# Host Media Processing T1/E1 Board

## SOFTWARE INTERFACE

Documentation Revision 0.0a: January 15, 2015


266M000

This manual applies to the PCI and the  PCI Express versions of the board.  Because of the differences between the bus interface, there are minor differences in the boards These differences have been noted where appropriate.

# Contents

The Host Media Processing HMP T1/E1 Board Software Interface Manual



American Tel-A-System, Inc.
800-356-9148
• 4800 Curtin Drive • McFarland, WI 53558 •
• 4145 North Service Road, Suite 200 • Burlington, Ontario L7L 6A3 •
• 266M000 •

# 1.0 Introduction

The Host Media Processing T1/E1 Board was designed to be compatible with Asterisk and other similar environments. To that end, Amtelco has provided the appropriate drivers and channel drivers to interoperate with Asterisk. However, as there may be developers who, because Asterisk may not be suitable for their applications, wish to develop their own environment tailored for the needs of their specific application. The purpose of this manual is to provide these developers with the information necessary for them to complete this task.

## 1.1 The HMP T1/E1 Board

The HMP T1/E1 Board provides the hardware necessary to interface to DS1 digital telephony circuits. These circuits can be either T1 circuits as used in North America or E1 circuits as used in Europe and most of the rest of the world. The board can be equipped with either four or eight interfaces. Under software control, the interfaces can be configured as all T1, all E1, or half T1 and half E1. The board can also be configured on a per interface basis for Primary Rate ISDN, Robbed-Bit, Signaling, or Channel Associated Signaling to be compatible with most currently deployed DS1 circuits.

Because of the complexities of DS1 circuits, the board is equipped with an on board processor. This processor manages the details of the lower protocol layers relieving the host computer of this burden.

The details of the hardware aspects and the installation of the board are beyond the scope of this manual, and can be found in *the Host Media Processing T1/E1 Board Technical Manual 266M003*. That manual also contains a short primer on T1 and E1 circuits.

## 1.2 Communications with the Board

Communications between the host processor and the board takes place through the medium of messages. The messages are passed through a portion of memory that is accessible to both the host and the on board processor. This memory will be referred to as Dual-Ported Memory. It contains two mailboxes, one for messages to the board and one for messages from the board. There are also two flag bytes which are used to generate interrupts of the host and on board processor.

Messages consist of a two byte value giving the length of the message and the body of the message. The body of the messages is normally a NUL terminated ASCII string. The single exception to this is the message that is used to communicate the contents of Layer 3 messages as they may contain octets whose value is 0x00. The procedure for sending a message is to place the body in the mailbox and then set the length. When receiving a message, the body of the message should be removed and then the length cleared. The board will generate an interrupt when a message is placed in its transmit mailbox and when a message has been removed from its receive mailbox.

## 1.3 Drivers

Amtelco provides a LINUX driver for initializing and communicating with the board. The source code for this driver is available. For those developers using other operating system or who wish to customize the driver, this driver can serve as a model.

### 1.4 Protocol Layers

The HMP T1/E1 Board firmware provides support for the three lowest layers of the ISDN OSI reference model. These are Layer 1, the Physical Layer, Layer 2, the Data Link Layer, and Layer 3, the Network Layer.

The Layer 1 functions include specifics of the T1 and E1 circuits such as signal levels, bit rates, synchronization and framing. It also includes maintenance functions, error detection and reporting, and alarms.

The Layer 2 is concerned with those functions related to the reliable transport of data between the interfaces at the two ends of the circuits. This includes making sure that messages are received correctly and in order, and that if an error occurs, the proper steps are taken to recover from that error.

Layer 3, the Network Layer deals with the establishment and clearing of calls. This is done through a sequence of messages which are transported by the Data Link Layer.

The next three sections of this manual detail those aspects of these layers relevant to the host processor interface.

## 1.5 Robbed Bit and Channel Associated Signaling

In addition to supporting the ISDN protocols, the HMP T1/E1 board has support for the methods used to emulate analog circuits. For T1 circuits, the method is called Robbed-Bit Signaling and for E1 circuits, the method supported is Channel Associated Signaling.

### 1.6 Organization of the Manual

This manual covers the information needed by a developer to build applications using the HMP T1/E1 Board. The three sections after the Introduction deal with ISDN:

Section 2.0 Layer 1 - The physical layer.

Section 3.0 Layer 2 - The Data Link layer.

Section 4.0 Layer 3 - The Network Layer.

Section 5.0 covers Robbed Bit Signaling and Channel Associated Signaling.

Section 6.0 Covers Diagnostics, Events, and Maintenance issues.

Appendix A is a summary of the messages to and from the board.

# 2.0 Layer 1 - The Physical Layer

DS1 circuits are digital circuits that are used to interconnect telephony equipment and are capable of providing multiple channels of audio information. They come in two forms, T1 and E1. The T1 circuits commonly used in North America operate at a bit rate of 1.544 Mbps. and can provide 24 64 kbps. channels. E1 circuits as used in Europe and most of the rest of the world have a bit rate of 2.048 Mbps. and can provide 30 64 kbps.

The physical aspects of T1 interfaces are given in a set of ANSI specifications designated the T1.403 series. These include T1.403 which covers the electrical characteristics, T1.403-01 which covers Layer 1 of the T1 interface when used as a Primary Rate ISDN, and T1.403-02 which defines the use of robbed bit signaling. Additional standards cover other aspects such as the upper layers of PRI interfaces and are dealt with in subsequent sections.

The physical aspects of E1 interfaces are given in a set of ITU and ETSI specifications. ITU specification G.703 describes the electrical characteristics, and G.704 and G.706 describe framing and CRC procedures. The ETSI specifications deal with ISDN, and include ETS 300 011 which covers Layer 1 of the E1 interface when used as a Primary Rate ISDN, as well as ETS 300 166 and ETS 300 233. Additional standards cover other aspects such as the upper layers of PRA interfaces and channel associated signaling, and are dealt with in subsequent sections.

This section will give a brief description of the first physical layer and how they relate to the inter-workings of the HMP T1/E1 board and the application. It is not meant as an exhaustive reference or definition. For this, the reader is referred to the relevant specifications.

## 2.1 DS1 Electrical Interface

The physical characteristics of the T1 DS1 Interface are defined in *ANSI T1.403-1999 - Network and Customer Installation Interfaces - DS1 Electrical Interface*. The physical characteristics of the E1 Interface are defined in ITU *G.703 Physical/Electrical Characteristics of Hierarchical Digital Interfaces.* Framing is described in ITU *G.704 Synchronous Frame Structures Used at 1544, 6312, 2048, 8448 and 44736 kbit/s Interfaces*.

When dealing with a Primary Rate ISDN interface this is referred to as Layer 1. The DS1 electrical interface deals with the functions of timing and synchronization. It also provides the following services, signaling, data transmission, error detection, and maintenance functions.

DS1 interfaces are digital circuits that run at either a 1.544 or 2.048 Mbps. bit rate. A major problem and the reason for many of the design choices of these circuits is insuring the proper interpretation of data after it has been transmitted an extended distance over a connection consisting of physical wiring. This wiring imposes a variety of limitations such as loss, capacitance, reflections from impedance mismatches, and so on. The complexities of the DS1 interface arises from the methods used to overcome these problems.

The major issue is one of clocking. The two ends of a DS1 interface do not share a common clock. Instead, all clocking and framing information is contained in the DS1 signal. The network end, which provides the clocking for the interface, is typically tied into a hierarchy of clocks going to a master source operating at the national or international level. The customer end derives its clock from clues that are present in the signal sent over the DS1 interface from the network.

## 2.1.1 T1 Framing and Clocking

One second of the T1 bit stream is divided into 8,000 segments called frames. Each frame consists of 192 bits of data and one framing bit. These 192 bits are divided into 24 8-bit channels. These frames allow the DS1 interface to carry multiple channels of voice sampled at the 8000 8 bit sample rate for a rate of 64 kbps for each channel. Of course, the channels can also be used to carry data instead of voice. In addition to the 192 bits of data per frame, an additional framing bit is added for reliability and maintenance purposes.

Historically, the DS1 format has evolved over time as attempts were made to improve performance and reliability. This has resulted in several formats being in use at the same time. The two major formats are D4 and ESF. The difference between the formats is largely how frames are grouped together into what are called superframes. Superframes are used to define the location in the bit streams of bits used for signaling or maintenance. The D4 format has a superframe that consists of 12 frames. The ESF format which stands for Extended SuperFrame has a superframe that consists of 24 frames.

In addition to framing format, there are several other consideration. One of these is insuring that there is no DC offset to the signal. To do this, a scheme called Alternate Mark Inversion or AMI, is used. In this scheme, every other mark or one signal is indicated by a negative rather than a positive voltage. Another concern is zero suppression. If too many zeros are transmitted in a row, the signal can degrade. To prevent this, the bit encoding scheme is modified to prevent this from occurring. The scheme commonly used for this purpose is B8ZS or Bipolar with 8-Zero Substitution. This document will not go into the details of these schemes, but for an interface to work it is important that both ends use the same framing and zero suppression formats.

It is also necessary to match impedances and levels to minimize reflections and insure the proper interpretation of signals. This is called "line build out". The characteristics of the interface should be tailored to the length and type of wire used to connect the two ends.

### 2.1.2 E1 Framing and Clocking

One second of the E1 bit stream is also divided into 8,000 segments called frames. Each frame consists of 256 bits of data. These 256 bits are divided into 32 8-bit timeslots numbered 0-31. Timeslot 0 of each frame is used to carry framing information and timeslot 16 is used for signaling information. These frames allow the E1 interface to carry multiple channels of voice sampled at the 8 kbps rate for a rate of 64 kbps for each channel. Of course, the channels can also be used to carry data instead of voice.

Historically, the E1 format has evolved over time as attempts were made to improve performance and reliability. This has resulted in several formats being in use at the same time. The two major framing formats are FAS and CRC4. The difference between the formats is that FAS or Frame Aligned Signaling uses a simple code in timeslot 0 to align frames as even or odd while CRC4 adds additional error detection information using bit 1 of timeslot 0 in each frame. Sixteen frames are grouped together into a superframe. The even frames carry a 4 bit cyclic redundancy check code (CRC4) for each of two subframes, while the odd frames carry a bit pattern which indicates frame alignment within the multiframe with two bits used to indicate that a CRC error has been detected at the far end. Timeslot 0 in odd frames carry additional signaling information for maintenance purposes. Bit 3 is used for sending a Remote Alarm Indication (RAI), while bits 4 through 8 (Sa4-Sa8) are used for a maintenance data channel.

In addition to framing format, there are several other considerations. One of these is zero suppression. If too many zeros are transmitted in a row, the signal can degrade. To prevent this, the bit encoding scheme is modified to prevent this from occuring. Two schemes are commonly used for this purpose, AMI or Alternate Mark Inversion, and HDB3 or

High Density Bipolar with a maximum of 3 Zeros. This document will not go into the details of these schemes, but for an interface to work it is important that both ends use the same framing and zero suppression formats.

As with the T1 interface it is necessary to match impedances and levels to minimize reflections and insure the proper interpretation of signals. E1 circuits use one of two impedance values, 75 or 120 ohms.

## 2.2 Configuring the Interfaces

Several steps are needed to configure the DS1 interfaces on the HMP T1/E1 Board. The first of these is to specify whether the interface is to act as the Network Termination (NT) end or as the Terminal Equipment (TE) end. This is important because the TE end derives its clock from the interface while the NT end generates the clocks for the interface. Specifying which end the interface is to be is done with the **"ST"** command. This command consists of "ST" followed by a character for each of the interfaces on the board. This character can be a "T" for the TE end, an "N" for the network end, a "U" for an unused or undefined interface, or an "*" if no change is to be made. As an example, if the first two interfaces are to be set as NTs, the next four as TEs and the remaining are not being used, the command would be **STNNTTTTUU**. Note that for a board with only four interfaces, only four characters would need to be sent.

The next step is to specify the framing format for each interface. This is done with a command of the form **SFddfzsi** where dd is the interface number, f is a character defining the framing format, z is a character defining the zero suppression scheme, s specifies the signaling format, and i specifies the impedance value.

For a T1 interface the choices for framing format are "D" for D4 and "E" for ESF. The choices for zero suppression are "A" for AMI and "B" for B8ZS. The signaling format can be “N” for no signaling, “R” for robbed bit signaling, or “P” for Primary Rate ISDN. The buildout choices are specified with a number between 0 and 7 according to the following table:

| | |
|---|---|
| 0 | DSX-1 (0 to 133 feet)/0dB CSU |
| 1 | DSX-1 (133 to 266 feet) |
| 2 | DSX-1 (266 to 399 feet) |
| 3 | DSX-1 (399 to 533 feet) |
| 4 | DSX-1 (533 to 655 feet) |
| 5 | -7.5dB CSU |
| 6 | -15dB CSU |
| 7 | -22.5dB CSU |

For an E1 interface the choices for framing format are “N” for non-CRC4 or “C” for CRC4. The choices for zero suppression are “A” for AMI or “H” for HDB3. The choices for signaling format are “N” for no signaling, “C” for channel associated signaling, and “P” for Primary Rate ISDN. The impedance value choices are “B” for 75 ohms and “R” for 120 ohms. These characters refer to the type of connector typically used with each impedance, a BNC for 75 ohms and an RJ45 for 120 ohms. Note that the board is equipped with RJ45 type connectors.

At startup, the interfaces default to the most common choices. T1 interfaces are set to ESF, B8ZS, PRI and DSX-1 (0 to 133 feet). E1 interfaces default to CRC4, HDB3, PRI, and 120 ohms.

Another configuration issue is setting the bus clock mode. If one or more interfaces are acting as TEs and are connected to the network, then one of the interfaces should serve as the clock source for the board. This is done by sending a message of the form **SCx** where x is the number of the interface acting as the clock source. If x is set to “X”, no interface acts as a clock source. Note, that if the interface acting as the clock source, another interface should be selected to replace it.

## 7.3 Alarms

Alarms are used to indicate that there is a problem with the interface. There are three different alarm conditions which are designated by color, Yellow, Blue, and Red, with Red being the most severe. Green indicates that no alarm condition has been detected. The meaning of the alarms is as follows:

| | |
|---|---|
| Yellow Alarm | Remote Alarm Indication (RAI), a specific pattern is detected that depends on the framing mode |
| Blue Alarm | Alarm Indication Signal (AIS), an unframed all-ones signal is detected |
| Red Alarm | Loss Of Signal (LOS) |

For E1 interfaces, the red alarm is defined as Receive Carrier Loss or (RCL).

The Yellow and Blue alarms can be generated by the interface. The Yellow alarm or RAI should be sent when the incoming signal is effectively lost, such as when the interface detects a Red alarm. The Blue alarm should be sent when the originating signal has been lost, or when an action is taken that will cause a service disruption, for example when a line loopback has been implemented.

A message is sent every time an alarm changes state. This message takes the form **ARdd, ABdd,** or **AYdd** for red, blue, yellow alarms respectively, where dd is the circuit number. If all alarms are cleared, an **AGdd** message is sent.

The alarm states may be determined by sending the query message **"QA"**. The response message will have the form **QAaaaaaaaa** where each "a" represents the alarm state of a circuit starting with circuit 0. The alarm states are indicated by a 'G', 'Y', 'B' or 'R' for green, yellow, blue, or red alarm condition.

An interface can be set to generate a Yellow and Blue alarm condition at the far end under application control. This is done with a message of the form **AYdds** or **ABdds** for the yellow and blue alarm where dd is the circuit number and s is the alarm state, “S” to set the alarm, and “C” to clear the alarm.

For E1 interfaces where CRC4 is enabled, there are two additional error bits in each multiframe that are used to indicate CRC4 errors. These bits are automatically set by the interface.

On E1 interfaces several other conditions may produce alarms. For circuits using CAS signaling, a Receive Distant MF Alarm condition occurs when bit 6 of timeslot 16 in frame 0 has been set for two consecutive multiframes. This is indicated by a **AMddS** message where dd is the circuit number. When this condition is cleared an **AMddC** message will be issued. A Received Signaling All 1’s event occurs when the contents of timeslot 16 contains fewer than 3 0’s over 16 consecutive frames. This event is indicated by an **ASdd** message where dd is the circuit.

## 2.4 Line Interface Status Events

When certain physical conditions are detected by the Line Interface Unit, an event message will be generated. These events reflect open or short circuit conditions in the wiring and are not dependent on the framing or clocking of the interface. When such an event is detected, a message of the form **ESddss** is sent where dd is the circuit number and ss is a value indicating which condition or conditions exist. The bit values in ss are:

| bit | description |
|---|---|
| 1 | an open circuit has been detected between the transmit tip and ring outputs |
| 2 | the transmit current limiter has been activated |
| 3 | the receive carrier signal is lost |
| 4 | the jitter attenuator limit has been reached |

The other bits are reserved for future use. Note, that this message is sent whenever any of the event bits change.

## 2.5 Elastic Store Buffer Events

The synchronization of clocks on the two ends of the circuit may not be perfect. This can be due to the design of the equipment, noise, or jitter in the clocks. To compensate for this, each circuit on the HMP T1/E1 Board is equipped with an elastic store buffer in each direction. Under some circumstances, the elastic store buffer may be unable to handle the clock differences. When this occurs, an elastic store event will be generated. This causes a message of the form **EXddss**, where dd is the circuit number, and ss is a value indicating which event or events has occurred. Each bit in ss indicates a different event:

| bit | description |
|---|---|
| 0 | RSLIP - receive elastic store slip event |
| 1 | RESEM - receive elastic store buffer empties, frame repeated |
| 2 | RESF - receive elastic store buffer full, frame deleted |
| 3 | TSLIP - transmit elastic store slip event |
| 4 | TESEM - transmit elastic store buffer empties, frame repeated |
| 5 | TESF - transmit elastic store buffer full, frame deleted |

# 2.6 Maintenance Functions

Several maintenance functions are available to test whether a circuit is operating correctly. For T1 interfaces these functions conform to those which are described in *ANSI T1.403-1999*, and include provisions for performing loopbacks and collecting performance data. For E1 interfaces these functions conform to those which are described in *ETS 300 233* and include provisions for performing loopbacks. In general, these functions are initiated by the NT (central office) end of the circuit, or involve data that is sent from the TE (customer) end to the NT.

## 2.6.1 D4 Loopback Codes

Loopback codes are repeating patterns of bits that are sent from the NT to the TE to activate or deactivate loopbacks. D4 framing uses a 3 or 5 bit pattern that persists for at least 5 seconds to send the loopback request codes. The detection of 5 seconds of the pattern is considered to be the request for loopback.

Loopback codes may be sent from an NT interface using a command of the form **XCddc** where dd is the circuit number and c is the code mode. Valid values for c are, “0” for no code, “1” to activate a loopback, and “2” to deactivate the loopback. Timing of the loopback codes is the responsibility of the application. For D4, the code must be sent for at least 5 seconds.

When a TE interface receives a loopback code for 5 seconds it will respond with a message of the form **XLddc** where dd is the circuit number and c is the code. Valid codes are “1” to activate the line loopback and “0” to deactivate the line loopback. A “0” indicates the code has stopped. The application is responsible for activating and deactivating the loopback with the appropriate “XL” command.

## 2.6.2 ESF Bit-Pattern Messages

The ESF format uses a 16 bit pattern on the ESF facilities data link that repeats at least 10 times to transmit messages using the Facilities Data Link. These messages consist of 6 bits plus 10 framing bits, a 0 before and after the 6 bit code and eight consecutive 1s. The meanings of the 6 bit codes are defined in *ANSI T1.403-1999*, Table 4. Bit-patterned messages may be sent from either the NT or the TE, but certain messages are restricted to one or the other end. Three message values, 00h, 15h, and 1Fh are considered priority messages. These messages shall be transmitted for as long as a condition exists, but not for less than 1 second.

Most of the messages are concerned with activating or deactivating different loopbacks, however, some messages are reserved for other purposes which are beyond the scope of this document. The messages that are concerned with loopbacks and alarms are given in the table below:

Priority Messages

| | |
|---|---|
| 00h | RAI |
| 15h | Loopback retention and acknowledge |
| 1Fh | RAI-TE |

Command and Response Messages

| | |
|---|---|
| 07h | Line loopback activate |
| 1Ch | Line loopback deactivate |
| 0Ah | Payload loopback activate |
| 19h | Payload loopback deactivate |
| 09h | Reserved for network use (loopback activate) |
| 12h | Universal loopack (deactivate) |
| 17h | ISDN line loopback (NT2) |
| 10h | TE/CSU line loopback (NT1) |
| 0Eh | For network use (indication of NT1 power off) |

Bit patterned messages are sent with a command of the form **XCddcc** where dd is the circuit and cc is the message code. The bit pattern can be cleared with a message of the form **XCddFF**. It is the applications responsibility to time the message. A received bit pattern message is indicated by a message of the form **XLddcc** where dd is the circuit and cc is the message. An RAI message causes a Yellow alarm and no "XL" message will be sent. With one exception, messages will be reported when the pattern has cleared. That exception is the loopback retention and acknowledge message 15h. The message will be reported when it is detected. A message code of 55h will be reported when this message clears. The application is responsible for responding to bit patterned loopback messages by activating or deactivating loopbacks with the "XL" command.

### 2.6.3 E1 Loopback Codes

On E1 circuits loopback codes are repeating patterns of bits that are sent from the NT to the TE to activate or deactivate loopbacks. The Sa6 bits in the multiframe are used to carry the loopback codes, and the Sa5 bits are used to indicate that a loopback is activated.

Loopback codes may be sent from an NT interface using a command of the form **XCxxc** where xx is the circuit number and c is the code mode. Valid values for c are, "0" for loopback release, "1" to activate a loopback 2, and "2" to activate a loopback 1.

When a TE interface receives a loopback code it will respond with a message of the form **XLxxc** where xx is the circuit number and c is the code. Valid codes are "0" to release loopbacks, "A" to activate a loopback 2, and an "F" to activate a loopback 1. When a command to activate a loopback 2 is detected, the loopback is activated and the Sa5 bit is set to 0 in acknowledgment. The loopback is cleared when a loopback release command is received.

## 2.6.4 Loopback Control

Two loopback types are defined in *ANSI T1.403-1999*, a line loopback and a payload loopback. A line loopback involves transmitting received framing and channel data, while a payload loopback does not loopback the framing and CRC6 bits are not looped back. Payload loopbacks are only supported with ESF framing. A third loopback form, the framer loopback is also supported by the HMP T1/E1 board.

A loopback may be activated on a circuit by sending a command of the form **XLddm** where dd is the circuit number and m is the loopback mode. The following table gives the valid values for m:

| | |
|---|---|
| 0 | clear loopbacks |
| 1 | line loopback (TE) or local loopback (NT) |
| 2 | framer loopback |
| 3 | payload loopback |
| 4 | per channel payload loopback |
| 5 | analog loopback |
| 6 | remote loopback 1 |
| 7 | dual loopback (remote and local loopbacks) |

With a line loopback, data received from the line is looped back to the interface. Data from the interface is also transmitted to the internal telephony bus. With a local loopback, data from the internal telephony bus will be looped back to the telephony bus as well as being transmitted to the circuit. When a framing loopback is activated, data from the internal telephony bus is looped back to the telephony bus and a Blue alarm (AIS) is sent to the circuit. Payload loopbacks involve looping back data (excluding framing bits) from the circuit back to the circuit. The data is also transmitted to the internal telephony bus. With a per channel payload loopback, only data from selected channels is looped back to the circuit.

The payload loopback loops all 24 or 32 channels. Specific channels may be looped back using a command of the form **XLdd4aabbccee**. The hexadecimal arguments aa, bb, and cc define a bit map indicating which channels are to be looped back. Channels 0-7 are defined by aa, 8-15 by bb, and 16-23 by cc. For E1 interfaces ee corresponds to channels 24-31. The least significant bit in these arguments represents the lowest numbered channel. A set bit indicates the channel is to be looped back. To clear this loopback, a message of the form **XCdd400000000** must be sent. The command **XCdd0** will not clear this loopback.

## 2.6.5 Performance Report Messaging

*ANSI T1.403-1999* defines a Facilities Data Link or FDL for ESF framing on T1 circuits. This data link may be used to send unacknowledged frames of data using the LAPD format and the framing bits. The primary use of the data link is to send performance data from the TE to the NT to indicate whether the interface is operating correctly.

The Performance Report Message should be sent by an TE with ESF framing once a second and contains performance information about each of the last four seconds. The information sent includes the number of CRC errors, and whether a slip, line code violation, framing bit error, or framing event has occurred in that second. These messages are enabled by sending a command of the form **XF02.**

For interfaces set as an NT, the PRM messages are not normally passed up to the application to avoid excessive messaging. However, they can be turned on by sending a message of the form **XF01.** They can be turned off by sending a message of the form **XF00**. The messages to the application take the form **XXddd1d1d2d2d3d3d4d4** where dd is the circuit number, and d1d1 is the data for the first second, d2d2 is the data for the second second, and so on. The data for each second is sent as a 4 digit hexadecimal number where the bits are:

G3 LV G4 U1 U2 G5 SL G6     m.s.b of data

FE SE LB G1 R G2 Nm NI l.s.b. of data

where:

| | |
|---|---|
| G1 = 1 | CRC error event count = 1 |
| G2 = 1 | CRC error event count >1 & < 6 |
| G3 = 1 | CRC error event count> 5 & < 11 |
| G4 = 1 | CRC error event count > 10 & < 101 |
| G5 = 1 | CRC error event count > 100 & < 320 |
| G6 = 1 | CRC error event count > 320 |
| SE = 1 | Severely errored framing event > 0 (FE = 0) |
| FE = 1 | Framing synchronization bit error > 0 (SE = 0) |
| LV = 1 | Line code violation > 0 |
| SL = 1 | Slip event > 0 |
| LB = 1 | Payload loopback activated |
| U1,U2,R | Sent as 0s, may be modified |
| Nm,NI | modulo 4 sequence number |

A typical message when a line code violation occurred in the second before the most recent second reported would appear as:

**XX000003400200010000**

### 2.6.6 BERT Testing

Bit Error Rate Testing or BERT is used to test and stress communications links to determine if there are problems with the tranmissions facilities. It consists of sending a defined pattern and then detecting and counting any errors over an interval. A fixed rate of errors may be introduced at the transmit side to insure that the error detection at the receive end is functioning. The BERT pattern may be sent from one end to the other, or the far end may be put into a loopback so that the pattern is returned to the transmitting end.

Four commands are used to control the BERT transmit function. The first of these is a command of the form:

**XBddTEei**

which is used to set the pattern mode where dd is the circuit, e is the patten mode, and i selection the inversion mode for transmitted and received data. The pattern modes range from 0-7 and are:

| | |
|---|---|
| 0 | Pseudorandom 2E7-1 |
| 1 | Pseudorandom 2E11-1 |
| 2 | Pseudorandom 2E15-1 |
| 3 | Pseudorandom Pattern $QR_{ss}$ - a $2^{20}$ - 1 pattern with 14 consecutive zero restriction |
| 4 | Repetitive pattern |
| 5 | Alternating word pattern |
| 6 | Modified 55 octet (Daly) pattern as defined in ATIS T1-TRN25 |
| 7 | Pseudorandom 2E9-1 |

Bits 0 and 1 of the inversion mode i are used to control the inversion of the transmit and received data streams respectively, with a value of 0 being normal and 1 being inverted.

If mode 4 (repetitive pattern) is selected, the pattern is set by a command of the form:

**XBddPppppppppllcc**

where dd is the circuit number, the pp..pp argument defines a repetitive pattern that may range from 17 to 32 bits, ll is the length of the pattern, and cc is the alternating word count which is used in the alternating word mode to select the number of times a pattern defined by the first 16 bits is repeated before changing to an alternate pattern defined by the second 16 bits.

The command to select which timeslots will be used for transmitting the BERT data is defined by a command of the form:

**XBddTTtttttt**

where dd is the circuit number and the tttttt argument is a bit map of the timeslots with the most significant byte defining timeslots 0-7 and so on. To enable the BERT transmit function, the pattern would first be defined, then the pattern mode, and finally the timeslots would be selected. To disable the BERT transmit function, a command of the form:

**XBddTD**

should be issued where dd is the circuit number.

To enable the BERT receive function, the mode is set using a command of the form:

**XBddREei**

which is used to set the pattern mode where dd is the circuit, e is the patten mode, and i selection the inversion mode for transmitted and received data. The modes are the same as for the "XBddTE" command defined above. The timeslots receiving data are then selected using a command of the form:

**XBddRTtttttt**

where dd is the circuit number and tttttt is a bit map of the timeslots as defined above. The BERT receive function is disabled using a command of the following form where dd is the interface number:

**XBddRD**

The BERT error counters can be read using a command of the form:

**XBddRC**

where dd is the circuit number. The count will be returned in a message of the form:

**XBddRCeeeeeecccccccc**

where dd is the circuit number, eeeeee is a 24 bit error count in hexadecimal, and cccccccc is a 32 bit received bit count in hexadecimal. The received bit count can be used to determine the ratio of errors to bits received.

Two commands exist to introduce errors into the bit stream. To introduce a single bit error, a command of the form:

**XBddEB**

is used, where dd is the circuit number. To introduce errors at a fixed rate, a command of the form:

**XBddEIr**

is used were dd is the circuit number and r is the rate as follows:

| | |
|---|---|
| 0 | no errors are inserted |
| 1 | 10E-1 |
| 2 | 10E-2 |
| 3 | 10E-3 |
| 4 | 10E-4 |
| 5 | 10E-5 |
| 6 | 10E-6 |
| 7 | 10E-7 |

It should be noted that error rates greater than 10E-3 may cause problems with synchronization.

It may be necessary under some circumstances to resynchronize the BERT receiver. This is done with a command of the form **XBddS** where dd is the circuit number of the receiver.

It is beyond the scope of this document to discuss the details of BERT testing.

## 2.7 Monitor Modes

The HMP T1/E1 Board may be used for passively monitoring a T1 or E1 interface. When this is done, two interfaces are required to monitor the circuit, one to monitor the signal from the NT side and one to monitor the TE side. To set an interface to monitor the NT side, the interface type should be set to “M”, and to monitor the TE side, the type should be set to “O.”

When operating in the monitor mode, the audio signal from the interface will be processed as normal. Received Layer 3 messages will also be reported in the normal manner. Note, though, that because two interfaces are required, one for each direction, it is the responsibility of the application to correlate and combine the information from the two interfaces to form a complete call.

this page intentionally left blank

# 3.0 Layer 2 - The Data Link

Primary Rate ISDN conforms to a model defined in a set of standards (I.431, Q.921, Q.931 etc.) that is composed of a number of layers dealing with the physical interface, peer to peer signaling and call control. The Infinity Series H.100 T1/Primary Rate ISDN Board provides most of the control over the first two layers and optionally provides an interface to deal with the details of the third or call control layer. This section will give a brief description of the first two layers and how they relate to the inter-workings of the board and the application. It is not meant as an exhaustive reference or definition. For T1 interfaces, the reader is referred to the relevant specifications or Bellcore documents. For E1 interfaces, refer to ETS 300 125.

## 3.1 Layer 2

Layer 2 deals with the exchange of messages between peer devices and is referred to as the Data Link Layer. It is defined by the Q.921 specification. The protocol for the establishment of data links and the exchange of messages is called LAPD. On a Primary Rate Interface, LAPD messages or frames are exchanged over the 64 kHz. D-channel in the last timeslot of a T1 interface or timeslot 16 of an E1 interface.

Primary Rate ISDN normally employs only one data link on the D-channel at the same time. The data link is identified by a two part address called the Data Link Connection Identifier or DLCI. The two parts are the Service Access Point Identifier or SAPI, and the Terminal Endpoint Identifier or TEI. SAPI values can have a range of 0-63 and TEI values can range from 0-127. However, for the purposes of the Primary Rate ISDN interfaces only a single address is normally used, SAPI = 0, TEI = 0. Because only a single fixed DLCI is used, no TEI management procedures are employed.

LAPD messages or frames can take three formats. Information or I frames are used to perform information transfers between Layer 3 entities. Supervisory or S frames are used to perform data link supervisory control functions such as the acknowledgment of I frames, requesting the retransmission of I frames, or request the suspension or resumption of the transmission of I frames. The third format is Unnumbered or U frames which provide additional data link control functions and unnumbered information transfers for unacknowledged information transfers.

Supervisory Frames

| | |
|---|---|
| **RR** | Receive Ready, used to acknowledge I frames |
| **RNR** | Receiver Not Ready, used to suspend transmission |
| **REJ** | Reject, reject an I frame due to an error |

Unnumbered Frames

| | |
|---|---|
| **SABME** | Set Asynchronous Balanced Mode Extended Command |
| **DISC** | Disconnect Command |
| **UA** | Unnumbered Acknowledgment Response |
| **DM** | Disconnect Mode Response |
| **UI** | Unnumbered Information Command |
| **FRMR** | Frame Reject Response |

To exchange Information frames, a data link must first be established. This is done by a terminal requesting a TEI using a UI frame. The network can grant the TEI also using a UI frame. There are a number of possible states a data link can have once a TEI has been assigned. To exchange I frames, the link must be in the multi-frame established state. This is achieved by one end or the other sending an SABME frame and receiving a UA response. Once the data link is in the multi-frame state it can remain there indefinitely unless an error is detected or a request is made to disconnect from the multi-frame state.

Each I frame contains two state variables N(S) and N(R) which are sequence numbers for the I frame sent and the next I frame that is expected to be received. The supervisory frame that acknowledges the I frame has corresponding variables. These are used to insure that no I frames are lost. They can also be used to request the retransmission of an I frame should an error in transmission occur. Each frame also includes Frame Check Sequence or FCS field to insure the integrity of the I frame contents.

The HMP T1/E1 Board manages all of the details of the Layer 2 protocol. Acknowledgments and supervisory frames are sent as needed and recovery procedures initiated without requiring the intervention of the host application. The necessary timers are maintained on the board to handle retransmissions as specified by Q.921. Under normal circumstances, an application need only use the "D" command and response messages. Direct access to the information field can be dealt with using "LC" commands and responses (see section 3.4).

It is not necessary for the application to deal directly with the Layer 2 data link states. However, the state information for data link is available in the dual-ported memory. For details see section 8.3. If a Layer 2 error condition is detected that can not be handled with the normal recovery procedures, an "EM" error message will be sent by the board to the application allowing the application to take further action such as initiating a TEI management procedure or restarting the port. See section 5.2 for details of these messages.

## 3.2 Layer 2 States

Each of the spans has eight bytes reserved for Layer 2 state information beginning at an offset of 1C00h. The first byte for a span represent the Layer 2 states for the data link, the other bytes are reserved. These states are:

| | |
|---|---|
| 1 | TEI unassigned |
| 4 | TEI assigned, a TEI has been assigned, but multi-frame operation has not been established |
| 5 | Awaiting multi-frame operation, an SABME frame has been sent and awaiting a UA frame acknowledgment |
| 6 | Awaiting release from multi-frame state, a DISC frame has been sent and awaiting a DM frame |
| 7 | Multi-frame operation, exchange of I frames is possible |
| 8 | Timer recovery, a timer has expired and recovery procedures are in progress |

## 3.3 TEI Management

As Primary Rate ISDN uses a single, fixed data link, there are no management procedures to request or remove TEI assignments. The PRI interface is also a point-to-point interface with only a single terminal device connected to the network. Therefore there is no need for procedures to verify TEI assignments.

Initially, the data link is in the TEI assigned state. An attempt to send a message will place the data link in the multi-frame established state. However, the link can be disconnected or reestablished by using commands of the form **TDxxtt** and **TExxtt** where xx is the span number and tt is the TEI (00). A data link that is disconnected is placed in the TEI assigned state. If an attempt is made to send a Layer 3 message on a data link in the TEI assigned state, an attempt to reestablish the multi-frame established state will automatically be made by the board.

## 3.4 Sending and Receiving Layer 3 Messages

When the protocol level for a port is set at Layer 2 using the “SL” command, Layer 3 messages are sent and received using the “LC” message. The contents of the information field are appended to the “LC” message after the TEI characters. The “LC” message takes the form **LCxxsstt<info>** where xx is the interface number, ss is the SAPI, and tt is the TEI of the data link to be used. The SAPI and TEI normally both have values of 00.

When a Layer 3 message is received, an "LC" message of the form **LCxxsstt** will be placed in the receive mailbox. The info frame octets will be appended after the TEI.

There is no restriction on the contents of the Layer 3 info message other than a maximum length of 260 octets. It should be noted that as some of the octets may have a value of 0, "LC" mesasges are **not** NUL terminated. Currently only a DLCI with a SAPI of 0 and a TEI of 0 on a Primary Rate span is supported.

As an example to send a Q.931 CONNect message for a call reference of 1, the following bytes would be placed in the transmit mailbox:

| offset | bytes | description |
|---|---|---|
| 00h | 0Dh | length |
| 01h | 00h | high order byte of length |
| 02h | 'L' | |
| 03h | 'C' | |
| 04h | '0' | interface number |
| 05h | '3' | |
| 06h | '0' | SAPI |
| 07h | '0' | |
| 08h | '0' | TEI |
| 09h | '0' | |
| 0Ah | 08h | Q.931 message |
| 0Bh | 02 | call reference length of 2 |
| 0Ch | 80h | call reference of 1 with flag bit set |
| 0Dh | 01h | |
| 0Eh | 07h | CONNect message type |

To send this message the command would be placed in the transmit mailbox at an offset of 2, after which the length would be placed at the beginning of the mailbox, and the transmit flag would then be set to 01h. Receiving a message is the mirror image of this process.

# 4.0 Layer 3 - The Network Layer

## 4.1 Q.931 Messages

Call control for ISDN interfaces is handled at the Network Layer, also known as Layer 3. The specification that defines the message set used for call control is Q.931. Specifics for the U.S. implementation of the ISDN interface can be found in the ANSI document *Integrated Services Digital Network (ISDN) - Layer 3 Signaling Specification for Switched Circuit Bearer Service for Digital Subscriber Signaling System Number 1 (DSS1) (ANSI T1.607-1998)* and related ANSI documents for supplemental services. The corresponding European document is the European Telecommunication Standard *Integrated Services Digital Network (ISDN): User-network interface layer 3 Specifications for basic call control ETS 300 102* and other standards in the ETS 300 series.

Most switches currently deployed in North America use a common subset of Q.931 messages and procedures to simplify the interworking of equipment. This subset is called National ISDN, and has been implemented in a set of phases referred to as NI-1, NI-2, and NI-3. The details of these phases are described in various Telcordia (formerly Bellcore) documents. Some switches support additional features called Supplementary Services which are detailed in Q.932 and other specifications. Some of these are part of National ISDN 2, while others are part of standards such as QSIG or are proprietary.

In Europe, the most common form of ISDN is known as EuroISDN as defined by the ETSI standards. EuroISDN supercedes previous national standards and has been deployed in most countries in Europe, though some legacy systems may still conform to the various older national standards. For connections between PBXs or private branch exchanges the set of standards known as QSIG defined by the European Computer

Manufacturer's Association (ECMA) are used.

The message set defined in Q.931 consists of a number of command messages to establish and break down a call. These include **SETUP** to initiate a call, **ALERTing, CALL PROCeeding,** and **PROGress**, to handle the various stages involved in accepting or rejecting a call, **CONNect** to complete the connection, and **DISConnect, RELEASE,** and **RELEASE COMplete** to break down a call. Additional messages are provided as acknowledgments, **SETUP ACKnowledge** and **CONNect ACKnowledge**.

Features beyond the basics of making a call are supported by additional messages including **NOTIFY, STATus, STATus ENQuiry, and FACILITY.**

In addition to the basic codes for the various message types, each Q.931 message consists of a number of information elements. These information elements are used to contain items such as the called and calling numbers, whether a call is speech or data, the B channel involved, and other information about the progress of a call. For some messages such as the SETUP message a large number of information elements are possible, many of which are optional or only used for certain types of calls. As an example a typical SETUP message from a central office switch contains eight information elements and eighty octets of information.

Each active call has a call reference that is part of every Q.931 message dealing with that call. This allows for the juggling of multiple calls on a B-channel. The call reference is granted by the originating party and remains in effect until the call is finally released, no matter what happens in the interim. However, a call reference is only valid at that particular interface in messages between the network termination and the terminal equipment.

Q.931 also describes a state model for the progress of a call. The states are defined by which type of message has been received or sent. State information may be exchanged through STATus messages as a check on the operation of the terminal and to insure that the terminal and network are in agreement on the condition of a particular call. For each state of a call, only certain message types are appropriate.

## 4.2 "D" Command & Response Messages

The Q.931 message set can be complicated, with many options and information elements to deal with. On the HMP T1/E1 Board, an application can directly access the information field of a Q.931 message by using the Layer 2 support level. In some cases this may be desirable when dealing with equipment that does not support National ISDN 1 or has non-standard features or messages. However, the HMP T1/E1 Board does provide a simpler interface through the Layer 3 support level for those instances when only simple call control is needed for equipment that conforms to NI-1, NI-2, and NI-3 or EuroISDN. Note that the board automatically makes the adjustments to the protocol for EuroISDN for interfaces that are set to E1.

The Layer 3 support level is achieved using "D" channel commands and responses. These messages consist of an initial "D", a second character indicating the Q.931 message type, for example an "A" for an ALERTing message or a "C" for a CONNect message, a two digit hexadecimal number indicating the B channel, and for some messages additional arguments. The 23 B channels for interface 0 are numbered 00-16, the B channels for interface 1 are numbered 20-36, and so on. E1 interfaces have 30 B channels and are numbered 00-1D, 20-3D etc.

Each of the additional arguments are typically encoded as a single character or as a pair of hexadecimal digits in the case of the Cause element or Call references. The intent is to keep the messages short and simple with only the necessary information and allow the on board firmware to fill in the details of the actual Q.931 messages sent.

"D" channel messages and the corresponding Q.931 messages are given in the following list:

| | |
|---|---|
| **DAxx** | ALERTing |
| **DCxx** | CONNect |
| **DCxxA** | CONNect ACKnowledge |
| **DDxx** | DISConnect |
| **DFxx** | FACILITY |
| **DIxx** | INFOrmation |
| **DKxx** | Keypad element INFOrmation message |
| **DNxx** | NOTIFY |
| **DPxx** | CALL PROCeeding |
| **DPxxP** | PROGress |
| **DRxx** | RELease COMplete |
| **DRxxR** | RELease |
| **DSxx** | SETUP |
| **DXxx** | STATus |

To further simplify the application interface, the board will automatically send an acknowledgment messages in some cases. A CONNect ACK is sent where needed in response to a CONNect message. A RELease is sent in response to a DISConnect, and a RELease COMplete is sent in response to a RELease message. A CALL PROCeeding message is sent in response to a SETUP message terminal.

The on board firmware selects a call reference when a call is originated with a SETUP command. Additional commands for that B-channel will use this call reference so the application doesn't need to supply it. For received layer 3 messages, the call reference is always included in the corresponding "D" message for those cases where it may be needed by the application. The call reference consists of a four digit hexadecimal number. The most significant bit of this number is cleared when used by the originating side of the call and set when used by the terminating side of the call.

The "D" message set provides the application with a simple interface that gives the application complete control of the call but frees it and the programmer from being forced to deal with the details of coding Q.931 messages.

## 4.3 Information Elements

Information elements are used to encode information that may be required to handle a call. In "D" messages on the HMP T1/E1 Board, these elements are usually represented by single argument characters. Only those elements that are needed are included in the messages. Elements that are unambiguous or constant are encoded by the board in the Q.931 messages but are not part of the "D" messages. Some elements that are rarely used are not part of the "D" message scheme. Details on which arguments are used in which specific message can be found in Appendix A.

### 4.3.1 Call Reference

The call reference element is used to keep track of a specific call when handling multiple calls on an interface. The call reference for Primary Rate ISDN is a number between 0001-7FFFh. When used by the terminating side of a call, the most significant bit of the reference is set, i.e. the ranges is from 8001-FFFFh. The call reference is assigned by the originating side of the call. The HMP T1/E1 Board will automatically assign call references for calls that originate from the board. The call references are assigned sequentially for each interface..

### 4.3.2 Bearer Capability

The Bearer capability is used to describe the type of call being placed. This argument is present in both SETUP commands and responses. The five options for bearer capability are:

- **A** 3.1 kHz audio, 64 kbps, circuit mode
- **D** Unrestricted digital information, 64 kbps, circuit mode
- **E** ECMA (QSIG) unrestricted digital data
- **R** Rate adaption from 56 kbps, 64 kbps, circuit mode
- **S** Speech, 64 kbps, circuit mode

Note, that for speech and audio, these are encoded as Mu-Law in North America and A-Law in Europe and most of the rest of the world. It is always assumed that T1 interfaces are Mu-Law and E1 interfaces are A-Law.

### 4.3.3 Progress Indicator

Progress indicators are used to indicate something about the nature of the interworking of a call. This can be information about the fact that the call may not be an end to end ISDN call, or it may indicate the presence of inband information such as call progress tones or announcements. The codes for a progress indicator are:

- **C** Call is not end to end ISDN, call progress information may be available inband
- **D** Destination address is non-ISDN
- **O** Origination address is non-ISDN
- **I** Inband information or appropriate pattern now available
- **W** Delay in response at destination interface
- **N** no progress indicator

### 4.3.4 Sending Complete

The Sending Complete information element is included when the sender knows that the information in the called party number information element is sufficient to complete the call. This information element is not part of the National ISDN guidelines and is typically not used for calls on the public phone network in North America though it may be used for calls between PBXs. It is typically present in Europe, especially on calls originating from the network side. It is not necessary to include this element, and the user or network may determine when sufficient address digits have been received by examining the digits themselves. To indicate that a Sending Complete information element should be included in a SETUP message, an "@" character should be placed after the progress indicator character but before the calling party number. If a Sending Complete information element has been included in an incoming SETUP message, a "C" character will be used as the separator between the calling and called party numbers instead of the "/".

### 4.3.5 Calling Party Number and Subaddress

The Calling Party information element is used in a SETUP message to indicate the number or the party originating the call. Calling party numbers in "D" messages are represented by a string of digits following the progress indicator or sending complete characters.

In addition to the digits of the calling party number there is also information about the type of number and numbering plan and presentation status contained in the Calling Party information element.

The default number type and numbering plan is always unknown number type in an unknown numbering plan. With this coding, the terminating switch will route the call based on the digits of the number. Under some circumstances, it may be necessary to change this. This can be done by appending a character to the beginning of the digit string. The choices for this character are:

| | |
|---|---|
| **I** | International number/ISDN numbering plan |
| **L** | Subscriber number/ISDN numbering plan |
| **N** | National number/ISDN numbering plan |
| **n** | National number/unknown numbering plan |
| **S** | Subscriber number/private numbering plan |

In received SETUP messages, a character indicating the number type and plan is always included. In addition to the above characters, the “U” character will indicate an unknown number type in an unknown numbering plan. When encodings other than the default are required are network dependent.

The default presentation indicator for calling party numbers is “presentation allowed.” Appending a “P” to the beginning string will set the presentation indicator to “presentation restricted.” If a “P” is not followed by any digits before the separator character, the calling party number element will be present with “presentation restricted” but no digits. The “P” may not be used in conjunction with a character to set the numbering plan as described in the previous paragraph.

A subaddress number may also be added to the number string. This is done by attaching a hyphen as a separator and the subaddress digits to the number string. For example, in the string “5551212-1234”, 5551212 is the address number and 1234 is the subaddress. Subaddresses are used to indicate extensions when calling to or from a private exchange such as a PBX. The use of subaddresses is network dependent.

### 4.3.6 Called Party Number and Subaddress

The Called Party Number information is used in a SETUP message to indicate the intended destination address of the call. In primary rate ISDN, the Called Party Number is a mandatory information element.

In SETUP command and response messages, the Called Party Number follows the Calling Party Number. The two are separated by the characters “/” or “C” depending on whether or not the Sending Complete information element is present in the Layer 3 message.

In command messages, the number type and numbering plan may be indicated with the same prefix characters as for the Calling Party Number. The number type and numbering plan are not indicated in the response message.

A subaddress number may also be added to the number string. This is done by attaching a hyphen as a separator and the subaddress digits to the number string. For example, in the string "5551212-1234", 5551212 is the address number and 1234 is the subaddress. Subaddresses are used to indicate extensions when calling to or from a private exchange such as a PBX. The use of subaddresses is network dependent.

### 4.3.7 Redirecting & Original Called Numbers

The Redirecting Number information element is an address from which a call has been deflected or forwarded, and usually appears in a SETUP message. In SETUP command and response messages, the redirecting number comes after the called party number and subaddress and takes the form **Rr[#]** where the "R" indicates a redirecting number, the r is the reason the call was redirected, and # are the digits of the redirecting number. Valid redirect reasons are, "A" for always, "B" for busy, "D" for deflected, "N" for no answer, and "O" for out of order. More than one redirecting number may be present.

Pre-NI1 Nortel DMS-100 and DMS-250 switches used a proprietary information element, the Original Called Number to indicate the number that had been called by the user. This information element serves the same purpose as the first Redirecting Number information element. In SETUP command and response messages, this takes the same form as the Redirecting Number except that an "O" replaces the "R."

The following information elements are used in messages other than the SETUP message.

### 4.3.8 Cause

The Cause element is present in a number of different messages. Its purpose is to give a reason why a particular message such as a

DISConnect has been issued. Some causes such as Normal Clearing are part of a normal call while other causes may indicate an error condition. The cause element is quite lengthy, and therefore to simplify things the argument is represented by a two digit hexadecimal number in “D” messages. The first digit of the cause serves to divide the causes into different classes, i.e. a ‘0’ indicates a cause related to dialing. The list of causes are:

| | |
|---|---|
| **01** | Unallocated Number |
| **02** | No route to specified transit network |
| **03** | No route to destination |
| **06** | Channel unacceptable |
| **07** | Call awarded and being delivered in an established channel |
| **10** | Normal clearing |
| **11** | User busy |
| **12** | No user responding |
| **13** | User alerting, no answer |
| **15** | Call rejected |
| **16** | Number changed |
| **1A** | Non-selected user clearing |
| **1B** | Destination out of order |
| **1C** | Invalid number format (incomplete address) |
| **1D** | Facility rejected |
| **1E** | Response to STATus ENQuiry |
| **1F** | Normal, unspecified |
| **22** | Circuit/channel congestion |
| **29** | Temporary failure |
| **2A** | Switching equipment congestion |
| **2B** | Access information discarded |
| **2C** | Requested channel not available |
| **2F** | Resource unavailable, unspecified |
| **32** | Requested facility not subscribed |
| **39** | Bearer Capability not authorized |
| **3A** | Bearer Capability not presently available |
| **3F** | Service or Option not available, unspecified |

| | |
|---|---|
| **41** | Bearer capability not implemented |
| **45** | Requested facility not implemented |
| **4F** | Service or option not implemented, unspecified |
| **51** | Invalid call reference value |
| **58** | Incompatible destination |
| **60** | Mandatory information element is missing |
| **61** | Message type non-existent or not implemented |
| **63** | Information element non existent or not implemented |
| **64** | Invalid information element contents |
| **65** | Message not compatible with call state |
| **66** | Recovery of timer expiry |
| **6F** | Protocol error, unspecified |
| **7F** | Interworking, unspecified |

### 4.3.9 Connected Number

The Connected Number information element may be included in a CONNECT or DISCONNECT message to indicate that the connected party is not the party originally called, as when a call has been forwarded or transferred by a PBX. To add a Connected Number to a CONNECT command, a "#" followed by the digits of the connected number are added after the progress indicator character. Similarly, the presence of a Connected Number in a CONNECT message will be indicated by a "# followed by the digits of the connected number after the progress indicator character in a CONNECT or DISCONNECT response message.

### 4.3.10 Notification Indicator

The Notification Indicator information element is used in the NOTIFY and some other types of messages to impart information relating to a call. In "D" command and response messages, it is indicated by a single character:

| | |
|---|---|
| **A** | Call Transfer - Alerting call |
| **B** | User Bridged onto call |
| **D** | Privacy Disabled |
| **E** | Privacy Enabled |
| **F** | Call is Forwarded |
| **H** | Call on Hold |
| **h** | Remote Hold |
| **I** | Idle resource or monitored user |
| **M** | Monitoring discontinued |
| **N** | No notification indicator or unknown |
| **R** | Call Retrieved from Hold |
| **r** | Remote Hold Release |
| **T** | Call Transfer - Active call |
| **U** | Update of Service Profile |
| **W** | Call Waiting |

### 4.3.11 Display

The Display information element is sometimes included in a message to impart additional information such as the calling party name, the status of the call, or features. Text can be sent to a terminal in an INFOrmation message or it can be included as an information element of another message. Text is normally only sent in messages from the NT (CO) side. It should be noted that both in North America and Europe, the Calling Party Name is not sent using a Display information element, but is sent as a Supplementary Service using a Facility information element.

The Display information element is defined in ANSI T1.607, Q.931 and the QSIG standards. In this format, text consists of unformatted characters up to a maximum of 20 or 80 (depending on the implementation).

In this format, text is buffered prior to the command for the Layer 3 message that will include it. The command to put text in the buffer takes the form **DTA[text].** This command is sent before the command for the Layer 3 message that is to include the text. The response message for text contained in a message takes the form **DTxx[text]**.

Pre NI-1 Nortel DMS-100 and DMS-250 switches used a proprietary form of the Display information element which could be sent from either the network or customer side to pass name information. To send information a command of the form **DTDt[text[** is used to place the text in the buffer, where t is a tag indicating the type of text. The allowed types are “C” for Calling Party Name, “O” for Originating Party Name, and “X” for Connected Party Name. When a Display element in this format is received, a response of the form **DTxxt[text]** will be sent where xx is the B-channel, and t is the name type as indicated above.

### 4.3.12 Network Specific Facilities

In North America some carriers offer special routing facilities that may provide lower tariffs when placing outgoing calls into the network. To access these facilities, the SETUP message must include a Network Specific Facilities information element. This information element consists of a Carrier Identification Code and a facilities code specified by the network. The carrier code consists of three ASCII digits assigned as part of the North American Numbering Plan. As an example, AT&T is identified by “288” which spells out “ATT” on the keys of a touch tone telephone. The facilities code is a single octet. For instance AT&T’s MEGACOMM service is indicated by the octet 0E3h.

If the Network Specific Facilities information element is present, the network will use that facility when routing outbound calls. If the element is not present, the default facility will be used.

A Network Specific Facilities element may be specified by a command of the form **SAddabchh** where dd is the interface, abc is the Carrier Identification Code, and hh is a pair of hexadecimal digits representing the facilities code. This command will affect all outgoing calls made on that interface. As an example “SA00288E3” would set the network specific facilities to AT&T MEGACOMM for interface 00. A command of the form **SAdd** will specify that no network specific facilities element is included in SETUP messages.

## 4.4 Call Handling Procedures

This section will describe the basic procedures used to establish and clear a call. The proper messages and their order will be given.

### 4.4.1 Call Establishment

A call is initiated with a SETUP message. The SETUP message contains a number of pieces of information. The first item is the Bearer Capability. This describes the type of call, i.e. Speech, Data, etc. The second element is the Channel Identity which specifies the B-channel. These items are mandatory. Calls from a central office may also include a progress indicator. This is used to indicate that a call is not ISDN from end to end. The SETUP message may also contain the calling party number, though this is optional. Primary Rate ISDN SETUP messages will always have the called party number. Finally, SETUP messages from a CO may include a display text element. Other information elements are listed as optional in the specifications, but are not typically used in basic call handling. These elements are not supported in the HMP T1/E1 "D" message set.

When originating a call on an interface, the SETUP command takes the form **DSxxbp(#)/#** where xx is the B-channel, b is the bearer capability, p the progress indicator, # is the optional calling party number, and /# is the called party number. Sending Complete, Redirecting Number or Original Called Number information elements may also be added to this command. The Calling Party Name or a Display information element may be added by sending the appropriate command message for those features immediately before the "DS" command.

When receiving a call, the response message is also a "DS" message. It will take the form **DSxxrrrrbp#/#** where rrrr is the call reference, b the bearer capability and p the progress indicator. The first number is the calling party number while the number after the "/" is the called party number. Note, that if the SETUP message includes a Sending Complete, the "/" will be replaced with a "C."

A CALL PROCEEDING is automatically sent in response to a received SETUP message if there are no errors in the SETUP message. A CALL PROCeeding message is sent by the terminating end to indicate that enough information has been received to process the call. Several different messages can be sent by the application in response to a SETUP message. These are PROGress, ALERTing, or CONNect or. The order of these messages is fixed, though not all may be sent for each call. A PROGress message is sent to indicate that an interworking situation is present, for example, the destination is not an ISDN device. The ALERTing message is sent when the call is presented to the user, and the CONNect message is sent when the user answers the call. Because each message reflects the call state, a CALL PROCeeding or PROGress message should never be sent once an ALERTing message has been sent, and a CALL PROCeeding, PROGress, or ALERTing message should not be sent once a CONNect message has been sent.

If a call is to be rejected, a RELease COMplete message should be sent if no other message, for example, an ALERTing message has been sent previously. If one of the messages above has been sent, then a DISConnect message should be used. The reason for rejecting the call should be specified using the cause parameter.

The PROGress command takes the form **DPxxPccp.** The ALERTing command takes the form **DAxx(p)** where xx is the B-channel and p is a progress indicator. The CONNect command takes the form **DCxx(p)** where xx is the B-channel, and p is the optional progress indicator. A connected number may also optionally be added to the command after the progress indicator character.

A PROGress response message will take the form **DPxxrrrrPccp** where rrrr is the call reference, cc is the cause code, and p is the progress indicator. The ALERTing response message will include the call reference taking the form **DAxxrrrrp** where rrrr is the call reference and p is the progress indicator. CONNect response messages will take the form **DCxxrrrrp** where xx is the B-channel, rrrr is the call reference, and p is the progress indicator. If they are present in the Layer 3 message, the connected number and connected party name may be appended at the end of the "DC" message.

A CONNect ACKnowledge may be sent in response to a CONNect message. This is mandatory for network terminations but optional for terminal endpoints. The board will normally respond to CONNect messages by sending the CONNect ACKnowledge message, however, if system option1 is set to "Y", then it is up to the application to send the CONNect ACKnowledge command which takes the form **DCxxA**. The response message for a CONNect ACKnowledge takes the form **DCxxrrrrA.**

### 4.4.2 Call Clearing Procedures

Call clearing procedures fall into two cases. In one the far end initiates the clearing, and in the other, the near end is the initiator. In either case, the initiator sends a DISConnect message. The other end of the connection then responds with a RELease message. The initiator then sends a RELease COMplete. At this point, the call is no longer in existence, the call reference is no longer valid, and the B-channel is no longer in use. The "D" command to send a DISConnect has the form:

**DDxxcc**

where xx is the B-channel and cc is the cause value.

The response to this should be a RELease message which will cause the Layer 3 Protocol software to send a message of the form **DRxxrrrrRcc** to the application. It will also automatically send a Layer 3 RELease COMplete message. An example of disconnecting is:

| | | |
|---|---|---|
| xmt: | **DD1210** | DISConnect, cause normal clearing |
| rcv: | **DR120001R10** | RELease message |

The Layer 3 Protocol software will automatically respond to a DISConnect message by sending the RELease response. The other end should then respond with a RELease COMplete. An example of this message sequence is:

| | | |
|---|---|---|
| rcv: | **DD12rrrr10N** | DISConnect, normal clearing |
| rcv: | **DR12rrrr00** | RELease COMplete, no cause specified |

Note that there is no need for the application to send a RELEASE message.

Calls should be disconnected using the DISConnect message except in a few circumstances. When originating a call, and no ALERTing or CALL PROCeeding message has been received in response, or when a B-channel is not assigned to the call, the call should be released using a RELease COMplete message. The following is an example where no response has been received:

| | | |
|---|---|---|
| xmt: | **DS06SN5551234** | SETUP message, called #5551234 |
| | no response | |
| xmt: | **DR0610** | RELease COMplete, normal clearing |

It is possible for calls that are not assigned to a B-channel to be cleared. This is done by specifying the call reference. The DISConnect command takes the form **DDxxccrrrr** where rrrr is the call reference. The RELease command takes the form **DRxxRrrrr,** and the RELease COMplete command takes the form **DRxxccrrrr**. The responses take the same form, however, the B-channel will be coded as B-channel 23 on the interface to indicate that the call is not associated with a B-channel, that is B-channel 17h, 37h, etc. will be used. For E1 interfaces this will be B-channel 30 on the interface, that is 1Dh, 3Dh, etc.

## 4.5 Call Handling Examples

This section will give some simple examples of the "D" messages that would be sent and received for a typical speech call. Only the "D" messages are shown.

### 4.5.1 Originating a Call

In this example, an outgoing call is initiated, answered, and then disconnected by the far end hanging up. The seventh B-channel (number 06) of the first interface is used.

| commands | responses | description |
|---|---|---|
| **DS06SN5551234/5551212** | | SETUP message |
| | **DP060003N** | CALL PROCeeding |
| | **DA060003N** | ALERTing message |
| | **DC060003N** | CONNect message |
| | **DD06000310N** | DISConnect message |
| | **DR06000300** | RELease COMplete |

In the "DS" command, the 06 indicates the B channel, in this case the seventh channel for the first interface, the "S" indicates a bearer capability of speech, the "N" indicates that there is no progress indicator, the calling number is 5551234, and the called number is 5551212. The far end sends a CALL PROCeeding to indicate that it has received the SETUP. The far end then sends an ALERTing message indicating that it has accepted the call and is alerting the user. The CONNect message indicates the user has answered the call. The DISConnect occurs when the user hangs up. The cause code indicates normal clear. This causes the board to send a RELease message. The far end finally responds with a RELease COMplete message to end the call. The cause code of 00 indicates there was no cause code.

### 4.5.2 Receiving an Inbound Call

In this example, a call is placed by the far end and terminated by the board.

| commands | responses | description |
|---|---|---|
| | **DS06SN5551000C5551212** | SETUP message |
| **DA06I** | | ALERTing message |
| **DC06** | | CONNect message |
| **DD0610** | | DISConnect message |
| | **DR06R00** | RELease message |

The initial SETUP message indicates that this is a speech call. The digits following the “N” are the calling party number, the “C” indicates a Sending Complete and the digits after the “C are the called party number. B-channel 06 was selected. When the destination user is notified of the call, an ALERTing message is sent with the progress indicator for inband tones. Ringback would be played to the B-channel to notify the caller. A CONNect follows when the call is answered. A “DD” command is issued with a cause of normal clearing to end the call. The far end responds with a RELease message. This causes the board to send a RELease COMplete message to end the call.

In this example, the SETUP message from the network indicates a speech call with no progress indicator and no signal. The called number is 5551000 and the called number is 5552000. The application sends a “DA” command to cause an ALERTing message to be sent with a progress indicator of “I” to indicate that ringback tone is being played.. A CONNect follows which causes the network to respond with a CONNect ACKnowledge. A DISConnect message with normal clearing is sent to clear the call. The network responds with a RELease which causes the board to send a RELease COMplete to end the call.

## 4.6 Call Processing Errors

Calls do not always progress smoothly. Sometimes there is an error in dialing, the called number is busy, or no route may be available. It is also possible for a terminal to request a facility that is not supported or subscribed to. Switching errors can occur or some other problem may arise creating an error situation.

In these cases, rather than just clearing the call, it may be desirable to play a tone or announcement to inform the caller that there is a problem. Examples would be playing a busy tone if the destination was in use or playing reorder if it is out of service. When this is the case, a PROGress message should be sent. This message will indicate the nature of the problem and include a progress indicator of "inband tone or announcement". To send a PROGress message, the application should a command of the form **DPxxPccp**, where xx is the B-channel of the call, cc is the cause of the PROGress message, and p is the progress indicator.

As an example, the message **DP06P11I**, would indicate that the terminal trying to place a call on the 06 B-channel has dialed a busy number (cause 11). The "I" indicates that there is an inband tone present. It is the application's responsibility to provide the inband busy signal.

Similarly, if the HMP board originated the call, a response of the form **DPxxrrrrPccp** will indicate that a problem has occurred. The cause will be indicated by the cc, and the progress indicator will indicate if any error tones or messages are available.

## 4.7 Layer 3 Timers

Several timers are specified at Layer 3 to deal with cases where the expected response is not received. These timers prevent calls from just hanging indefinitely. If the timer times out, a specific action is taken to either elicit a response or clear the call. The following table lists the supported timers.

No action relating to the board is required on the part of the application when a timeout occurs, as the firmware will send the appropriate message. Of course the application must take the appropriate internal actions such as removing a call from internal tables or clearing the near end connection. Layer 3 timers may be disabled by setting system option 4 to 'Y'.

| Timer | timeout value | cause for initiation | cause for termination | action on expiry |
|---|---|---|---|---|
| 1st T303 | 4 sec. | SETUP is transmitted | CALL PROC, ALERT., CONNect or RELease COMplete | retransmit SETUP |
| 2nd T303 | 4 sec. | SETUP retransmitted | as above | send RELease |
| T305 | 30 sec. | DISConnect sent | DISConnect, RELease, RELease COMplete | send RELease |
| 1st T308 | 4 sec. | RELease sent | RELease, RELease COMplete | retransmit RELease |
| 2nd T308 | 4 sec. | RELease retransmitted | as above | send RESTART |
| T310 | 10 s. NT<br>120 s. TE | CALL PROC received | ALERTing, CONNect or DISConnect | send DISConnect |
| T316 | 120 sec. | RESTART | RESTART ACK | retransmit RESTART |
| T322 | 4 sec. | STATUS ENQUIRY | STATUS, DISC.<br>RELease<br>RELease COM | send DISConnect |

When a timer times out, the board will send a message of the form **DExxe** where xx is the B-channel, and e is a value specifying the timer.

The values of e are:

| | |
|---|---|
| **0** | T303 has expired the first time |
| **1** | T303 has expired the second time |
| **2** | T310 has expired |
| **3** | T305 has expired |
| **4** | T308 has expired the first time |
| **5** | T308 has expired the second time |
| **6** | T316 has expired |
| **7** | T322 has expired |

## 4.8 Supplementary Services

Supplementary Services are additional features that go beyond basic call handling. They may include the provision of additional information about a call such as an identifying name associated with one of the parties to a call, or they involve routing procedures such as transferring a call or call deflection.

Supplementary Services have been defined by a number of entities. Proprietary services have been specified by the various switch vendors, and are unique to a specific vendor or switch type. Those services that fall under the various phases of National ISDN have been defined by Telcordia (formerly Bellcore) in documents generated by that body and by American National Standards Institute. In Europe, an effort known as QSIG has defined a number of services to promote interoperability between Private Branch Exchanges (PBXs). These are detailed in documents from the European Computer Manufacturer's Association (ECMA), the European Telecommunications Standards Institute (ETSI) and the International Standards Organization (ISO). These three bodies have tried to harmonize their documents with the ISO documents taking precedence. While QSIG was initially a European effort, many switch vendors have incorporated QSIG features into switches available in North America. QSIG services are not generally available as part of the public network in either Europe or North America.

Supplementary Services are usually enabled through a subscription parameter if connecting to the PSTN or through an administration procedure if connecting to a PBX. To determine which services are available and if they are enabled, it will be necessary to consult with the PSTN, the administrator, or the switch vendor.

### 4.8.1 Facility Messages

Many Supplementary Services use FACILITY messages to initiate or acknowledge an action. These messages include a Facility information element which is used to contain the information necessary for the service. Each Facility element includes a component. Components come in four types, Invoke, Return Result, Return Error, and Reject. The Invoke component is used to initiate a service. It includes an invoke identity tag which is used to identify a specific instance of an invocation. This tag is used in subsequent response messages to indicate that the response is for that specific invocation. An invoke tag is a unique value from 01h to FFh (in some cases 0h to 0FFFFh) and must not be used by another invocation until the requested service has been completed. A Return Result component is used to acknowledge an invoke request, and it may or may not include additional information. A Return Error component is used to indicate that an error has occured in the service invocation and will include an error code indicating the reason. A Reject component is used to indicate that an invoked service has been rejected, and will contain a problem code indicating the cause of the rejection. This will usually be because the facility information element can not be understood.

Each service has it's own format and argument structure, which will be described in subsequent sections. However, the HMP T1/E1 Board provides a generic form for FACILITY response messages where a more detailed format is not required. The generic format for the FACILITY message is:

**DFxxcid(tt)**

where xx is the B-channel of the call for which the service is being invoked, c is the component type, id is the invoke id tag, and tt is an optional value such as an operation type or error value. The component types are:

| | |
|---|---|
| **I** | invoke |
| **R** | return result |
| **E** | return error |
| **J** | reject |

As an example of the message sent for a return result component:

rcv: **DF00R47** return result, invoke id 47

Note that in this case no value was returned as part of the return result. If an error condition had occurred the response would have been:

rcv: **DF00E4703** return error, invoke id 47, error 3 “not available”

If a reject component was received the response would be:

rcv: **DF00J4701** reject, invoke id 47, unrecognized operation

It should be noted that due to the complexity of some of the components, the Facility elements for all services are not supported at this time. Those that are supported are detailed in the following sections.

## 4.8.2 Name Identification Service

The Calling or Connected Party Name may be part of a message if the Name Identification Presentation supplementary service is supported. This feature is described in *ANSI T1.641-1998* and Telcordia *GR-1367-*

*Core Issue 3, March 2000.* The QSIG specifications for this feature are contained in *ECMA-163* and *ECMA-164*, while the ISO documents are *ISO/IEC 13864* and *ISO/IEC 13868*.

The Name Identification information is carried in a Facility information element as part of a SETUP or CONNECT message. If present, it appears as a message of the form **DFxxNo[name]** where xx is the B-channel assigned to the call, o is a number representing the operation, and name is the string of characters. The operation values are 0 for calling name, 1 for called name, 2 for connected name, and 3 for busy name. This string is normally limited in length to 15 or 20 characters. If the Facility information element with the name is contained in a SETUP or CONNECT message, the Name will be appended to the “DS” or “DC” message after any number information.

As examples:

**DS028005SN5551234/5551212N0John Smith**
**DC030005N2Jane Doe**

To include a calling or connected name in a SETUP or CONNECT message, the name is sent in a command of the form **D@Nido[name]** where id is a hexadecimal invoke identity in the range 00-FFh, o is the operation code as defined above, and the name is the character string to be sent. This command must be sent before the “DS” or “DC message with which it is to be associated. As an example of a calling party name:

**D@N130JOHN SMITH**
**DS04SN5551234/5551212**

The Telcordia Generic Requirements specifies a slightly different form for the Facility information element than that specified by QSIG. This form can be sent using a command of the form **D@nido[name]** where all the fields are as described in the paragraph above.

### 4.8.3 Release Link Transfer

Release Link Transfer or RLT is a proprietary feature of some Nortel switches in North America. It is available on the DMS-100 NI-1 switch and some models of the DMS-250. It is not available on DMS-100 NI-2 switches (Enhanced Explicit Call Transfer may be used instead). This feature is described for the DMS-100 in the Nortel document *NIS-A211-1 ISDN Primary Rate User-Network Interface Specification.* When connecting to the DMS-100 switch, the switch level must be set to "D" with the "SL" command.

The RLT procedure consists of two steps. First, when placing a call to the number the call is to be transferred to, a facility information element with an operation of RLT Operation Indication is included in the SETUP message. If the switch decides the call can be transferred, it will return a Call ID in a facility element in either an ALERTing or PROGress message. Which message will be used depends on whether there is an interworking situation, but it will always be the first message received. This Call ID is not the same as the call reference. When the call is to be transferred, which must be after an ALERTing or CONNect message has been received for this call, a FACILITY message is sent using the call reference of the first call and a facility information element including the Call ID of the second call with an operation value of RLT Third Party. If accepted, the two parties will then be connected and the two calls will be cleared using the normal disconnect procedure.

To include the RLT Operation Indication in a SETUP message, the command **D@R** must be issued immediately before the "DS" message for the SETUP. The Call ID will be returned in a message of the form **DFxxNI<CallID>** where the Call ID is represented as six or eight hexadecimal digits. To complete the transfer a message of the form **DFxxL<CallID>** should be sent on the B-channel of the call to be transferred with the Call ID being the same digits as received. If the RLT is accepted, a FACILITY message of the form **DFxxNT** will be received. If there is an error, a message of the form **DFxxNEidee** will be received where xx is the B-channel, id is the invoke id which will be 01 if the error comes at the first step and 02 during the second, and ee is the error. Valid error codes are:

| | |
|---|---|
| 10 | RLT Bridge Fail |
| 11 | RLT Call ID Not Found |
| 12 | RLT Not Allowed |
| 13 | RLT Switch Equipment Congestion |

If the RLT is rejected, a message of the form **DFxxNJidppee** where xx is the B-channel, id is the invoke id, pp is the problem code, and ee is the error code. Normally only an invoke id of 00, a problem code of 82 and an error code of 02 is valid.

The following example shows the messaging from the TE side:

| commands | responses | description |
|---|---|---|
| | **DS00SN/5551212** | SETUP message of first call |
| **DC00** | | CONNECT for first call |
| | **DC00A8001** | CONNect ACK |
| **D@R** | | RLT Operation Indication |
| **DS01SN/5551234** | | SETUP message for second call |
| | **DP01I** | CALL PROCeeding for 2nd call |
| | **DA01I** | ALERTing message for 2nd call |
| | **DF01NI123456** | Call ID of second call |
| **DF00L123456** | | FACILITY RLT Third Party |
| | **DF00NT** | FACILITY RLT accepted result |
| | **DD0010** | DISCONNect for 1st call |
| | **DD0110** | DISCONNect for 2nd call |
| | **DR0010** | RELease COMplete for 1st call |
| | **DR0110** | RELease COMplete for 2nd call |

The HMP T1/E1 board may also emulate the NT side of an RLT. When this is done, a **FNR** appended to the end of a "DS" message will indicate the RLT Operation Indication invoke component. The Call ID is sent in a command of the form **D@I<CallID>** which must be sent immediately before the ALERTing or PROGress message for the second call. A received message of the form **DFxxNT<CallID>** will indicate an RLT Third Party invoke component. The return result component is sent with a command of the form **DFxxLR02**. Note that it is the responsibility of the application to manage the Call ID values and to clear the calls after making the transfer.

### 4.8.4 Enhanced Explicit Call Transfer

Enhanced Explicit Call Transfer (ECT), also called Two B-channel Transfer, is a mechanism to join the parties associated with the far end of two calls together while the near end parties drops out and the B-channels are released. One of the calls must be in the connect state, while the other call may be in either the connect state if an incoming call, or in the connect or alerting states if an outgoing call. This feature is only available in North America and is included in NI-2. It is described in *ANSI T1.643-1998* and Telcordia *GR-2865-Core Issue 3, March 2000.*

In a typical application, a party A places an incoming call to the interface. A call is then placed to party B using a different B-channel. This call can be in either the alerting state (ringing) or the connect state (answered). The transfer is then initiated from the interface. If the transfer is accepted, party A and party B are connected together and both calls are cleared and the B-channels released.

To invoke Explicit Call Transfer (ECT), the initiating interface sends a FACILITY message for one B-channel which contains the call reference of the call on the other B-channel as part of a facility element. If one of the calls in the alerting state, the Facility message must be sent for the B-channel of the connected call. The “D” message to do this takes the form **DFxxTidrrrr** where xx is the B-channel of one of the calls, id is an invoke identity, and rrrr is the call reference of the other call. The invoke id is a number in the range 00-7F which is used to identify the request for transfer. It is only valid until the transfer has either been granted or rejected.

In the following example, an incoming call is connected on the 02 B-channel. A second call is then made on the 06 B-channel. When that call is answered, a FACILITY message is sent to request the ECT. The request is granted and both calls are released. This example is shown from the CI side.

| commands | responses | description |
|---|---|---|
| | **DS02SN/5551234** | Incoming SETUP message |
| **DA02** | | ALERTING for 1st call |
| **DC02** | | CONNect for 1st call |
| | **DC02A8001** | |
| **DS06SN/5551212** | | SETUP message for 2nd call |
| | **DA06I0003** | ALERTing message |
| | **DC06N0003** | CONNECT message |
| **DF02T130003** | | FACILITY invoking ECT, id=13 |
| | **DD0210** | DISConnect for 1st call |
| | **DR0200** | RELease COMplete for 1st call |
| | **DD0610** | DISConnect for 2nd call |
| | **DR0600** | RELease COMplete for 2nd call |

The FACILITY message used to invoke the call transfer has an invoke id of 13 and uses the call reference of the second call, 0003. There may be a received FACILITY message with a Return Results component. If the transfer request is not accepted, there should be a Facility message with either an Error or Reject component.

If the two calls involved in the transfer use different D-channels, a different form of the command is used. This form is **DFxxTidrrrr-dcid** where dcid is the D-channel identity. The D-channel identity may be discovered by sending a message of the form **DFxxTidI** where xx is the B-channel served by the D-channel in question and id is the invoke id.

The identity will be returned in a message of the form **DFxxTidIdcid** where xx is the B-channel, id is the invoke id used in the request, and dcid is the D-channel identity. Note, as the D-channel identity is a static value, there is no need to request it for every call.

An earlier form of this feature, Explict Call Transfer is still used on some legacy switches. For this service, the command should take the form **DFxxEidrrrr.**

### 4.8.5 Call Deflection

Call Deflection is a Supplementary Service that allows an incoming call to be diverted to another destination. This is used, for example, when the called party is busy or unavailable. Call Deflection is described in *ANSI T1.642-1995*, Telcordia GR-*1310-CORE, or SR-4494 2000 NI PRI CPE Generic Guidelines Issue 1, December 1999.* It is only available in North America.

To deflect a call, the command is of the form **DFxxDidr[#]** where xx is the B-channel of the call, id is the invoke id, r is the reason, and # is the directory number of the destination the call is being deflected to. If the call deflection request is accepted, the call will be cleared in the normal manner with a DISConnect message. Valid reasons for deflection are:

| | |
|---|---|
| **B** | user busy |
| **D** | call deflection |
| **N** | no response |
| **U** | unconditional |
| **X** | unknown |

As an example, the following shows a call being deflected because the user is busy:

| command | response | decription |
|---|---|---|
| | **DS04SN/5551212** | SETUP message |

| | |
|---|---|
| **DA04** | ALERTing message |
| **DF04D07B5551234** | FACILITY message |
| **DF04R800113** | FACILITY, Return Results |
| **DD0410** | DISConnect message |
| **DR0400** | RELease COMplete message |

The FACILITY message has an invoke id of 07, a deflection reason of “user busy” and a destination number of 555-1234. A FACILITY message with a Return Results component is received indicating that the call deflection request has been accepted.

### 4.8.6 QSIG Call Diversion and Call Forwarding

QSIG Call Diversion and Call Forwarding (QSIG-CF) are services that allow a user to send calls for that user to another number. The calls may be forwarded unconditionally (CFU), if the user is busy (CFB), if there is no answer at the user (CFNR - Call Forwarding No Reply), or on a call by call basis (CD - Call Deflection). The ECMA documents describing these services are *ECMA-173* and *ECMA-174* and the ISO documents are *IS0/IEC 13872* and *ISO/IEC 13873*.

With the Call Forwarding service, a call is placed from the originating party to the diverting user. This is called Leg 1. A call can then be placed from the PBX serving the diverting user to the forwarding destination which is called forward switching, or it can be placed directly from the PBX serving the originating party which is called rerouting. In either case, this is called Leg 2. The connection from the destination to the originating party is called Leg 3. The messages involved in call forwarding inform the calling and destination users that diversion has occurred.

From the viewpoint of the HMP T1/E1 Board, there are two cases, either a call is diverted to the board, or a call originating on the board is diverted. It is assumed that the board is acting as a CI connected to a PBX.

In the case where a call is diverted to the board, there will be a SETUP message containing a Facility information element with diverting Leg 2 information. This will be indicated by a string of the form **F2cr#(/name)** being appended to the "DS" SETUP response message, where xx is the B-channel, c is the diverting count, i.e. the number of times the call has been diverted, r is the diversion reason, # is the number the call was diverted from, and /name is an optional diverting party name. Diversion reasons can be "B" for busy, "D" for deflected, "N" for no answer, or "U" for unconditional, that is all calls are being forwarded. The application should respond with a diverting Leg 3 information command message of the form **DFxxF3id** where xx is the B-channel of the SETUP message, and id is an invoke id.

To include a diverting Leg 1 Facility information element in a SETUP message, a command of the form **DFxxF2idcr#** should be issued immediately before the "DS" command. In this command, the id is the invoke id, c is the diverting count, r is the diversion reason and # is the number the call was diverted from.

In the case where a call originating from the board is diverted there will be two response messages. The first will be a diverting Leg 1 information FACILITY message of the form **DFxxF1ro#** where xx is the B-channel of the call, r is the diversion reason, o is the subscription option and # is the number the call was diverted to. Subscription option values are "0" for no notification, "1" notification without diverted to number, and "2" for notification with diverted to number. The second response message will take the form **DFxxF3** indicating a diverting Leg 3 information FACILITY message on B-channel xx.

### 4.8.7 QSIG Call Transfer by Join

QSIG Call Transfer by Join (QSIG-CT) is a service that allows a PBX to notify another PBX that it has joined two separate calls into a single call. The notified PBX can then inform the user that a join has taken place by updating a display or through other means. The ECMA documents describing this service are *ECMA-177* and *ECMA-178* and the ISO documents are *ISO/IEC 13865* and *ISO/IEC 13869*.

In call transfer by join a call is placed from party A to party B with the two parties in different switches. A call is then placed from party B to party C which is in a switch other than the one serving party B. When the transfer takes place, A and C are connected together and B drops out of both calls. Note that both paths from A and C to the switch serving B remain.

To notify the switches serving A and C that a transfer has taken place and they are now connected together, a FACILITY message is sent to each switch with a Facility information element with the ctComplete operation. This element contains the new party number that the other party is now connected to, that is, the message to the switch serving A contains C's number and vice versa. The switch can then inform the parties of the number using a display or other means.

To send a ctComplete FACILITY message a command is issued of the form **DFxxCide#**. where xx is the B-channel of the call, id is an invoke id, e is the end designator and # is the number. The end designator can be either 'P' for Primary which would be A in the above example, or 'S' for Secondary which would be C. The default coding for the number is as an publicPartynumber, nationalNumber. In some cases where the number is an internal extension this may need to be encoded as a privatePartyNumber, pISNSpecificNumber. This can be done by preceding the number with a 'P', for example, "P1234".

If party C has not yet answered, an 'A' should be added to the end of the message to the primary end to inform the switch that the call is in the alerting state. In this case, a ctActive FACILITY message should be sent to the primary end when party C goes to the connect state. This is done with a message of the form **DFxxAid#** where xx is the B-channel of the primary end call, id is the invoke id, and # is the party number of C.

Some switches may send a ctUpdate Facility message after receiving a ctComplete. The transferring switch should forward this message on to the other leg of the transfer with identical information. The ctUpdate message takes the form **DFxxUid#** where id is the invoke id and # is the redirection number. This message takes the same form for both the command and response.

### 4.8.8 QSIG Path Replacement

QSIG Path Replacement is a service which allows a connection involving multiple PBXs to be replaced with a simpler path. In particular, a connection created by a Call Transfer by Join which joins two calls to another PBX can be replaced by an internal connection in the PBX which contains the two end points. This situation is sometimes called a "trombone" connection for obvious reasons. The PBX with the end points will recognize that a trombone connection has been created using a stimulus such as receiving two ctComplete FACILITY messages with complementary redirecting numbers. It will then initiate the path replacement by sending FACILITY messages with a pathReplacePropose operation.

This facility message will take the form **DFxxPid,c-#** where xx is the B-channel, id is the invoke id, c is the call identity, and # is the redirection number. Note that the call identity can be more than one digit long. The PBX that joined the two calls should respond by echoing the pathReplacePropose message on the other call to signal acceptance. The initiating PBX will then create an internal connection and clear the calls. The default type for the redirection number is unknown. A letter preceding the number indicates a different type. The letter "N" indicates a public National number, while an "L" or a "P" indicate a private local or private PISN Specific number type.

The following is an example of a Call Transfer by Join followed by a Path Replacement as seen from the perspective of the joining PBX:

| commands | responses | description |
|---|---|---|
| | **DS008003SN1234/1000** | SETUP for primary call |
| **DA00** | | ALERTing message |
| **DC00** | | CONNect message |
| **DS02SN1234/1212** | | SETUP for secondary call |
| | **DA02I0001** | ALERTing message |
| **DF00C12P1212A** | | ctComplete, primary call |
| **DF02C13S1234** | | ctComplete, secondary call |
| | **DC020001N** | CONNect message |
| **DF00A141212** | | ctActive message |
| | **DF00P15,47-1234** | pRPropose message |
| **DF02P15,47-1234** | | pRPropose message |
| | **DR00800310** | RELease COMplete |
| | **DR02000110** | RELease COMplete |

### 4.8.9 QSIG Call Transfer by Rerouting

QSIG Call Transfer by Rerouting is another method of accomplishing a transfer on PBXs that support the service. The ECMA documents describing this service are *ECMA-177* and *ECMA-178* and the ISO documents are *ISO/IEC 13865* and *ISO/IEC 13869*.

The scenario for Call Transfer by Rerouting is similar to that for Call Transfer by Join and Path Replacement, except that the Rerouting is initiated by the local switch. This is done by the local switch requesting a call identity for the call placed to party C. This call identity is then used to initiate the rerouting.

The call identity is obtained by sending a FACILITY message using the call reference of the call to party C and an invoke component with a ctIdentify operation. This is done with a command of the form **DFxxIid** where xx is the B-channel of the call to party C and id is the invoke identity. The remote switch will send a FACILITY message with a result component using the same invoke id. The result component will contain the call identity as well as the redirecting party number. This will be in a response message of the form **DFxxIid,c-#** where xx is the B-channel, id is the invoke identity, c is the call identity and # is the redirecting party number. The call identity is typically several digits long. The rerouting is then initiated using a command of the form **DFxxRid,c-#** where xx is the B-channel, id is the invoke identity, and c and # are the call identity and redirecting party number obtained using the ctIdentify command.

The following is an example of a Call Transfer by Rerouting from the perspective of the local PBX:

| commands | responses | description |
|---|---|---|
| | **DS008002SN1234/1000** | SETUP for primary call |
| **DA00** | | ALERTing message |
| **DC00** | | CONNect message |
| **DS02SN1234/1212** | | SETUP for secondary call |
| | **DA020001I** | ALERTing message |
| | **DC020001N** | CONNect message |
| **DF00I14** | | ctIdentify message |
| | **DF00I14,47-1234** | ctIdentify result message |
| **DF02R15,47-1234** | | ctInitiate message |
| | **DR00800210** | RELease COMplete |
| | **DR02000110** | RELease COMplete |

## 4.8.10 QSIG Single Step Call Transfer

QSIG Single Step Call Transfer (QSIG SSCT)is a service that allows a user to transfer a call to another user without first placing a second call to the transferred user. It is a service intended for PBXs rather than for PSTN connections, The ECMA documents describing this feature are *ECMA-299* and *ECMA-300* and the ISO documents are *ISO/IEC 19459* and *ISO/IEC 19460*.

As it's name implies, the QSIG SSCT service involves a single step. It is initiated by sending a FACILITY message with the call reference of a connected call which includes an invoke component with an ssctInitiate operation with the number of the transferred-to party. If accepted, the switch places a call to this party. Two modes are allowed. In the first the call is transferred upon the switch receiving an ALERTing message, while in the second mode the transfer does not take place until a CONNect message is received.

To initiate a transfer, a command of the form **DFxxSidm#/#** is issued, where xx is the B-channel of the call to be transferred, id is the invoke id, a two digit hexadecimal number, m is the mode, either 'A' if the transfer is made upon receiving an ALERTing message or 'C' if it is made upon receiving a CONNect message, the first '#' is the transferred-to parties number, and the second '#' is the number of the transferred call that is to be presented to the transferred-to party.

If the transfer is successful, then a DISConnect message will be received by the transferring user to clear the call. If the transfer is not accepted, a message of the form **DFxxEidee** will be received where id is the invoke id in the command initiating the transfer and ee is an error code.

### 4.8.11 QSIG Message Waiting Indication

The QSIG Message Waiting Indication allows a device such as a voice mail server to control an indication for a user on a PBX so that the user may know if there are any messages waiting for that user. The form of the indicator may be a display, a distinctive dial tone, or other means. The ECMA documents describing this service are *ECMA-241* and *ECMA-242* and the ISO documents are *ISO/IEC 15505* and *ISO/IEC 15506*.

Controlling Message Waiting Indicators involves placing a call to the PBX serving the user. The SETUP message will include a Facility information element to activate or deactivate the indicator. The Bearer Capability should be set to ECMA unrestricted digital data and no B-channel should be assigned. No audio path is necessary for this call.

Immediately before sending the "DS" command for the SETUP message, a command message of the form **D@Mido#(/#)(=m)** should be sent to construct the Facility information element, where id is the invoke id, o is the operation, either 'A' to activate or 'D' to deactivate the message waiting indicator, and # is the user number. Optionally this message may contain a number for the message center which will take the form /#, and/or '=m' where m is the number of messages waiting. The SETUP message should then be sent. The command to do this takes the form **DSxxEN**#/# where xx is the B-channel, the first '#' is the calling party number and the second '#' is the called party number which corresponds to the hunt group of the user. Note that to select a channel id of no B-channel, the B-channel should be set to a channel one more than the highest channel on that interface, i.e. 17 for interface 0, 37 for interface 1 etc.

The called switch should respond with a CALL PROCeeding and CONNect message as normal. The CONNect message should include a Facility information element with a Return Result component to acknowledge the operation. The call should then be cleared in the normal way.

As an example of activating call waiting for station 1234 which belongs to hunt group 1000:

| command | response | description |
|---|---|---|
| **D@M03A1234** | | build Facility i.e. |
| **DS17EN/1000** | | SETUP message |
| | **DP17N0004** | CALL PROCeeding message |
| | **DC17N0004** | CONNect message |
| | **DF17R03** | FACILITY, Return Results |
| **DR17R** | | RELease message |
| | **DR1700** | RELease COMplete |

# 4.9 Layer 3 Maintenance Messages

Several Layer 3 messages are provided for maintenance purposes. These consist of the RESTART message which is used to reset a B-channel or interface and the SERVICE message which is used to mark a B-channel as available or unavailable.

## 4.9.1 RESTART & RESTART ACKNOWLEDGE

The Restart Procedure is used to reset a B-channel or interface to insure that both sides of the interface are in agreement as to the state of B-channel or interface by placing the B-channel or channels in the null state. This is done if there has been some indication that there has been a problem which can not be resolved by the normal clearing procedures. It is sometimes done upon recovery of a persistent Layer 1 error.

The side receiving a RESTART message should clear the indicated B-channels and send a RESTART ACKnowledge message. Upon receiving a RESTART ACKnowledge message the initiating side should clear the indicated B-channels. The Restart procedure is symmetrical and can be sent by either the NT or CI side.

A RESTART message can be send by issuing a command of the form **DZxx** to reset a interface, **DZxxC** to reset the indicated B-channel, or **DZxxA** to reset all the interfaces associated with the D-channel in an NFAS arrangement.

The reception of a RESTART message is indicated by a response message of the form **DZxxA** if all interfaces associated with the D-channel are to be reset, **DZxxC** if only the indicated B-channel is to be reset, or **DZxxI** if all B-channels on the interface starting with the indicated B-channel are to be reset.

The reception of a RESTART ACKnowledge message is indicated by response messages of the form **DZxxAA**, **DZxxAC**, or **DZxxAI**, for reset all interfaces, reset a specific B-channel, or reset B-channels on a interface starting with the indicated interface, respectively.

It is not necessary for the application to take any action to clear the B-channels on the board. Of course, the application must take any internal actions that are necessary to reflect the fact that the B-channels have been reset.

### 4.9.2 SERVICE & SERVICE ACKNOWLEDGE

In North America the SERVICE message is used to indicate B-channel availability or to indicate a D-channel switchover in NFAS arrangements where there is a backup D-channel.

To initiate a change in the status of a channel or channels, a SERVICE message is sent. The other side of the interface replies with a SERVICE ACKnowledge message. A channel can be marked as being in service, out of service or in a maintenance state. Details of the procedures associated with these messages can be found in Annexes F and U of *ANSI T1.607.*

The command to change the status of a B-channel takes the form **DMxxsC** for an individual B-channel, or **XSxxs** for an entire interface where xx is the B-channel and s is the state. For a D-channel the command takes the form **DMxxsD** where xx is the first B-channel on the interface and s is the state. Allowed values for the state are “I” for in service, “O” for out of service, and “M” for maintenance.

The response message indicating a SERVICE message has been received takes the form **DMxxsC** for a single B-channel, **DMxxs** for all B-channels on a interface, and **DMxxsD** for a D-channel. The SERVICE ACKnowledge message will automatically be sent by the board. The response message indicating a SERVICE ACKnowledge message has been received takes the form **DMxxAsC** for a single B-channel, **DMxxAs** for all B-channels on a interface, and **DMxxAsD** for a D-channel.

It is the responsibility of the application to mark the availability of B-channels and to avoid initiating calls on channels that are out of service. It is also the responsibility of the application to ensure that the correct D-channel is used on an NFAS arrangement by sending the appropriate values in the “SN” command.

It should be noted that the protocol discriminator used for SERVICE messages is not 08h indicating a Q.931 message. A protocol discriminator of 43h is specified by *ANSI T1.607* and the Telcordia NI-2 specifications. However, some pre NI-2 switches use a protocol discriminator of either 01h (some early AT&T switches) or 03h (Nortel pre NI-2). When connecting to these switches the switch level in the “SL” command should be set to “A” or “D” respectively. For NI-2 switches it should be set to “N”.

## 4.10 Non-Facilities Associated Signaling (NFAS)

When connecting multiple Primary Rate ISDN spans to the public network, it may be advantageous to use a single D-channel to control multiple spans. This is called Non-Facilities Associated Signaling or NFAS, as call control is handled by a D-channel not associated with the span carrying the call. This feature is only available in North America.

To enable NFAS on a set of spans one span is selected as the primary span. This span will have a D-channel and the framing is set in the same manner as a normal PRI span. The framing for the other spans must be set for no signaling, i.e. “SFddEBbN” where dd is the span, and b is the buildout code. This will free up the twenty-fourth timeslot on the span for call handling. If the NFAS arrangement is to include a backup D-channel, the span with the backup must also have its’ framing set for PRI signaling. However, the D-channel will not be used until backup procedures are initiated.

If Layer 3 call control messaging is to be handled by using the "LC" message procedure, nothing more needs to be done to set up the spans. However, the Channel Identity information element will need to include the appropriate interface identifier information.

If call control is being handled through "D" messages, i.e. the Protocol Level support is set other than "2", then there is an additional step to setting up the spans. This is done through a command of the form **SNic..ic** where each pair of digits ic is used to control one span. There will need to be one pair for each of the spans on the board. The 'i' digit is used to indicate which interface identifier is associated with the span. The 'c' digit is used to indicate the span carrying the D-channel for the set of NFAS spans. Bit 3 of this digit must be set to indicate that the span is part of an NFAS arrangement. If a span is not part of an NFAS arrangement, it should be coded as '00'. As an example the following command sets up three spans as an NFAS arrangement with the first span carrying the D-channel. The other spans are not part of NFAS arrangements.

**SN0818280000000000**

If D-channel backup procedures are invoked, this command will need to be issued reflecting the new D-channel.

## 4.11 Passive Monitoring

The HMP T1/E1 Board has the capability of passively monitoring a T1 or E1 circuit. When an interface is set to one of the monitor modes, Layer 3 messages on the circuit will be reported in the same manner and format as for an interface set for the NT or TE modes. Note, that because an monitoring interface intercepts the circuit in only one direction, it will only report Layer 3 messages in that direction. For full duplex monitoring, two interfaces are required and it is the responsibility of the application software to correlate the messages on the two interfaces. Also, when set to a monitoring mode, an interface can not be used to send Layer 3 messages.

this page intentionally left blank

# 5.0 Robbed-Bit & Channel Associated Signaling

In addition to Primary Rate ISDN, there are other methods of signaling available on T1 and E1 circuits. These methods predate ISDN signaling and were designed to emulate the behavior of analog circuits. In North America, the scheme used on T1 circuits is called Robbed-Bit Signaling. For E1 circuits, a method known as Channel Associated Signaling is used.

## 5.1 Robbed Bit Signaling

Robbed bit signaling is used on non-ISDN T1 circuits for call control. It allows each channel on the span to emulate a variety of analog line types. As its name implies, it robs bits from each channel for call control information, the least significant bit from every eighth frame. Use of robbed bit signaling does result in a loss of bandwidth, that is only 56 kbps are available instead of 64 kbps. However, as the bit in question is the least significant bit, it does not cause a noticeable degradation for voice circuits. For a D4 frame, two bits are available, called A and B. For ESF, two additional bits are available, C and D. However, in most implementations, the C and D bits repeat the A and B bits.

These bits can be used to indicate hook status, ringing, and ground signaling. By using them, a channel can emulate the operation of an E&M, ground start, or loop start circuit. For ground start and loop start circuits, the two ends are not symmetric, that is one end emulates is the central office and the other end emulates a station set or telephone instrument. Thus which end of the circuit a channel is emulating must be specified. When using robbed-bit signaling, the line type must therefore be set for each channel. This is done using a command of the form **SPxxtd(p)** where xx is the channel number, t is the line type, d is the direction, and p is an additional protocol parameter for address signaling protocols that may be supported (see Sec. 5.4). Line types that are supported are “E” for E&M, “G” for ground start, “L” for loop start and “N” for none. When none is selected, robbed bit signaling is not implemented. The direction specifies the end and can be either “O” for the central office end, or “S” for the station end. The p parameter allows for the support of immediate start “I”, or wink start “W” protocols.

## 5.2 Channel Associated Signaling

Channel associated signaling is used on non-ISDN E1 spans for call control. It allows each channel on the span to emulate a variety of analog line types. As its name implies, signaling bits are associated with each channel for call control information. These bits are carried by timeslot 16 of each frame within the multiframe. Frame 0 contains a pattern to indicate the multiframe, frame 1 contains the bits for channels 1 and 2, and so on. Two bits are available for signaling, called A and B. Two additional bits are available, C and D, are used to insure that timeslot 16 is not all zeros. Normally the C bit is a “0” and the D bit is a “1”.

These bits can be used to indicate hook status, ringing, and ground signaling. By using them, a channel can emulate the operation of an E&M, ground start, or loop start circuit. For ground start and loop start circuits, the two ends are not symmetric, that is one end emulates the central office and the other emulates a telephone instrument or station set. Thus which end of the circuit a channel is emulating must be specified. When using CAS signaling, the line type must therefore be set for each channel. This is done using a command of the form **SPxxtd(pd)** where xx is the channel number, t is the line type, d is the direction, and p is an additional parameter for address signaling protocols that may be supported (see Sec. 5.4). Line types that are supported are “E” for E&M, “G” for ground start, “L” for loop start, “Q” for Q.421 signaling (as defined in ITU Q.421), and “N” for none. When none is selected, channel associated signaling is not implemented. The direction specifies the end and can be either “O” for the central office end, or “S” for the station or CPE end. The p parameter allows for the support of immediate start “I”, or wink start “W” protocols.

## 5.3 Controlling a Channel

There are two ways in which an application can control the signaling bits. The first method is direct control where the application assumes all responsibility for setting and interpreting the signaling bits. In the second method, the application uses a state machine managed by the on board processor. This method simplifies the application software.

### 5.3.1 Direct Control

In the direct control method, the signaling bits are set using a command of the form **CZxxb** where xx is the channel number and b is a hexadecimal number representing the bits, with the A bit being the most significant and the D bit the least. For example, the command “CZ02B" would set the signaling bits for channel 02. The bit values would be: A = 1, B = 1, C = 0, D = 1.

Changes in the received signaling bits are reported in a response message of the form **SZxxb** where xx is the channel number and b is a hexadecimal number representing the bits as defined in the previous paragraph.

To enable a channel for direct control, the channel protocol should be set to XS using the “SP” command. As an example, the command “SP02XS” would set channel 02 to direct control.

It should be noted that when using CAS signaling on an E1 interface, bit

D should always be set to a 1 to insure proper framing.

### 5.3.2 Using the State Machine

In the state machine method, the board firmware manages a state model of a channel and sets the signaling bits to the appropriate values depending on both the state and the type of circuit the channel has been set to emulate. This simplifies the task of the application as some of the timing and signaling issues are handled by the board and not the application.

Using the "SP" command, each channel can be set to emulate a number of different analog line types. These include loop start, ground start, E&M, and Q.421. With the exception of the E&M line type, these line types are not symmetric, that is the two ends of the channel behave differently, with one end acting as the central office and the other end acting as a station set. In the "SP" command, which end a channel is to emulate is specified by the characters "O" for office or "S" for station. For example, the command "SP02LS" would set channel 02 to emulate the station end of a loop start circuit. Valid channel settings are:

**EO** E&M
**ES** E&M
**GO** Ground start, central office
**GS** Ground start, station
**LO** Loop start, central office
**LS** Loop start, station
**QO** Q.421, central office
**QS** Q.421, station

Q.421 is a CAS signaling standard defined in the ITU document Q.421. Note that because E&M emulations are symmetric, the behavior of "EO" and "ES" channels is identical.

The state machine defines a number of states and the signaling bit value associated with the state. Transitions between the states are made in response either to commands or changes in the received signaling bits. The states are:

| | |
|---|---|
| idle | the channel is inactive |
| hold | the channel is off-hook |
| connect | the channel is off hook with audio |
| ringing | ringing has been detected on the channel |
| bell | ringing is being generated by the channel |
| off hook | an off hook has been detected by the channel |
| off hook idle | the channel has returned to on hook, but the far end is still off hook |

Note, that the signaling bit values for different emulation types in the same state may be different. Some states are only applicable to specific emulation types. For example, the ringing state is only possible for the loop start and ground start station types, while the bell state is only possible for the loop and ground start central office types.

Control of the signaling is done by means of a command of the form:

**C**cxx

where c is a letter specifying the command and xx is the channel number. Valid commands are:

| | |
|---|---|
| **CA** | audit, off hook, one way audio to the channel |
| **CB** | busy, this checks to see if the channel is idle, if it is, it sets the signaling bits to off hook |
| **CC** | connect, off hook, two way audio |
| **CD** | disconnect, on hook, no audio |
| **CF** | flashhook, generate a timed hook flash |
| **CH** | hold, off hook, no audio |
| **CR** | ringing, signal ringing, LO and GO only |
| **CX** | transmit, off hook, audio from the channel |

Changes in the state caused by a command or changes in the received signaling bits will be reported in state change response messages. These messages take the form **Ssxx** where s is the state change and xx is the channel number. The state change messages are:

**SB** the channel is not in the idle state, response to CB
**SC** the channel is in the connect state, response to CC
**SF** a receive off hook has been detected
**SH** the channel is in the hold state, response to CH or CB
**SI** the channel is in the idle state, response to CD or on hook
**SL** the channel is in the listen state, response to CA
**SN** a received on hook has been detected
**SR** a received ringing signal has been detected
**SX** the channel is in the transmit state, response to CX
**SW** a received wink has been detected

It should be noted that the listen, transmit, and connect states reflect the state of the signaling bits only and do not reflect whether audio is being sent to or from the channel. They have been included for compatibility with earlier Amtelco products.

### 5.3.3 Address Signaling Protocols

When emulating analog interfaces, address information, that is the destination or source numbers of the calling parties, is sometimes sent using DTMF or MF signaling on the channels of the T1 or E1 circuit. While the HMP T1/E1 board does not have the facilities to detect these signals, some of these address protocols require an acknowledgment signal at initiation of the call. This acknowledgment signal takes the form of a “wink,” that is a short off-hook/on-hook transition. The timing of this wink is critical, the wink itself is typically 140-300 msec. long and must begin within 70 msec. of the off-hook signal initiating the call. To relieve the application of the burden of handling the wink, the firmware can be programmed to generate the wink in response to a received off-hook signal.

To program a channel for this function, a “W” is added to the end of the “SP” command. As an example, “SP02ESW” will set channel 02 so that it generates a wink signal. Setting a channel to “wink” will also program that channel to expect a wink signal when initiating a call.

When an off-hook signal is received in the idle state, the timing for the wink will begin. An “SF” message will also be sent to indicate that the far end has gone off-hook. At this point it is the responsibility of the application to initiate the mechanism to detect the address digits.

As an example of this, the following messages might be sent:

| | | |
|---|---|---|
| **rcv:** | **SN02** | the off-hook is detected |
| | | the application detects the digits in the audio |
| **xmt:** | **C**C02 | the channel goes off-hook to indicate an answer |
| **rcv:** | **SC02** | the answer is acknowledged |
| **xmt:** | **CD02** | the channel goes on-hook to indicate a disconnect |
| **rcv:** | **SI02** | the channel returns to the idle state |

When originating a call when a wink is expected, a message will be sent when the wink has been detected. This will be in a message of the form **SWxx** where xx is the channel number. If a wink is not detected within 800 msec. an error message will be sent which takes the form **SWxxE** where xx is the channel number.

this page intentionally left blank

# 6.0 Diagnostics & Event Messages

## 6.1 Diagnostic Commands

Several diagnostic commands are available:

**VA** Used to request the checksum of the firmware in the alternate segment of the board. This is returned in a message of the form `VAxxxx` where `xxxx` is the checksum of the firmware in the alternate segment of ROM.

**VC** Used to request the version of the firmware on the board. A message of the form `VCxxxxyyyyHTA8` is returned, where `xxxx` is the checksum of the firmware stored in the main segment of ROM, `yyyy` is a four-digit version number, `HTA8` indicates the board type and version (HMP T1/ E1, board revision A, 8 interfaces). For four interface boards the board type is `HTA4`.

**QA** Queries the alarm status of the T1/E1 interfaces. The response takes the form `QAaaaaaaaa` where each `a` represents the alarm condition of one of the interfaces starting with interface 0. The values of `a` are “G”, “Y”, “B”, or “R” for green (no alarm), yellow alarm, blue alarm, and red alarm respectively. An inactive interface will be indicated by an “N”.

**QLss** Queries the receive signal level on interface ss. The response takes the form `QLssl` where ss is the interface and l is the level value in -2.5 dB steps.

**QRxx** Queries the call reference value and call state of the call currently assigned to B-channel `xx`. The results are returned as `QRxxrrrrss` where `xx` is the B-channel, rrrr is the call reference value, and `ss` is the call state. This command is only valid for Primary Rate ISDN interfaces.

## 6.2 Event Messages

The board will detect a number of error conditions and respond with appropriate error messages. These messages are:

**ABdd** A Blue Alarm (AIS) event for interface `dd`. An all-ones Alarm Indication Signal has been detected.

**AGdd** A Green Alarm event for interface `dd`. All alarm conditions have been cleared.

**AMdds** A Distant MF Alarm event for interface dd. A Distant MF Alarm condition has been detected if s is 'S' or cleared if s is 'C'. A Distant MF Alarm occurs when bit 6 of timeslot 16 in frame 0 has been set for two consecutive MF frames. E1 CAS circuits only.

**ARdd** A Red Alarm event for interface `dd`. A Loss of Signal (RLOS) has been detected

**ASdd** A Receive Signaling All Ones event has been detected on interface dd when over 16 consecutive frames fewer than three 0's are detected. E1 CAS circuits only.

**AYdd** A Yellow Alarm event for interface `dd`. A Remote Alarm Indication (RAI) has been detected.

**DExxe** A Layer 3 timer event has occurred on B-channel xx causing actions as defined in Q.931. The specific timer event is indicated by the value of e:

- 0: T303 has expired for the first time
- 1: T303 has expired for the second time
- 2: T310 has expired
- 3: T305 has expired
- 4: T308 has expired for the first time
- 5: T308 has expired for the second time
- 6: T316 has expired
- 7: T322 has expired

**ELdde** An error has been detected in either Layer 1 or Layer 2 for interface `dd`. The specific error is indicated by the value of `e`:

- 1: Timer T1 has expired indicating a persistent Layer 1 error condition on interface dd
- 2: Timer T2 has expired indicating a persistent Layer 1 error condition recovery on interface dd
- 5: A Layer 2 N(R) state variable recovery has taken place

**EMdde** A Layer 2 protocol error has occurred on interface `dd`. The error is indicated by the value of `e`. This corresponds to the Management Data Layer Indication in parentheses as defined by Q.921:

- 1: MDL error (A) unsolicited supervisory response (F=1)
- 2: MDL error (B) or (E) unsolicited DM response
- 3: MDL error (C) or (D) unsolicited UA

response
4: MDL error (F) peer initiated reestablishment (SABME)
5: MDL error (G) or (H) unsuccessful retransmission of SABME or DISC
6: MDL error (I) unsuccessful retransmission of a status enquiry
7: MDL error (J) N(R) state variable error
8: MDL error (K) receipt of FRMR response

**ESddss** A Line Interface Unit status event has occurred on interface `dd` with the event given by `ss` where the bit values are:

1: Transmit open circuit detected
2: Transmit current limit exceeded
3: Receive carrier loss detected
4: Jitter attenuator limit trip event

**EXddss** An Elastic Store Buffer event has occurred on interface `dd` with the event given by `ss` where the bit values are:

0: Receive elastic store slip event
1: Receive elastic store empty event
2: Receive elastic store full event
3: Transmit elastic store slip event
4: Transmit elastic store empty event
5: Transmit elastic store full event

**SM0** An EEPROM operation has failed. This indicates that either a read or write to the EEPROM was unsuccessful.

**SM1** An EEPROM operation successfully completed.

**SWxx** A wink signal has been detected on B-channel xx.

**SWxxE** An expected wink signal was not received within 800 msec. for an outgoing call on B-channel xx.

**U*[cmnd]*** If the board does not recognize a command message, or if it does not have the appropriate number of arguments, the same message will be returned by the board preceded by a U to indicate an undefined message.

## 6.3 Diagnostic Tests

Several diagnostic tests can be run on the E1/PRA interfaces to check the operation of the interface or the attached wiring. The commands to initiate these tests are:

**XCxxc** This command sends the loopback code c out on interface xx (NT interfaces only). The code bit values are:

0: no code
1: request loopback 2 activation
2: request loopback 1 deactivation

**XFdd** Sets diagnostic control flags to dd. When set to 1, the diagnostic function is enabled. The functions for each bit are:

0: enables PRM message reporting (ESF T1)
1: enables PRM message transmission
2-7: reserved

**XLddm** This test puts interface `dd` in the loopback mode specified by `m`:

- 0: release loopbacks
- 1: local loopback (TE)/remote loopback 2 (NT)
- 2: framer loopback
- 3: payload loopback (all channels)
- 4: payload loopback (specific channels)
- 5: analog loopback
- 6: remote loopback 1
- 7: dual loopback (local and remote 2 loopback)

**XLdd4aabbcc**ee This test is used to generate a payload loopback for specific channels on span `dd`. The hexadecimal arguments `aa`, `bb`, and `cc` are used to specify which channels (0-7, 8-15, 16-23, and 24-31 respectively) are to be looped back. The lowest channel in each argument is the least significant bit.

The following test messages may be received during tests:

**XLddc** Loopback code `c` received on T1-D4 interface `dd`, code values are:

- 0: end of code
- 1: activate line loopback
- 2: deactivate line loopback

**XLddc** Loopback code `c` received on E1 interface `dd`, code values are:

- 0: release loopbacks
- A: activate loopback 2
- F: activate loopback 1

**XLddcc** Bit patterned T1-ESF data link message `cc` detected on interface `dd`. (See Section 2.6.2 and *T1 403-1999* Table 4 for code values.)

**XXdd(data)** A Performance Report Message has been received on span `dd`. The data will consist of four 4 digit hexadecimal numbers representing the values of error bits in the PRM message. (See Section 2.6.5 and *T1 403-1999* Fig. 6 for bit value meanings.)

this page intentionally left blank

# Appendix A: Message Summary

## Messages to the Board

### Alarm Commands

| | |
|---|---|
| **ABdde** | Blue alarm (AIS) event on dd, e = C - clear, S - set |
| **AMdde** | Distant MF Alarm on dd, e = C - clear, S - set |
| **ASdde** | Signaling all 1's on dd, e = C - clear, S - set |
| **AYdde** | Yellow alarm (RAI) on dd, e = C - clear, S - set |

### B-Channel Commands

| | |
|---|---|
| **CAxx** | Set B-channel xx to listen |
| **CBxx** | Set channel xx to hold if not busy, else return SBxx |
| **CCxx** | Connect B-channel xx |
| **CDxx** | Disconnect B-channel xx |
| **CFxxd** | Generate a hook flash on channel xx of duration d |
| **CHxx** | Put B-channel xx on hold |
| **CRxx** | Generate ringing on channel xx (robbed bit) |
| **CXxx** | Set B-channel xx to transmit |
| **CZxxb** | Set CAS bits b for B- channel xx |

### Layer 3 "D" Commands for NT Spans

| | |
|---|---|
| **DAxx(p)** | ALERTing message on B channel xx, opt. progress p |
| **DCxx(p)(#cnct)** | CONNect message B channel xx, opt. progress p, optional connected #, opt. party category pc |
| **DCxxA** | CONNect ACK message on B channel xx |
| **DDxx** | DISConnect message B channel xx, normal clearing |
| **DDxxcc(p)** | DISConnect message on B channel xx, cause cc, optional progress indicator p |
| **DFxxAid#** | FACILITY, ctActive, id - invoke id, # - redirecting number |
| **DFxxCide#(A)** | FACILITY, ctComplete, id - invoke id, e - end designator, # - redirection #, opt. A if alerting |

| | |
|---|---|
| **DFxxF1idro#** | FACILITY, divertingLegInformation1, id - invoke id, reason - r, o - subscription option, # - diverting to number |
| **DFxxF3id** | FACILITY, divertingLegInformation3, id - invoke id |
| **DFxxIid** | FACILITY, ctIdentify invoke, id - invoke id |
| **DFxxIid,c-#** | FACILITY, ctIdentify results, id - invoke id, c -call identity, # redirecting party number |
| **DFxxN(n)** | FACILITY on xx, optional notification indicator n |
| **DFxxPid,c-#** | FACILITY on xx, pathReplacePropose, id - invoke id, c - call identity, # - redirecting # |
| **DFxxRid,c-#** | FACILITY, ctInitiate, id - invoke id, c - call identity, # - redirecting party number |
| **DFxxSidm#/#** | FACILITY (SSCT) on B-channel xx, id - invoke id, m - mode, # - transfer to number, /# - transferred # |
| **DFxxUid#** | FACILITY, ctUpdate, id - invoke id, # redirection number |
| **DNxxR** | NOTIFY, call resumed |
| **DNxxS** | Notify, call suspended |
| **DNxxnn** | NOTIFY message, notification indicator nn |
| **DPxx(p)** | CALL PROCeeding message on B channel xx, optional progress indicator p |
| **DPxxPccp** | PROGress on B channel xx, cause cc, prog. ind. p |
| **DRxxcc** | RELease COMplete message on B channel xx, cause cc |
| **DRxxRcc** | RELease message on B channel xx, cause cc |
| **DSxxbp(#)/#(Rr#)** <br> **DSxxbp@(#)/#(Rb#)** | SETUP message, bearer capability b, progress ind. p, @ = sending comp., optional calling #, called #. optional redirect # |
| **DTA(text)** | Add display text to buffer |
| **DTC** | Clear display text buffer |
| **DXxx** | Status Query for B-channel xx |
| **DZxx** | RESTART interface starting at B channel xx |

| | |
|---|---|
| **DZxx(C)** | RESTART interface, indicated B channel xx |
| **D@F2idcr#** | Put divertingLegInformation2 in buffer, id - invoke id, c -diversion count, r - reason, # diversion number |
| **D@Nido<name>** | Put name in buffer, id - invoke id, o - operation |

**Layer 3 "D" Commands for TE Spans**

| | |
|---|---|
| **DAxxp(rrrr)** | ALERTing message on B channel xx, progress ind. p, call reference rrrr optional |
| **DCxxp(#cnct)** | CONNect message B channel xx, opt. progress indicator p, opt. connected # |
| **DCxxA** | CONNect ACK message on B channel xx |
| **DDxx** | DISConnect message B channel xx, normal clearing |
| **DDxxcc(p)** | DISConnect message on B channel xx, cause cc, opt. progess p |
| **DFxxAid#** | FACILITY, ctActive, id - invoke id, # - redirecting number |
| **DFxxCide#(A)** | FACILITY, ctComplete, id - invoke id, e - end designator, # - redirection #, opt. A if alerting |
| **DFxxF1idro#** | FACILITY, divertingLegInformation1, id - invoke id, reason - r, o - subscription option, # - diverting to number |
| **DFxxF3id** | FACILITY, divertingLegInformation3, id - invoke id |
| **DFxxIid** | FACILITY, ctIdentify invoke, id - invoke id |
| **DFxxIid,c-#** | FACILITY, ctIdentify results, id - invoke id, c -call identity, # redirecting party number |
| **DFxxPid,c-#** | FACILITY, pathReplacePropose, id - invoke id, c - call identity, # - redirecting number |
| **DFxxRid,c-#** | FACILITY, ctInitiate, id - invoke id, c - call identity, # - redirecting party number |
| **DFxxSidm#/#** | FACILITY (SSCT) on B-channel xx, id - invoke id, m - mode, # - transfer to number, /# - transferred # |
| **DFxxUid#** | FACILITY, ctUpdate, id - invoke id, # redirection number |
| **DPxx(p)** | CALL PROCeeding message on B channel xx, |

| | |
|---|---|
| | optional progress ind. p |
| **DPxxPccp** | PROGress message, cause cc, prog. ind. p |
| **DRxxcc** | RELease COMplete message on B channel xx, cause cc |
| **DRxxR(cc)** | RELease message on B channel xx, opt. cause cc |
| **DSxxbp(#)/#(Rr#)** | |
| **DSxxbp@(#)/#(Rr#)** | SETUP message, bearer capability b, progress ind. p, @ = sending comp., optional calling #, called #, optional redirect # |
| **DTA(text)** | Add display text to buffer |
| **DTC** | Clear display text buffer |
| **DXxx** | Status Query for B-channel xx |
| **DZxx** | RESTART interface starting at B channel xx |
| **DZxx(C)** | RESTART indicated B-channel xx |
| **D@F2idcr#** | Put divertingLegInformation2 in buffer, id - invoke id, c -diversion count, r - reason, # diversion number |
| **D@Mido#(/#)(=m)** | Put MWI operation in buffer, id - invoke id, 0 - operation, user #, opt. msg. center #, opt. m number of messages |
| **D@Nido<name>** | Put name in buffer, id - invoke id, o - operation |

**Interrupt Control Commands**

| | |
|---|---|
| **IN** | Enable transmit interrupts and messages |
| **IF** | Disable transmit interrupts and messages |

| | |
|---|---|
| **Query Commands** | |
| **QA** | Query alarms |
| **QEdd** | Query error counts for span dd |
| **QIrrrr** | Query DS26514 register rr |
| **QLss** | Query receive signal level |
| **QRxx** | Query call reference and state of B-channel xx |

| | |
|---|---|
| **Reset Commands** | |
| **RA** | reset all (resets spans, DSP functions, H.100 bus) |
| **RBxx** | reset B-channel xx (Layer 3) |
| **RPdd** | reset span dd |

| | |
|---|---|
| **Setup Commands** | |
| **SCm** | Set clock source mode, m = span, m = X if no span |
| **SFddfzsi** | Set framing for span dd,<br>T1: f = framing, D - D4, E - ESF<br>z = zero suppression, A - AMI, B - B8ZS<br>s = signaling, P - PRI, R - robbed bit, n - none<br>i = build out value<br>E1: f = framing, C - CRC4, N - non-CRC4<br>z = zero suppression, A - AMI, H - HDB3<br>s = signaling, C - channel associated, P - PRA, N - none<br>S - SS7 (timeslot 16 available)<br>i = impedance, B - 75 ohms, R - 120 ohms, |
| **SL(ll...ll)** | Set protocol layer for each span where l values are<br>2 - Layer 2 support<br>3 - Layer 3 support<br>A - Avaya QSIG<br>D - Nortel (Pre-NI2 DMS-100)<br>N - National ISDN 2/3<br>S - Siemens Corenet<br>T - Telcordia Generic Requirements |

| | |
|---|---|
| **SPxxtd(p)** | Set line protocol for channel xx; t = line type,<br>E - E&M<br>G - ground start<br>L - loop start<br>Q - Q.421<br>X-direct CAS control<br>N - none;<br>d = direction, O - FXO (CO), S - FSX (CPE);<br>p = optional address protocol, I - immediate, W - wink start |
| **SSabcdefgh** | Set system options a-h to Y - yes or N - no |
| **ST(xx...xx)** | Set span types for each span where x values are:<br>N - Network Termination (NT)<br>T - Terminal Equipment (TE)<br>M - Monitor an NT interface<br>O - Monitor a TE interface<br>U - Undefined/unused<br>* - No change to port type |
| **SZab** | Set receive impedance parameters a & b |

**TEI Management Commands**

| | |
|---|---|
| **TDddtt** | Disconnect data link tt on span dd |
| **TEddtt** | Establish data link tt on span dd |

**Version Requests**

| | |
|---|---|
| **VA** | Checksum of alternate segment request |
| **VC** | Version request |

**Download Commands**

| | |
|---|---|
| **@xxxx** | Download 1K block to address xxxx |
| **@Es** | Erase segment s |
| **GA** | Jump to Alternate Program |
| **GM** | Jump to Main Program |
| **@Ws** | Write from RAM to segment s |

**Diagnostics**

| | |
|---|---|
| **XBddEB** | Insert a single bit error on span dd |
| **XBddEIr** | Set BERT error rate to r on span dd |
| **XBddPpp..ppllcc** | Set BERT pattern pp..pp, pattern length ll, alternate word count cc, on span dd |
| **XBddRC** | Request BERT receive error count on span dd |
| **XBddRD** | Disable BERT receive function on span dd |
| **XBddREei** | Select BERT receive mode e, invert bits i |
| **XBddRTtttttt** | Enable BERT receive function on timeslots tttttt |
| **XBddS** | Resynchronize BERT receiver on span dd |
| **XBddTD** | Disable BERT transmit function on span dd |
| **XBddTEei** | Select BERT transmit mode e, invert bits i |
| **XBddTTtttttt** | Enable BERT trransmit funtion on timeslots tttttt |
| **XCddm** | Generate loopback code m on span dd (NT only) |
| **XLddm** | Set loopback m on span dd |
| **XOrrrrdd** | Set DS26514 register rrrr to dd, most significant bit selects chip |

## Messages from the Board

**Acknowledgments**

| | |
|---|---|
| **IA** | Acknowledge interrupts enabled |
| **RA** | Reset all acknowledged |
| **RPdd** | Reset span dd acknowledged |

**Alarm Events**

| | |
|---|---|
| **ABdd** | Blue alarm (AIS) event on span dd |
| **AGdd** | Green alarm event on span dd (no alarms active) |
| **AMddy** | Receive Distant MF Alarm, y - C=clear, S=set |
| **ARdd** | Red alarm (RCL) event on span dd |
| **ASdd** | Receive signaling all 1's |
| **AVddy** | V5.2Link Detected condition, y - C=clear, S=set |
| **AYdd** | Yellow alarm (RAI) on span dd |

**Layer 3 "D" Responses for NT Spans**

| | |
|---|---|
| **DAxxrrrrp** | ALERTing on B channel xx, progress p, call ref. rrrr |
| **DCxxrrrrp(#cnct)Ntname** | CONNect on B channel xx, prog. indicator p, call ref. rrrr, optional connected #, opt. connected name |
| **DCxxrrrrA** | CONNect ACKnowledge on B channel xx, call ref. rrrr |
| **DDxxrrrrccp** | DISConnect B channel xx, cause cc, opt. progress p, call ref. rrrr |
| **DExxc** | Layer 3 timer timeout c on channel xx |
| **DFxxcid(tt)** | FACILITY message on xx, component c, invoke id, opt. tag value tt |
| **DFxxAid#** | FACILITY, CTActive, id - invoke id, # - redirecting number |
| **DFxxCide#(A)** | FACILITY, CTComplete, id - invoke id, e - end designator, # - redirection #, opt. A if alerting |

| | |
|---|---|
| **DFxxF1idro#** | FACILITY, divertingLegInformation1, id - invoke id, reason - r, o - subscription option, # - diverting to number |
| **DFxxF2cr#(/name)** | FACILITY, divertingLegInformation2, c - diverting count, r - diversion reason, # - diverting number, /name - optional diverting name |
| **DFxxF3id** | FACILITY, divertingLegInformation3, id - invoke id |
| **DFxxIid** | FACILITY, ctIdentify invoke, id - invoke id |
| **DFxxIid,c-#** | FACILITY, ctIdentify results, id - invoke id, c -call identity, # redirecting party number |
| **DFxxNo<name>** | Facility Name element on xx, operation o,name |
| **DFxxPid,c-#** | FACILITY, pathReplacePropose, id - invoke id, c - call identity, # - redirecting number |
| **DFxxRid,c-#** | FACILITY, ctInitiate, id - invoke id, c - call identity, # - redirecting party number |
| **DFxxSidm#/#** | FACILITY (SSCT) on B-channel xx, id - invoke id, m - mode, # - transfer to number, /# - transferred # |
| **DFxxUid#** | FACILITY, ctUpdate, id - invoke id, # redirection number |
| **DPxxrrrrp** | CALL PROCeeding message, progress indicator p, call reference rrrr |
| **DPxxrrrrPccp** | PROGress message, cause cc, progress indicator p, call reference rrrr |
| **DRxxrrrrcc** | RELease COMplete on B channel xx, cause cc, call reference rrrr |
| **DRxxrrrrRcc** | RELease message, cause cc, call reference rrrr |
| **DSxxrrrrbp(#)** | SETUP message, bearer capability b, progress p, optional calling #, overlap sending |
| **DSxxrrrrbp(#)/(#)(Rr#)(Ntname)**<br>**DSxxrrrrbp(#)C(#)(Rr#)(Ntname)** | SETUP message, call reference rrrr, bearer capability b, progress ind. p optional calling #, C = sending complete, called #. optional redirect #, optional name |

**DXxxrrrrccss** STATUS message on B channel xx, cause cc, call state ss, call reference rrrr

**DZxx(C)** RESTART interface, if opt. C, restart specified B-channel

**DZxxA(C)** RESTART ACKnowledge, if opt. C, restart specified B-channel

**Layer 3 "D" Responses for TE Spans**

**DAxxrrrrp** ALERTing on B channel xx, progress indicator p, signal s, call reference rrrr,

**DCxxrrrrp(#cnct)(Ntname)** CONNect on B channel xx, progress p, call ref. rrrr, opt. connected #, optional connected name

**DCxxrrrrA** CONNect ACKnowledge on B channel xx, call ref. rrrr

**DDxxrrrrccp** DISConnect on B channel xx, cause cc, progress p, call ref. rrrr

**DExxc** Layer 3 timer timeout c on channel xx

**DFxxcid(tt)** FACILITY message on xx, component c, invoke id, opt. tag value tt

**DFxxAid#** FACILITY, ctActive, id - invoke id, # - redirecting number

**DFxxCide#(A)** FACILITY, ctComplete, id - invoke id, e - end designator, # - redirection #, opt. A if alerting

**DFxxF1idro#** FACILITY, divertingLegInformation1, id - invoke id, reason - r, o - subscription option, # - diverting to number

**DFxxF2cr#(/name)** FACILITY, divertingLegInformation2, c - diverting count, r - diversion reason, # - diverting number, /name - optional diverting name

**DFxxF3id** FACILITY, divertingLegInformation3, id - invoke id

**DFxxIid** FACILITY, ctIdentify invoke, id - invoke id

**DFxxIid,c-#** FACILITY, ctIdentify results, id - invoke id, c -call identity, # redirecting party number

| | |
|---|---|
| **DFxxMidAR** | FACILITY, MWI Return Result, id - invoke id |
| **DFxxNo<name>** | Facility Name element on xx, operation o,name |
| **DFxxPid,c-#** | FACILITY on xx, pathReplacePropose, id - invoke id, c - call identity, # - redirecting # |
| **DFxxRid,c-#** | FACILITY, ctInitiate, id - invoke id, c - call identity, # - redirecting party number |
| **DFxxSidm#/#** | FACILITY (SSCT) on B-channel xx, id - invoke id, m - mode, # - transfer to number, /# - transferred # |
| **DFxxUid#** | FACILITY, ctUpdate, id - invoke id, # redirection number |
| **DNxxnn** | NOTIFY message on B-channel xx, notification indicator nn |
| **DNxxR** | NOTIFY message, call resumed |
| **DNxxS** | NOTIFY message, call suspended |
| **DPxxp** | CALL PROCeeding message on B channel xx, progress indicator p |
| **DPxxrrrrPccp** | PROGress message on B channel xx, cause cc, progress indicator p, call reference rrrr |
| **DRxxrrrrcc** | RELease COMplete message, cause cc,call reference rrrr |
| **DRxxrrrrRcc** | RELease message, cause cc, call reference rrrr |
| **DSxxrrrrbp(#)/(#)(Rr#)(Ntname)** <br> **DSxxrrrrbp(#)C(#)(Rr#)(Ntname)** | SETUP message, call reference rrrr, bearer capability b, progress ind. p optional calling #, C = sending complete, called #. optional redirect #, optional name |
| **DSxxrrrrAp** | SETUP ACKnowledge, prog. ind. p, call ref. rrrr |
| **DTxx<text>** | Display text for B-channel xx, text |
| **DXxxccssrrrr** | STATUS message, cause cc, call state ss, call ref. rrrr |
| **DZxx(C)** | RESTART interface, if opt. C, restart specified B-channel |
| **DZxxA(C)** | RESTART ACKnowledge, if opt. C, restart specified B-channel |

**Error Messages**

| | |
|---|---|
| **EDddy** | Diagnostic error y on span dd |
| **ELddy** | Layer 1 or 2 error y on span dd |
| **EMddy** | MDL error y on span dd (Layer 2 protocol errors) |
| **ESddss** | Line Interface Unit event bits ss on span dd |
| **EXddss** | Elastic store event bits ss on span dd |

**Query Responses**

| | |
|---|---|
| **QAaaaaaaaa** | Query Alarm resonse, a - alarm state<br>G - no alarm<br>Y - Yellow Alarm (RAI)<br>B - Blue Alarm (AIS)<br>R - Red Alarm (RLOS or RCL) |
| **QEdd(data)** | Query Error Counts response, span dd |
| **QIrrrrdd** | Query DS26514 response, rrrr-register, dd-data |
| **QLssl** | Query receive level response, ss - span, l - level |
| **QRxxrrrrss** | Query B-channel reference response<br>rrrr = call reference, ss = call state |

**B-Channel State Change Messages**

| | |
|---|---|
| **SBxx** | Channel xx busy (response to CBxx message) |
| **SCxx** | Connect on B-channel xx acknowledged |
| **SFxx** | Off hook detected on channel xx (CAS) |
| **SHxx** | Hold on B-channel xx acknowledged |
| **SIxx** | Disconnect on B-channel xx acknowledged |
| **SLxx** | Listen on channel xx acknowledged |
| **SNxx** | On-hook detected on channel xx (CAS) |
| **SRxx** | Ringing detected on channel xx (CAS) |
| **SXxx** | Transmit on channel xx acknowledged |
| **SWxx** | Wink detected on channel xx |
| **SWxxE** | Wink timeout on channel xx |
| **SZxxb** | CAS bits changed to b for B-channel xx |

**Diagnostic Responses**

| | |
|---|---|
| **VAxxxx** | Checksum of the alternate segment |

**VCxxxxyyyyHTA8** Version response, 8 interface board
**VCxxxxyyyyHTA4** Version response, 4 interface board
xxxx = checksum of main segment, yyyy = version number, HT = board type, A = board revision, 4/8 = # of interfaces

| | |
|---|---|
| **VDxxxx** | DSP version xxxx |
| **U(msg)** | An undefined or unparseable message response |

**Maintenance Responses**

| | |
|---|---|
| **XbddRCeeeeeecccccccc** | BERT error count interface dd, eeeeee - number of errors, cccccccc - number of bits |
| **XLddc** | Loopback code c detected on interface dd |
| **XLddcc** | Bit-pattern message cc detected on ESF interface dd |