in error (bad CRC, modem signals down, etc).

X251.008

Level: C–INFO

Short Syntax: X251.008 Frm Rxd nt *network ID*

Long Syntax: X251.008 Frame received from network *network ID*

Description: A good frame was received from the network.

X251.009

Level: CE–ERROR

***Short Syntax*:** X251.009 Frm Tx Flsh nt *network ID*

***Long Syntax*:** X251.009 Outbound frame flushed on network *network ID*

Description: A frame transmit was aborted due to protocol state or event.

X251.010

Level: CE–ERROR

***Short Syntax*:** X251.010 Frm Txd Fail st *status* nt *network ID*

***Long Syntax*:** X251.010 Frame transmission failed, status *status*, on network *network ID*

Description: A frame transmission to the network failed; the driver returned the specified status.

X251.011

Level: C–INFO

***Short Syntax*:** X251.011 Frm Txd nt *network ID*

***Long Syntax*:** X251.011 Frame successfully transmitted to network *network ID*

Description: A frame was successfully transmitted to the network.

X251.012

Level: CI–ERROR

***Short Syntax*:** X251.012 Cfg err nt *network ID*

***Long Syntax*:** X251.012 Configuration error on network index *network ID*

Description: The X.25 network handler cannot continue initialization due to a missing datum or conflict in the network configuration. Check the node address, Virtual Circuit ranges and PVC assignments (if any).

X251.013

Level: CE–ERROR

***Short Syntax*:** X251.013 Tx flsh cmp *network ID*

***Long Syntax*:** X251.013 Outbound buffer flush completed by driver on network *network ID*

Description: A protocol event has required that the frame layer flush all buffers queued to the driver. It does so by issuing a flush command. The driver marks the last such buffer, which yields this message.

X251.014

Level: UI–ERROR

Short Syntax: X251.014 Bad tkn vcb *vocab* cmd *cmd* fm *frm* ext *ext* buf *buf* net *network ID*

Long Syntax: X251.014 An internal message (token) with an unrecognized class (*vocab*) was received. The Command, From, Argument and Ptr entries were *cmd*, *frm*, *ext*, *buf* (respectively) on network *network ID*.

Description: The physical layer software has received an internal message which it does not recognize. This message was ignored. Please inform customer service of this event.

X251.015

Level: UI–ERROR

Short Syntax: X251.015 Bad tkn cmd *cmd* vcb *vocab* fm *frm* ext *ext* buf *buf* net *network ID*

Long Syntax: X251.015 Am internal message (token) GCOM token with an unrecognized command (cmd) was received. The Command, From, Argument and Ptr entries were (respectively): *vocab*, *frm*, *ext*, *buf* on network *network ID*.

Description: The physical layer software has received an internal message which it does not recognize. This message was ignored. Please inform customer service of this event.

X251.016

Level: U_TRACE

Short Syntax: X251.016 X25 bd slot *slot_num* PUD stat *pud_stat*

Long Syntax: X251.016 X25 board slot *slot_num* Power–On Diagnostics status *pud_stat*

Description: X25 Board Power–On Diagnostics status completed with the code shown. See Power–On Diagnostics manual for encoding.

X252

X.25 Network Interface Frame Layer

This chapter describes X.25 Network Interface messages. These messages specifically cover Frame Layer activity for Host–based X.25 configurations. For information about message content and how to use the message, refer to the Introduction.

X252.001

Level: C–INFO
Short Syntax: X252.001 frm lyr act nt *network ID*
Long Syntax: X252.001 Frame layer activated network *network ID*
Description: The frame layer has been activated.

X252.002

Level: C–INFO
Short Syntax: X252.002 frm lyr term nt *network ID*
Long Syntax: X252.002 Frame layer terminated network *network ID*
Description: The frame layer has been terminated.

X252.003

Level: C–INFO
Short Syntax: X252.003 frm lyr up nt *network ID*
Long Syntax: X252.003 Frame layer up network *network ID*
Description: The frame layer is up.

X252.004

Level: C–INFO
Short Syntax: X252.004 frm lyr dn nt *network ID*
Long Syntax: X252.004 Frame layer down network *network ID*
Description: The frame layer is down.

X252.005

Level: P–TRACE
Short Syntax: X252.005 I–frm rxd nt *network ID*
Long Syntax: X252.005 I–frame received from network *network ID*
Description: A good I–frame was received from the network.

X252.006

Level: P–TRACE
Short Syntax: X252.006 I–frm txd nt *network ID*
Long Syntax: X252.006 I–frame transmitted to network *network ID*
Description: A good I–frame was transmitted to the network.

X252.007

Level: P–TRACE
Short Syntax: X252.007 rr rxd nt *network ID*
Long Syntax: X252.007 rr received from network *network ID*
Description: A frame layer RR was received from the network.

X252.008

Level: P–TRACE
Short Syntax: X252.008 rr txd nt *network ID*
Long Syntax: X252.008 rr transmitted to network *network ID*
Description: A frame layer RR was transmitted to the network.

X252.009

Level: P–TRACE
Short Syntax: X252.009 rnr rxd nt *network ID*
Long Syntax: X252.009 rnr received from network *network ID*
Description: A frame layer RNR was received from the network.

X252.010

Level: P–TRACE
Short Syntax: X252.010 rnr txd nt *network ID*
Long Syntax: X252.010 rnr transmitted to network *network ID*
Description: A frame layer RNR was transmitted to the network.

X252.011

Level: P–TRACE
Short Syntax: X252.011 rej rxd nt *network ID*
Long Syntax: X252.011 rej received from network *network ID*
Description: A frame layer Reject was received from the network.

X252.012

Level: P–TRACE
Short Syntax: X252.012 rej txd nt *network ID*
Long Syntax: X252.012 rej transmitted to network *network ID*
Description: A frame layer Reject was transmitted to the network.

X252.013

Level: P–TRACE
Short Syntax: X252.013 sabme rxd nt *network ID*
Long Syntax: X252.013 sabme received from network *network ID*
Description: A SABME frame was received from the network.

X252.014

Level: P–TRACE
Short Syntax: X252.014 sabme txd nt *network ID*
Long Syntax: X252.014 sabme transmitted to network *network ID*
Description: A SABME frame was transmitted to the network.

X252.015

Level: P–TRACE
Short Syntax: X252.015 sabm rxd nt *network ID*
Long Syntax: X252.015 sabm received from network *network ID*
Description: A SABM frame was received from the network.

X252.016

Level: P–TRACE
Short Syntax: X252.016 sabm txd nt *network ID*
Long Syntax: X252.016 sabm transmitted to network *network ID*
Description: A SABM frame was transmitted to the network.

X252.017

Level: P–TRACE
Short Syntax: X252.017 disc rxd nt *network ID*
Long Syntax: X252.017 disc received from network *network ID*
Description: A DISC frame was received from the network.

X252.018

Level: P–TRACE
Short Syntax: X252.018 disc txd nt *network ID*
Long Syntax: X252.018 disc transmitted to network *network ID*
Description: A DISC frame was transmitted to the network.

X252.019

Level: P–TRACE
Short Syntax: X252.019 dm rxd nt *network ID*
Long Syntax: X252.019 dm received from network *network ID*
Description: A DM frame was received from the network.

X252.020

Level: P–TRACE
Short Syntax: X252.020 dm txd nt *network ID*
Long Syntax: X252.020 dm transmitted to network *network ID*
Description: A DM frame was transmitted to the network.

X252.021

Level: P–TRACE
Short Syntax: X252.021 ua rxd nt *network ID*
Long Syntax: X252.021 ua received from network *network ID*
Description: A UA frame was received from the network.

X252.022

Level: P–TRACE
Short Syntax: X252.022 ua txd nt *network ID*
Long Syntax: X252.022 ua transmitted to network *network ID*
Description: A UA frame was transmitted to the network.

X252.023

Level: UE–ERROR

Short Syntax: X252.023 frmr bd ctrl fld rxd nt *network ID*

Long Syntax: X252.023 frame reject for bad control field received from network *network ID*

Description: A frame reject indicating bad control field was received from the network.

X252.024

Level: UE–ERROR

Short Syntax: X252.024 frmr bd ctrl fld txd nt *network ID*

Long Syntax: X252.024 frame reject for bad control field transmitted to network *network ID*

Description: A frame reject indicating bad control field was sent to the network.

X252.025

Level: UE–ERROR

Short Syntax: X252.025 frmr I–frm too lng rxd nt *network ID*

Long Syntax: X252.025 frame reject for I–frame too long received from network *network ID*

Description: A frame reject indicating that an I–frame was too long was received from the network.

X252.026

Level: UE–ERROR

Short Syntax: X252.026 frmr I–frm too lng txd nt *network ID*

Long Syntax: X252.026 frame reject for I–frame too long transmitted to network *network ID*

Description: A frame reject indicating that an I–frame was too long was sent to the network.

X252.027

Level: UE–ERROR

Short Syntax: X252.027 frmr N(R) invld rxd nt *network ID*

Long Syntax: X252.027 frame reject for N(R) invalid received from network *network ID*

Description: A frame reject indicating that an invalid N(R) was received from the network.

X252.028

Level: UE–ERROR

Short Syntax: X252.028 frmr N(R) invld txd nt *network ID*

Long Syntax: X252.028 frame reject for N(R) invalid transmitted to network *network ID*

Description: A frame reject indicating that an invalid N(R) was received was sent to the network.

X252.029

Level: UE–ERROR

Short Syntax: X252.029 frmr prohib I–frm rxd nt *network ID*

Long Syntax: X252.029 frame reject for prohibited I–frame received from network *network ID*

Description: A frame reject indicating that a prohibited I–frame was received from the network.

X252.030

Level: UE–ERROR

Short Syntax: X252.030 frmr prohib I–frm txd nt *network ID*

Long Syntax: X252.030 frame reject for prohibited I–frame transmitted to network *network ID*

Description: A frame reject indicating that a prohibited I–frame was received was sent to the network.

X252.031

Level: UE–ERROR

Short Syntax: X252.031 invld frm rxd nt *network ID*

Long Syntax: X252.031 invalid frame received from network *network ID*

Description: An unrecognizable frame was received from the network.

X252.032

Level: C–INFO

Short Syntax: X252.032 t1 tmr exp nt *network ID*

Long Syntax: X252.032 T1 timer expired network *network ID*

Description: The T1 timer has expired for the indicated network.

X252.033

Level: C–INFO

Short Syntax: X252.033 t2 tmr exp nt *network ID*

Long Syntax: X252.033 T2 timer expired network *network ID*

Description: The T2 timer has expired for the indicated network.

X252.034

Level: C–INFO

Short Syntax: X252.034 n2 cnt exced nt *network ID*

Long Syntax: X252.034 N2 count exceeded network *network ID*

Description: The N2 count of transmit timeouts has been exceeded for the indicated network.

X253

X.25 Network Interface Packet Layer

This chapter describes X.25 Network Interface messages. These messages specifically cover Packet Layer activity for Host–based X.25 configurations. For information about message content and how to use the message, refer to the Introduction.

X253.001

Level: C–INFO
Short Syntax: X253.001 pkt lyr act nt *network ID*
Long Syntax: X253.001 Packet layer activated network *network ID*
Description: The packet layer has been activated.

X253.002

Level: C–INFO
Short Syntax: X253.002 pkt lyr term nt *network ID*
Long Syntax: X253.002 Packet layer terminated network *network ID*
Description: The packet layer has been terminated.

X253.003

Level: C–INFO
Short Syntax: X253.003 pkt lyr up nt *network ID*
Long Syntax: X253.003 Packet layer up network *network ID*
Description: The packet layer is up.

X253.004

Level: C–INFO
Short Syntax: X253.004 pkt lyr dn nt *network ID*
Long Syntax: X253.004 Packet layer down network *network ID*
Description: The packet layer is down.

X253.005

Level: P–TRACE
Short Syntax: X253.005 data pkt rxd lcn *lcn* nt *network ID*
Long Syntax: X253.005 Data Packet received on lcn *lcn* from network *network ID*
Description: A good Data Packet was received from the network.

X253.006

Level: P–TRACE
Short Syntax: X253.006 data pkt txd lcn *lcn* nt *network ID*
Long Syntax: X253.006 Data Packet transmitted on lcn *lcn* to network *network ID*
Description: A good Data Packet was transmitted to the network.

X253.007

Level: P–TRACE
Short Syntax: X253.007 call ind rxd lcn *lcn* nt *network ID*
Long Syntax: X253.007 Call indication received for lcn *lcn* from network *network ID*
Description: A Call Indication was received for the indicated lcn from the network.

X253.008

Level: P–TRACE
Short Syntax: X253.008 cll rq txd lcn *lcn* nt *network ID*
Long Syntax: X253.008 Call request packet transmitted for lcn *lcn* network *network ID*
Description: A Call Request was transmitted for the indicated lcn to the network.

X253.009

Level: P–TRACE
Short Syntax: X253.009 cll cnf rxd lcn *lcn* nt *network ID*
Long Syntax: X253.009 Call Confirmation Packet received for lcn *lcn* network *network ID*
Description: A call conformation for the indicated lcn was received from the network.

X253.010

Level: P–TRACE
Short Syntax: X253.010 cll acpt txd lcn *lcn* nt *network ID*
Long Syntax: X253.010 Call Accepted Packet transmitted for lcn *lcn* network *network ID*
Description: A Call Accepted for the indicated lcn was transmitted to the network.

X253.011

Level: P–TRACE
Short Syntax: X253.011 rr rxd lcn *lcn* nt *network ID*
Long Syntax: X253.011 RR Packet received for lcn *lcn* network *network ID*
Description: An RR for the indicated lcn was received from the network.

X253.012

Level: P–TRACE
Short Syntax: X253.012 rr txd lcn *lcn* nt *network ID*
Long Syntax: X253.012 RR Packet transmitted for lcn *lcn* network *network ID*
Description: An RR for the indicated lcn was transmitted to the network.

X253.013

Level: P–TRACE
Short Syntax: X253.013 rnr rxd lcn *lcn* nt *network ID*
Long Syntax: X253.013 RNR Packet received for lcn *lcn* network *network ID*
Description: An RNR for the indicated lcn was received from the network.

X253.014

Level: P–TRACE
Short Syntax: X253.014 rnr txd lcn *lcn* nt *network ID*
Long Syntax: X253.014 RNR Packet transmitted for lcn *lcn* network *network ID*
Description: An RNR for the indicated lcn was transmitted to the network.

X253.015

Level: P–TRACE
Short Syntax: X253.015 rej rxd lcn *lcn* nt *network ID*
Long Syntax: X253.015 REJ Packet received for lcn *lcn* network *network ID*
Description: A Reject packet for the indicated lcn was received from the network.

X253.016

Level: P–TRACE
Short Syntax: X253.016 rej txd lcn *lcn* nt *network ID*
Long Syntax: X253.016 Reject packet transmitted for lcn *lcn* network *network ID*
Description: A Reject packet for the indicated lcn was transmitted to the network.

X253.017

Level: P–TRACE

Short Syntax: X253.017 clr rq rxd lcn *lcn* cse *clearing_cause* diag *clearing_diagnostic* nt *network ID*

Long Syntax: X253.017 Clear request received for lcn *lcn* cause *clearing_cause* diagnostic *clearing_diagnostic* network *network ID*

Description: A clear request for the indicated lcn was received from the network.

X253.018

Level: P–TRACE

Short Syntax: X253.018 clr rq txd lcn *lcn* cse *clearing_cause* diag *clearing_diagnostic* nt *network ID*

Long Syntax: X253.018 Clear request transmitted for lcn *lcn* cause *clearing_cause* diagnostic *clearing_diagnostic* network *network ID*

Description: A clear request for the indicated lcn was transmitted to the network.

X253.019

Level: P–TRACE

Short Syntax: X253.019 clr cnf rxd lcn *lcn* nt *network ID*

Long Syntax: X253.019 Clear confirm received for lcn *lcn* network *network ID*

Description: A clear confirm for the indicated lcn was received from the network.

X253.020

Level: P–TRACE

Short Syntax: X253.020 clr cnf txd lcn *lcn* nt *network ID*

Long Syntax: X253.020 Clear confirm transmitted to lcn *lcn* network *network ID*

Description: A clear confirm for the indicated lcn was transmitted to the network.

X253.021

Level: P–TRACE

Short Syntax: X253.021 intrrpt rxd lcn *lcn* nt *network ID*

Long Syntax: X253.021 Interrupt received for lcn *lcn* network *network ID*

Description: An interrupt for the indicated lcn was received from the network.

X253.022

Level: P–TRACE

Short Syntax: X253.022 intrrpt cnf txd lcn *lcn* nt *network ID*

Long Syntax: X253.022 Interrupt confirm transmitted for lcn *lcn* network *network ID*

Description: An interrupt confirm for the indicated lcn was transmitted to the network.

X253.023

Level: P–TRACE

Short Syntax: X253.023 rset rxd lcn *lcn* cse *reset_cause* diag *reset_diagnostic* nt *network ID*

Long Syntax: X253.023 Reset received for lcn *lcn* cause *reset_cause* diagnostic *reset_diagnostic* network *network ID*

Description: A reset for the indicated lcn was received from the network.

X253.024

Level: P–TRACE

Short Syntax: X253.024 rset txd lcn *lcn* cse *reset_cause* diag *reset_diagnostic* nt *network ID*

Long Syntax: X253.024 Reset transmitted for lcn *lcn* cause *reset_cause* diagnostic *reset_diagnostic* network *network ID*

Description: A reset for the indicated lcn was transmitted to the network.

X253.025

Level: P–TRACE

Short Syntax: X253.025 rset cnf rxd lcn *lcn* nt *network ID*

Long Syntax: X253.025 Reset confirm received for lcn *lcn* network *network ID*

Description: A reset confirm for the indicated lcn was received from the network.

X253.026

Level: P–TRACE

Short Syntax: X253.026 rset cnf txd lcn *lcn* nt *network ID*

Long Syntax: X253.026 Reset confirm transmitted for lcn *lcn* network *network ID*

Description: A reset confirm for the indicated lcn was transmitted to the network.

X253.027

Level: P–TRACE

Short Syntax: X253.027 rstrt rxd cse *restart_cause* diag *restart_diagnostic* nt *network ID*

Long Syntax: X253.027 Restart received cause *restart_cause* diagnostic *restart_diagnostic* network *network ID*

Description: A restart was received from the network.

X253.028

Level: P–TRACE

Short Syntax: X253.028 rstrt txd cse *restart_cause* diag *restart_diagnostic* nt *network ID*

Long Syntax: X253.028 Restart transmitted cause *restart_cause* diagnostic *restart_diagnostic* network *network ID*

Description: A restart was transmitted to the network.

X253.029

Level: P–TRACE

Short Syntax: X253.029 rstrt cnf rxd nt *network ID*

Long Syntax: X253.029 Restart confirm received network *network ID*

Description: A restart confirm was received from the network.

X253.030

Level: P–TRACE

Short Syntax: X253.030 rstrt cnf txd nt *network ID*

Long Syntax: X253.030 Restart confirm transmitted network *network ID*

Description: A restart confirm was transmitted to the network.

X253.031

Level: P–TRACE

Short Syntax: X253.031 diag txd diag cde *diagnostic_code* nt *network ID*

Long Syntax: X253.031 Diagnostic transmitted diagnostic code *diagnostic_code* network *network ID*

Description: A diagnostic packet was transmitted to the network.

X253.032

Level: P–TRACE

***Short Syntax*:** X253.032 diag rxd diag cde *diagnostic_code* nt *network ID*

***Long Syntax*:** X253.032 Diagnostic received diagnostic code *diagnostic_code* network *network ID*

Description: A diagnostic packet was received from the network.

X253.033

Level: C–INFO

***Short Syntax*:** X253.033 rstrt tmr exp nt *network ID*

***Long Syntax*:** X253.033 Restart timer expired network *network ID*

Description: The restart timer has expired for the indicated network.

X253.034

Level: C–INFO

***Short Syntax*:** X253.034 clr tmr exp lcn *lcn* nt *network ID*

***Long Syntax*:** X253.034 Clear timer expired for lcn *lcn* network *network ID*

Description: The clear timer has expired for the indicated lcn.

X253.035

Level: C–INFO

***Short Syntax*:** X253.035 cll tmr exp lcn *lcn* nt *network ID*

***Long Syntax*:** X253.035 Call timer expired for lcn *lcn* network *network ID*

Description: The call timer has expired for the indicated lcn.

X253.036

Level: C–INFO

***Short Syntax*:** X253.036 rset tmr exp lcn *lcn* nt *network ID*

***Long Syntax*:** X253.036 Reset timer expired for lcn *lcn* network *network ID*

Description: The reset timer has expired for the indicated lcn.

X253.037

Level: UE–ERROR

***Short Syntax*:** X253.037 invld P(R) rxd lcn *lcn* nt *network ID*

***Long Syntax*:** X253.037 Invalid P(R) received lcn *lcn* network *network ID*

Description: A packet containing an invalid P(R) was received. The circuit will be reset.

X253.038

Level: UE–ERROR

Short Syntax: X253.038 invld P(S) rxd lcn *lcn* nt *network ID*

Long Syntax: X253.038 Invalid P(S) received lcn *lcn* network *network ID*

Description: A packet containing an invalid P(S) was received. The circuit will be reset, or the packet will be rejected if retransmission is supported.

X253.039

Level: CI–ERROR

Short Syntax: X253.039 no avail chn for cll nt *network ID*

Long Syntax: X253.039 No available channel for call network *network ID*

Description: A call request could not be sent because no channel number is available. If possible, increase the range of channels in the X.25 configuration that may be used for SVCs.

ZIP

AppleTalk Phase 1 Zone Information Protocol

This chapter describes AppleTalk Phase 1 Zone Information Protocol (ZIP) messages. ZIP is part of the AppleTalk Phase 1 family. For information about message content and how to use the message, refer to the Introduction.

ZIP.001

Level: U–INFO

Short Syntax: ZIP.001 del zone *zone*

Long Syntax: ZIP.001 deleting zone *zone*

Description: The indicated zone was deleted from the Zone Information Table.

ZIP.002

Level: UI–ERROR

Short Syntax: ZIP.002 no mem for new zone *zone*

Long Syntax: ZIP.002 no memory for new zone *zone*

Description: The indicated zone was not inserted into the Zone Information Table due to insufficient memory in the router.

ZIP.003

Level: UI–ERROR

Short Syntax: ZIP.003 no mem for ZIP query net *net_number*

Long Syntax: ZIP.003 no memory for ZIP query net *net_number*

Description: The router was unable to generate a zone name query for the indicated network because no memory was available for the outgoing packet.

ZIP.004

Level: UI–ERROR

Short Syntax: ZIP.004 query disc nt *network* rsn *error_code*

Long Syntax: ZIP.004 query discarded net *network* reason *error_code*

Description: A zone name query was not transmitted on the indicated net for the specified reason.

ZIP.005

Level: C–INFO

Short Syntax: ZIP.005 query for *net_num* snt to */node_num* nt *network*

Long Syntax: ZIP.005 query for *net_num* sent to */node_num* net *network*

Description: A ZIP query was sent for the indicated net to the specified router.

ZIP.006

Level: C–INFO

Short Syntax: ZIP.006 query for *net_num* brdcst nt *network*

Long Syntax: ZIP.006 query for *net_num* broadcast on net *network*

Description: A ZIP query was sent for the indicated net and was broadcast on the specified interface.

ZIP.007

Level: UE–ERROR

Short Syntax: ZIP.007 unrcgnzd ZIP typ *type* fr *src_node* nt *network*

Long Syntax: ZIP.007 unrecognized ZIP type *type* from *src_node* net *network*

Description: A ZIP packet with an unrecognized command type was encountered.

ZIP.008

Level: P–TRACE

Short Syntax: ZIP.008 rply rcvd frm *src_net/src_node* nt *network*

Long Syntax: ZIP.008 reply received from *src_net/src_node* net *network*

Description: A ZIP reply packet was received from the indicated router.

ZIP.009

Level: C–INFO

***Short Syntax*:** ZIP.009 ZIT entry, zn nm *zone* assgnd to nt *net_number*

***Long Syntax*:** ZIP.009 ZIT entry, zone name *zone* assigned to net *net_number*

Description: The specified zone name for the indicated net was added to the Zone Information Table.

ZIP.010

Level: UI–ERROR

***Short Syntax*:** ZIP.010 no buf for ZIP rply to *node_number*

***Long Syntax*:** ZIP.010 no packet buffer for ZIP reply to *node_number*

Description: No packet buffer was available for sending a ZIP reply to the specified router.

ZIP.011

Level: UI–ERROR

***Short Syntax*:** ZIP.011 rply disc nt *network* rsn *error_code*

***Long Syntax*:** ZIP.011 reply discarded net *network* reason *error_code*

Description: A ZIP reply was not sent for the indicated reason.

ZIP.012

Level: C–INFO

***Short Syntax*:** ZIP.012 rply net *net_num* in zn *zone_name* snt to /*node* nt *network*

***Long Syntax*:** ZIP.012 reply net *net_num* in zone *zone_name* sent to /*node* net *network*

Description: A ZIP reply was sent to the indicated router.

ZIP.013

Level: P–TRACE

***Short Syntax*:** ZIP.013 qry rcvd frm *src_net/src_node* nt *network*

***Long Syntax*:** ZIP.013 query received from *src_net/src_node* net *network*

Description: A ZIP query packet was received from the indicated router.

ZIP.014

Level: U–INFO

Short Syntax: ZIP.014 *command* ignrd frm *src_net/src_node* nt *network*

Long Syntax: ZIP.014 *command* ignored from *src_net/src_node* net *network*

Description: A ZIP takedown or bringup request was ignored because the configuration option to process takedowns and bringups was not enabled.

ZIP.015

Level: UE–ERROR

Short Syntax: ZIP.015 tkdwn rqst *problem* frm *src_net/src_node* nt *network*, ign

Long Syntax: ZIP.015 takedown request *problem* from *src_net/src_node* net *network*, ignored

Description: A ZIP takedown request was received that was not valid. If the problem was "bad network count", the network count was not the required value of 1. If the problem was "not broadcast", the packet was not sent as a broadcast. The request will be ignored.

ZIP.016

Level: UE–ERROR

Short Syntax: ZIP.016 brngp, not dwn frm *src_net/src_node* nt *network*

Long Syntax: ZIP.016 bringup, not down from *src_net/src_node* net *network*

Description: Either a ZIP bringup request was received but a takedown had not been previously received, or the network in question was reachable for some reason. The bringup will be ignored.

ZIP.017

Level: UE–ERROR

Short Syntax: ZIP.017 rply trunc frm *src_net/src_node* nt *network*

Long Syntax: ZIP.017 reply truncated from *src_net/src_node* net *network*

Description: A ZIP reply was received that was not long enough to contain all of the ZIP tuples. All tuples before the DDP end of the packet will be processed.

ZIP.018

Level: U–INFO

Short Syntax: ZIP.018 tkdwn rqst frm *src_net/src_node* nt *network*

Long Syntax: ZIP.018 takedown request from *src_net/src_node* net *network*

Description: A ZIP takedown request was received from the indicated host.

ZIP.019

Level: U–INFO

***Short Syntax*:** ZIP.019 rt del nt *dest_net* ZIP tkdwn

***Long Syntax*:** ZIP.019 route deleted net *dest_net* ZIP takedown

Description: The indicated destination net was deleted from the routing table as a result of a ZIP takedown request.

ZIP.020

Level: UE–ERROR

***Short Syntax*:** ZIP.020 brngp bd zn nm len *length* frm *src_net/src_node* nt *network*

***Long Syntax*:** ZIP.020 bringup bad zone name length *length* from *src_net/src_node* net *network*

Description: A ZIP bringup packet with a zone name length that was too short or long.

ZIP.021

Level: P–TRACE

***Short Syntax*:** ZIP.021 brngp frm *src_net/src_node* nt *network*

***Long Syntax*:** ZIP.021 bringup from *src_net/src_node* net *network*

Description: A ZIP bringup packet from the specified host was received.

ZIP.022

Level: UE–ERROR

***Short Syntax*:** ZIP.022 Rply err – zn nm cnflct nt *net_num* alrdy assgnd zn *zone_name*

***Long Syntax*:** ZIP.022 Rply error – zone name conflict net *net_num* already assigned zone *zone_name*

Description: A ZIP reply was received with a conflicting zone name for an existing ZIT entry.

ZIP.023

Level: UE–ERROR

***Short Syntax*:** ZIP.023 ATP shrt (*length*) frm *src_net/src_node* nt *network*

***Long Syntax*:** ZIP.023 ATP short (*length* bytes) from *src_net/src_node* net *network*

Description: An ATP packet was received that was too short to contain the ATP header. The packet will be discarded.

ZIP.024

Level: P–TRACE

Short Syntax: ZIP.024 *type* rcvd frm *src_net/src_node* nt *network*

Long Syntax: ZIP.024 *type* received from *src_net/src_node* net *network*

Description: A ZIP GetMyZone or GetZoneList ATP packet was received from the indicated host.

ZIP.025

Level: UE–ERROR

Short Syntax: ZIP.025 ATP bd hdr frm *src_net/src_node* nt *network*

Long Syntax: ZIP.025 ATP bad header from *src_net/src_node* net *network*

Description: There is a bad ATP header from the specified host. TReq not XO, or low bit of Bitmap is not set. The packet will be discarded.

ZIP.026

Level: UE–ERROR

Short Syntax: ZIP.026 ATP bd func *function* frm *src_net/src_node* nt *network*

Long Syntax: ZIP.026 ATP bd function *function* from *src_net/src_node* net *network*

Description: A ZIP ATP packet was received with a bad function code in the ATP user bytes. The packet will be discarded.

ZIP.027

Level: UE–ERROR

Short Syntax: ZIP.027 *type* too long (*length*) frm *src_net/src_node* nt *network*

Long Syntax: ZIP.027 *type* too long (*length* bytes) from *src_net/src_node* net *network*

Description: A ZIP GetMyZone or GetZoneList ATP request packet was too long.

ZIP.028

Level: UE–ERROR

Short Syntax: ZIP.028 GetZoneList strt indx 0 frm *src_net/src_node* nt *network*

Long Syntax: ZIP.028 GetZoneList start index 0 from *src_net/src_node* net *network*

Description: A ZIP GetZoneList packet was received with a start index of 0.

ZIP.029

Level: UE–ERROR

Short Syntax: ZIP.029 GetMyZone strt indx not 0 frm *src_net/src_node* nt *network*

Long Syntax: ZIP.029 GetMyZone start index not 0 from *src_net/src_node* net *network*

Description: A GetMyZone ATP packet was received where the start index was not 0. The packet will be discarded.

ZIP.030

Level: U–INFO

Short Syntax: ZIP.030 No zn nm assoc wth nt *network*

Long Syntax: ZIP.030 No zone name associated with net *network*

Description: There is no zone name associated with the indicated directly connected network.

Cause: This is a temporary condition where the router has received a ZIP GetMyZone packet before it has learned the zone name of the network for this interface.

ZIP.031

Level: UI–ERROR

Short Syntax: ZIP.031 *type*Reply disc nt *network* rsn *error_code*

Long Syntax: ZIP.031 *type*Reply discarded net *network* reason *error_code*

Description: A ZIP GetZoneListReply or GetMyZoneReply was not sent for the indicated reason.

ZIP.032

Level: UE–ERROR

Short Syntax: ZIP.032 Brngp trunc (*length*) frm *src_net/src_node* nt *network*

Long Syntax: ZIP.032 Bringup truncated (*length* bytes) from *src_net/src_node* net *network*

Description: A ZIP Bringup packet was received that was not long enough to contain the claimed zone name length.

ZIP.033

Level: UE–ERROR

Short Syntax: ZIP.033 *type* usr byt 2 not 0 frm *src_net/src_node* nt *network*

Long Syntax: ZIP.033 *type* user byte 2 not 0 from *src_net/src_node* net *network*

Description: A ZIP GetMyZone or GetZoneList ATP packet was received with user byte 2 of the ATP header, not 0 from the indicated host. The packet will be discarded.

ZIP.034

Level: UE–ERROR

Short Syntax: ZIP.034 GetZoneList st indx *index*, high frm *src_net/src_node* nt *network*

Long Syntax: ZIP.034 GetZoneList start index *index*, too high from *src_net/src_node* net *network*

Description: A ZIP GetZoneList packet was received asking for zones with indices above the one given, but none were found.

Cause: A change in the ZIT, such as a zone deletion, has caused the indices to change values since the last GetZoneList request.

Action: Try again.

Cause: The remote node has a programing error.

ZIP.035

Level: CE–ERROR

Short Syntax: ZIP.035 query cnt 0 frm *src_net/src_node* nt *network*

Long Syntax: ZIP.035 query count 0 from *src_net/src_node* net *network*

Description: A ZIP Query packet was received with a network count of 0.

ZIP.036

Level: CE–ERROR

Short Syntax: ZIP.036 rply cnt 0 frm *src_net/src_node* nt *network*

Long Syntax: ZIP.036 reply count 0 from *src_net/src_node* net *network*

Description: A ZIP Reply packet was received with a network count of 0.

ZIP.037

Level: UE–ERROR

Short Syntax: ZIP.037 short (*length*) frm *src_node* nt *network*

Long Syntax: ZIP.037 packet short (*length* bytes) from *src_node* net *network*

Description: A ZIP packet was received that was not long enough to contain the 2 byte ZIP header after the DDP header. The packet will be discarded.

ZIP.038

Level: UE–ERROR

Short Syntax: ZIP.038 cnt *network_count* & len (*length*) disag frm *src_net/src_node* nt *network*

Long Syntax: ZIP.038 Network count *network_count* and DDP length (*length* bytes) disagree from *src_net/src_node* net *network*

Description: A ZIP Query packet was received where the expected length based on the ZIP network count does not agree with the actual DDP length of the packet.

Cause: Programming error at remote node.

ZIP.039

Level: C–INFO

Short Syntax: ZIP.039 unk nt *network_number* in qry frm *src_net/src_node* nt *network*

Long Syntax: ZIP.039 Unknown network number *network_number* in Query from *src_net/src_node* net *network*

Description: A ZIP query packet was received with the specified network number in the ZIP data, but this network is not in the RTMP database, or does not have a zone name in the ZIP database. Processing of the packet will continue.

ZIP.040

Level: UE–ERROR

Short Syntax: ZIP.040 unk nt *network_number* in rply frm *src_net/src_node* nt *network*

Long Syntax: ZIP.040 Unknown network number *network_number* in Reply from *src_net/src_node* net *network*

Description: A ZIP Reply packet was received with the specified network number in the ZIP data, but this network is not in the RTMP database. Processing of the packet will continue.

ZIP.041

Level: UE–ERROR

Short Syntax: ZIP.041 brngp rqst *problem* frm *src_net/src_node* nt *network*, ign

Long Syntax: ZIP.041 takedown request *problem* from *src_net/src_node* net *network*, ignored

Description: A ZIP bringup request was received that was not valid. If the problem was “bad network count”, the network count was not the required value of 1. If the problem was “not broadcast”, the packet was not sent as a broadcast. If the problem was “net number not 0”, the network number in the ZIP tuple was not zero. The request will be ignored.

ZIP.042

Level: UE–ERROR

Short Syntax: ZIP.042 rply bd tpl nm len *length* nt *network* frm *src_net/src_node* nt *network*, ign

Long Syntax: ZIP.042 reply bad tuple name length *length* network *network* from *src_net/src_node* net *network*, ignored

Description: A ZIP reply packet was received where one of the zone names was not of a legal length (between 1 and 32 characters). Processing of the reply ends with the ZIP tuple for the noted network number.

ZIP2

AppleTalk Phase 2 Zone Information Protocol

This chapter describes AppleTalk Phase 2 Zone Information Protocol (ZIP) messages. ZIP is part of the AppleTalk Phase 2 family. For information about message content and how to use the message, refer to the Introduction.

ZIP2.001

Level: U–INFO

Short Syntax: ZIP2.001 del zone *zone*

Long Syntax: ZIP2.001 deleting zone *zone*

Description: The indicated zone was deleted from the Zone Information Table.

ZIP2.002

Level: UI–ERROR

Short Syntax: ZIP2.002 no mem for new zone *zone*

Long Syntax: ZIP2.002 no memory for new zone *zone*

Description: The indicated zone was not inserted into the Zone Information Table due to insufficient memory in the router.

ZIP2.003

Level: UI–ERROR

Short Syntax: ZIP2.003 no mem for ZIP query net *net_number*

Long Syntax: ZIP2.003 no memory for ZIP query net *net_number*

Description: The router was unable to generate a zone name query for the indicated network because no memory was available for the outgoing packet.

ZIP2.004

Level: UI–ERROR

Short Syntax: ZIP2.004 query disc nt *network* rsn *error_code*

Long Syntax: ZIP2.004 query discarded net *network* reason *error_code*

Description: A zone name query was not transmitted on the indicated net for the specified reason.

ZIP2.005

Deleted: Message deleted.

ZIP2.006

Level: C–INFO

Short Syntax: ZIP2.006 query for *net_num* brdcst nt *network*

Long Syntax: ZIP2.006 query for *net_num* broadcast on net *network*

Description: A ZIP query was sent for the indicated net was broadcast on the specified interface.

ZIP2.007

Deleted: Message deleted.

ZIP2.008

Level: P–TRACE

Short Syntax: ZIP2.008 rply rcvd frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.008 reply received from *src_net/src_node* net *network*

Description: A ZIP reply packet was received from the indicated router.

ZIP2.009

Level: C–INFO

Short Syntax: ZIP2.009 ZIT entry, zn nm *zone* assgnd to nt *net_number*

Long Syntax: ZIP2.009 ZIT entry, zone name *zone* assigned to net *net_number*

Description: The specified zone name for the indicated net was added to the Zone Information Table.

ZIP2.010

Deleted: Message deleted.

ZIP2.011

Level: UI–ERROR

Short Syntax: ZIP2.011 rply disc nt *network* rsn *error_code*

Long Syntax: ZIP2.011 reply discarded net *network* reason *error_code*

Description: A ZIP reply was not sent for the indicated reason.

ZIP2.012

Deleted: Message deleted.

ZIP2.013

Level: P–TRACE

Short Syntax: ZIP2.013 qry rcvd frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.013 query received from *src_net/src_node* net *network*

Description: A ZIP query packet was received from the indicated node.

ZIP2.014

Level: UE–ERROR

Short Syntax: ZIP2.014 Bad GtNtInf rq frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.014 Bad GetNetInfo request from *src_net/src_node* net *network*

Description: A ZIP GetNetInfo request was discarded due to either a short packet length or non–blank fields.

Cause: The remote node has a programming error.

ZIP2.015

Level: U–INFO

Short Syntax: ZIP2.015 GtNtInf rqst frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.015 GetNetInfo request from *src_net/src_node* net *network.*

Description: A ZIP GetNetInfo request was received from the indicated source.

ZIP2.016

Level: UI–ERROR

Short Syntax: ZIP2.016 no buf for ZIP GtNtInf rply to *src_net/src_node*

Long Syntax: ZIP2.016 no packet buffer for ZIP GetNetInfo reply to *src_net/src_node*.

Description: No packet buffer was available for sending a ZIP GetNetInfo reply to the specified source.

ZIP2.017

Level: UE–ERROR

Short Syntax: ZIP2.017 rply trunc frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.017 reply truncated from *src_net/src_node* net *network*

Description: A ZIP reply was received that was not long enough to contain all of the ZIP tuples. All tuples before the DDP end of the packet will be processed.

ZIP2.018

Level: UI–ERROR

Short Syntax: ZIP2.018 GtNtInf rply disc nt *network* rsn *error_code*

Long Syntax: ZIP2.018 GetNetInfo reply discarded net *network* reason *error_code*

Description: A ZIP GetNetInfo reply was not sent for the indicated reason.

ZIP2.019

Level: U–INFO

Short Syntax: ZIP2.019 GtNtInf rply for *net_range* frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.019 GetNetInfo reply for net *net_range* from *src_net/src_node* net *network*

Description: A GetNetInfo reply was received for the given net range from the indicated source over the indicated net.

ZIP2.020

Level: UE–ERROR

Short Syntax: ZIP2.020 GtNtInf rply trunc (*length*) frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.020 GetNetInfo reply truncated (*length* bytes) from *src_net/src_node* net *network*

Description: A GetNetInfo reply was received with the packet too short to hold all the information.

Cause: The remote node has a programming error.

ZIP2.021

Level: U–INFO

Short Syntax: ZIP2.021 Ntfy frm *src_net/src_node* nt *network*, ign

Long Syntax: ZIP2.021 ZIP Notify from *src_net/src_node* net *network*, ignored

Description: A ZIP Notify was received, these are currently ignored.

ZIP2.022

Level: UE–ERROR

Short Syntax: ZIP2.022 Rply err – zn nm cnflct nt *net_num* alrdy assgnd zn *zone_name*

Long Syntax: ZIP2.022 Rply error – zone name conflict net *net_num* already assigned zone *zone_name*

Description: A ZIP reply was received with a conflicting zone name for an existing ZIT entry.

ZIP2.023

Level: UE–ERROR

Short Syntax: ZIP2.023 ATP shrt (*length*) frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.023 ATP short (*length* bytes) from *src_net/src_node* net *network*

Description: An ATP packet was received that was too short to contain the ATP header. The packet will be discarded.

ZIP2.024

Level: P–TRACE

Short Syntax: ZIP2.024 type rcvd frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.024 type received from *src_net/src_node* net *network*

Description: A ZIP GetMyZone, GetZoneList, or GetLocalZones ATP packet was received from the indicated host.

ZIP2.025

Level: UE–ERROR

Short Syntax: ZIP2.025 ATP bd hdr frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.025 ATP bad header from *src_net/src_node* net *network*

Description: Bad ATP header from specified host. TReq not XO, or low bit of Bitmap not set. The packet will be discarded.

ZIP2.026

Level: UE–ERROR

Short Syntax: ZIP2.026 ATP bd func function frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.026 ATP bd function function from *src_net/src_node* net *network*

Description: A ZIP ATP packet was received with a bad function code in the ATP user bytes. The packet will be discarded.

ZIP2.027

Level: UE–ERROR

Short Syntax: ZIP2.027 *type* too long (*length*) frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.027 *type* too long (*length* bytes) from *src_net/src_node* net *network*

Description: A ZIP GetMyZone or GetZoneList ATP request packet was too long.

ZIP2.028

Level: UE–ERROR

Short Syntax: ZIP2.028 GetZoneList strt indx 0 frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.028 GetZoneList start index 0 from *src_net/src_node* net *network*

Description: An ZIP GetZoneList or GetLocalZones packet was received with a start index of 0.

ZIP2.029

Level: UE–ERROR

Short Syntax: ZIP2.029 GetMyZone strt indx not 0 frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.029 GetMyZone start index not 0 from *src_net/src_node* net *network*

Description: A GetMyZone ATP packet was received where the start index was not 0. The packet will be discarded.

ZIP2.030

Level: U–INFO

Short Syntax: ZIP2.030 No zn nm assoc wth nt *network*

Long Syntax: ZIP2.030 No zone name associated with net *network*

Description: There is no zone name associated with the indicated directly connected network.

Cause: This is a temporary condition where the router has received a ZIP GetMyZone packet before it has learned the zone name of the network for this interface.

ZIP2.031

Level: UI–ERROR

Short Syntax: ZIP2.031 *type*Reply disc nt *network* rsn *error_code*

Long Syntax: ZIP2.031 *type*Reply discarded net *network* reason *error_code*

Description: A ZIP GetZoneList, GetMyZone or GetLocalZones Reply was not sent for the indicated reason.

ZIP2.032

Level: UE–ERROR

Short Syntax: ZIP2.032 Ntfy trunc (*length*) frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.032 Notify truncated (*length* bytes) from *src_net/src_node* net *network*

Description: A ZIP Notify packet was received that was not long enough to contain the claimed zone name length.

ZIP2.033

Level: UE–ERROR

Short Syntax: ZIP2.033 *type* usr byt 2 not 0 frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.033 *type* user byte 2 not 0 from *src_net/src_node* net *network*

Description: A ZIP GetMyZone, GetZoneList or GetLocalZones ATP packet was received with user byte 2 of the ATP header not 0 from the indicated host. The packet will be discarded.

ZIP2.034

Level: UE–ERROR

Short Syntax: ZIP2.034 GetZoneList st indx *index,* high frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.034 GetZoneList start index *index,* too high from *src_net/src_node* net *network*

Description: A ZIP GetZoneList or GetLocalZones packet was received asking for zones with indices above the one given, but none were found.

Cause: A change in the ZIT, such as a zone deletion, has caused the indices to change values since the last GetZoneList request.

Action: Try again.

Cause: The remote node has a programing error.

ZIP2.035

Level: CE–ERROR

Short Syntax: ZIP2.035 query cnt 0 frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.035 query count 0 from *src_net/src_node* net *network*

Description: A ZIP Query packet was received with a network count of 0.

ZIP2.036

Level: CE–ERROR

Short Syntax: ZIP2.036 rply cnt 0 frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.036 reply count 0 from *src_net/src_node* net *network*

Description: A ZIP Reply packet was received with a network count of 0.

ZIP2.037

Deleted: Message deleted.

ZIP2.038

Level: UE–ERROR

Short Syntax: ZIP2.038 cnt *network_count* & len (*length*) disag frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.038 Network count *network_count* and DDP length (*length* bytes) disagree from *src_net/src_node* net *network*

Description: A ZIP Query packet was received where the expected length based on the ZIP network count does not agree with the actual DDP length of the packet.

Cause: Programming error at remote node.

ZIP2.039

Level: C–INFO

Short Syntax: ZIP2.039 unk nt *network_number* in qry frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.039 Unknown network number *network_number* in Query from *src_net/src_node* net *network*

Description: A ZIP query packet was received with the specified network number in the ZIP data, but this network is not in the RTMP database, or does not have a zone name in the ZIP database. Processing of the packet will continue.

ZIP2.040

Level: UE–ERROR

Short Syntax: ZIP2.040 unk nt *network_number* in rply frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.040 Unknown network number *network_number* in Reply from *src_net/src_node* net *network*

Description: A ZIP Reply packet was received with the specified network number in the ZIP data, but this network is not in the RTMP database. Processing of the packet will continue.

ZIP2.041

Level: C–INFO

Short Syntax: ZIP2.041 rq on unseed pt frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.041 Request on unseeded port from *src_net/src_node* net *network*

Description: A ZIP query or request was received on an unseeded port that hasn't obtained its net range from a seeded router. Processing of the packet will stop.

ZIP2.042

Level: UE–ERROR

Short Syntax: ZIP2.042 rply bd tpl nm len *length* nt *network* frm *src_net/src_node* nt *network*, ign

Long Syntax: ZIP2.042 reply bad tuple name length *length* network *network* from *src_net/src_node* net *network*, ignored

Description: A ZIP reply packet was received where one of the zone names was not of a legal length (between 1 and 32 characters). Processing of the reply ends with the ZIP tuple for the noted network number.

ZIP2.043

Level: UI–ERROR

Short Syntax: ZIP2.043 no mem for GtNtInf rq nt *network*

Long Syntax: ZIP2.043 no memory for GetNetInfo request net *network*

Description: The router was unable to generate a GetNetInfo request for the indicated network because no memory was available for the outgoing packet.

ZIP2.044

Level: UI–ERROR

***Short Syntax*:** ZIP2.044 GtNtInf disc nt *network* rsn *error_code*

***Long Syntax*:** ZIP2.044 GetNetInfo discarded net *network* reason *error_code*

Description: A GetNetInfo request was not transmitted on the indicated net for the specified reason.

ZIP2.045

Level: C–INFO

***Short Syntax*:** ZIP2.045 GtNtInf brdcst nt *network*

***Long Syntax*:** ZIP2.045 GetNetInfo broadcast on net *network*

Description: A GetNetInfo request for the indicated net was broadcast on the specified interface.

ZIP2.046

Level: UE–ERROR

***Short Syntax*:** ZIP2.046 zone *zonename* filtered from nt *network*

***Long Syntax*:** ZIP2.046 zonename *zonename* filtered from net *network*

Description: Zonename information was received on an interface but filtered by the input filter list.

ZIP2.047

Level: C–INFO

***Short Syntax*:** ZIP2.047 query for *net_num* snt to *net_num/node_num* nt *network*

***Long Syntax*:** ZIP2.047 query for *net_num* sent to *net_num/node_num* net *network*

Description: A ZIP query was sent for the indicated net to the specified router.

ZIP2.048

Level: UE–ERROR

***Short Syntax*:** ZIP2.048 unrcgnzd ZIP typ *type* fr *src_net/src_node* nt *network*

***Long Syntax*:** ZIP2.048 unrecognized ZIP type *type* from *src_net/src_node* net *network*

Description: A ZIP packet with an unrecognized command type was encountered.

ZIP2.049

Level: UI–ERROR

Short Syntax: ZIP2.049 no buf for ZIP rply to *net_num*/node

Long Syntax: ZIP2.049 no packet buffer for ZIP reply to *net_num*/node

Description: No packet buffer was available for sending a ZIP reply to the specified router.

ZIP2.050

Level: C–INFO

Short Syntax: ZIP2.050 rply net *net_num*snt to *src_net/src_node* nt *network*

Long Syntax: ZIP2.050 reply net *net_num* sent to *src_net/src_node* net *network*

Description: A ZIP reply was sent to the indicated router.

ZIP2.051

Level: UE–ERROR

Short Syntax: ZIP2.051 short (*length*) frm *src_net/src_node* nt *network*

Long Syntax: ZIP2.051 packet short (*length* bytes) from *src_net/src_node* net *network*

Description: A ZIP packet was received that was not long enough to contain the 2 byte ZIP header after the DDP header. The packet will be discarded.

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA
call 800–DIGITAL

In Canada
call 800–267–6215

In New Hampshire
Alaska or Hawaii
call 603–884–6660

In Puerto Rico
call 809–754–7575 x2012

ELECTRONIC ORDERS (U.S. ONLY)

Dial 800–234–1998 with any VT100 or VT200
compatible terminal and a 1200 baud modem.
If you need assistance, call 1–800–DIGITAL.

DIRECT MAIL ORDERS (U.S. AND PUERTO RICO*)

U. S. SOFTWARE SUPPLY BUSINESS
DIGITAL EQUIPMENT CORPORATION
10 Cotton Road
Nashua, New Hampshire 03063–1260

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
940 Belfast Road
Ottawa, Ontario, Canada K1G 4C2
Attn: A&SG Business Manager

INTERNATIONAL

DIGITAL
EQUIPMENT CORPORATION
A&SG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Services Business (SSB)
Digital Equipment Corporation, Westminster, Massachusetts 01473

*Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809–754–7575 x2012

Distributed Routing Software
Event Logging System Messages Guide
AA–QL2AB–TE

READER'S COMMENTS

What do you think of this manual? Your comments and suggestions will help us to improve the quality and usefulness of our publications.

Please rate this manual:

	Poor				Excellent
Accuracy	1	2	3	4	5
Readability	1	2	3	4	5
Examples	1	2	3	4	5
Organization	1	2	3	4	5
Completeness	1	2	3	4	5

Did you find errors in this manual? If so, please specify the error(s) and page number(s).

_____

_____

_____

_____

General comments:

_____

_____

_____

_____

Suggestions for improvement:

_____

_____

_____

_____

Name _____ Date _____

Title _____ Department _____

Company _____ Street _____

City _____ State/Country _____ Zip Code _____

DO NOT CUT – FOLD HERE AND TAPE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY LABEL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

digital™

Shared Engineering Services

550 King Street
Littleton, MA 01460–1289

DO NOT CUT – FOLD HERE