MUXPORT Protocol

Reference Guide

codex

MOTOROLA

MUXPORT SEMINAR

NATIONAL TECHNICAL SUPPORT

REFERENCE GUIDE

JULY 1986

Prepared by: Bart M. Zaino

INTERNATIONAL STANDARDS ORGANIZATION
OPEN SYSTEMS INTERCONNECT MODEL

1.1 OSI OVERVIEW

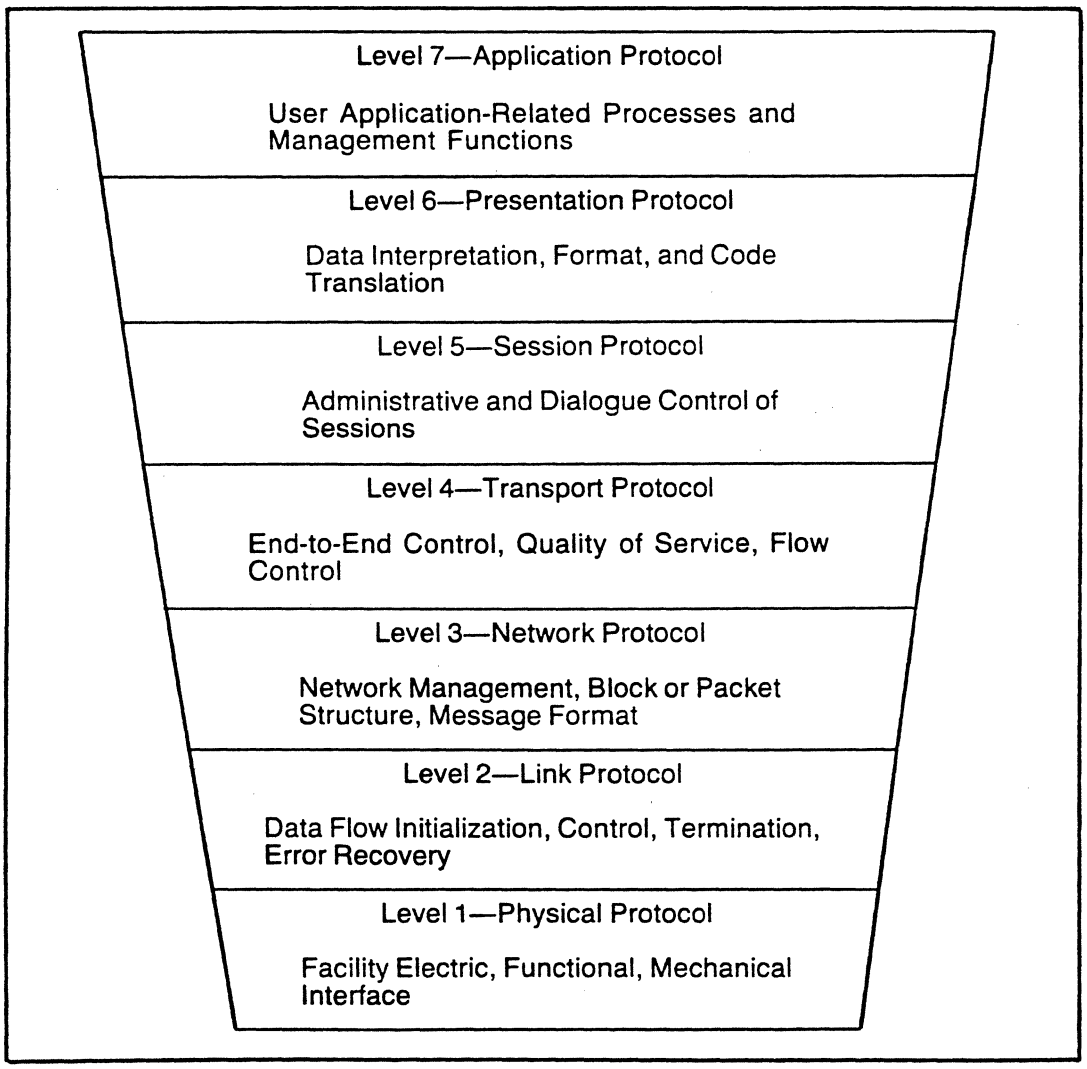
The term OPEN SYSTEMS means the ability of many vendors equipment to communicate with each other through a single network. This is difficult today because each vendor offers a unique network structure. Users may become locked into a single vendors network. To avoid this problem the International Standards Organization (ISO) developed the Open Systems Interconnect Model (OSI Model) to promote connectivity between vendors.

The OSI model is not a specification or a standard but rather a framework that permits interconnect procedures to be defined. Once the procedures are defined as protocols users will be able to exchange data regardless of the hardware used. The goals of the OSI are:

- Serve as a structure for communications between users
- Act as a framework for future services and protocols
- Promote compatibility between communications features
- Provide flexibility for future technology

The OSI Model (see figure below) is based upon a layered architecture. There are seven layers in the model, each layer has its own unique attributes. The purpose of these seven layers is to define the various functions that must be carried out when two machines communicate. Each layer requests services from the layer below it and provides services to the layer above.

Layer 1 involves circuit interface and standard connector cables. Layer 2 specifies facilities for frame transfer between network nodes/terminals. Layer 3 is the packet layer, it creates packets which go to the lower two layers for transmission. Levels 4-6 are end-to-end functions which account for the virtual passing of messages between application programs. Level 7 is the user specific application interface. A detailed explanation of each layer is given below.



1.1 DEFINITION OF THE OSI LAYERS

LAYER 1 - PHYSICAL LAYER - This layer is concerned with transmitting data bits (0's and 1's) and for acquiring, maintaining and disconnecting the physical circuits between systems. Some Physical Layer protocols are: RS-232, V.24, X.21. This is the only layer that actually moves data between machines.

LAYER 2 - DATA LINK LAYER - This layer is responsible for the reliable exchange of data across a link established by the physical layer. This layer breaks up the input data into frames, transmitting these frames and processing the acknowledgements from the received data. This layer manages the control, termination and establishment of the logical link. Because layer 1 only accepts a serial stream of bits without regard to meaning and structure, it is up to the data link layer to create and recognize frame boundaries. Data link protocols include BSC, HDLC, MUXPORT, SDLC.

LAYER 3 - NETWORK LAYER - This layer provides services for moving data through a network between end point nodes. This layer provides for the functions of internal network operations. Some of these operations are: addressing, routing, switching sequencing, flow control and error recovery of packetized data. An example of Level 3 protocols are X.25 and the DOD Internet Control Protocol (ICP). This layer is also called the packet or subnetwork layer.

LAYER 4 - TRANSPORT LAYER - This layer forms the boundary between data communications (layers 1-3) and the data processing functions (layers 5-7). This level provides end-to-end data integrity and optimization of user resources. The DOD's Transmission Control Protocol (TCP) is a level 4 protocol. IBM's SNA is also a level 4 or Host-to-Host protocol.

LAYER 5 - SESSION LAYER - A session is defined as the dialog that two devices carry out, the actual data transfer. This layer provides two classes of service to the higher levels, Administration and Dialog. Administration services act to establish (BIND) and release (UNBIND) a connection between two Presentation (layer 6) entities. The Dialog service controls the data transfer. A Session can span several Transport (layer 4) connections.

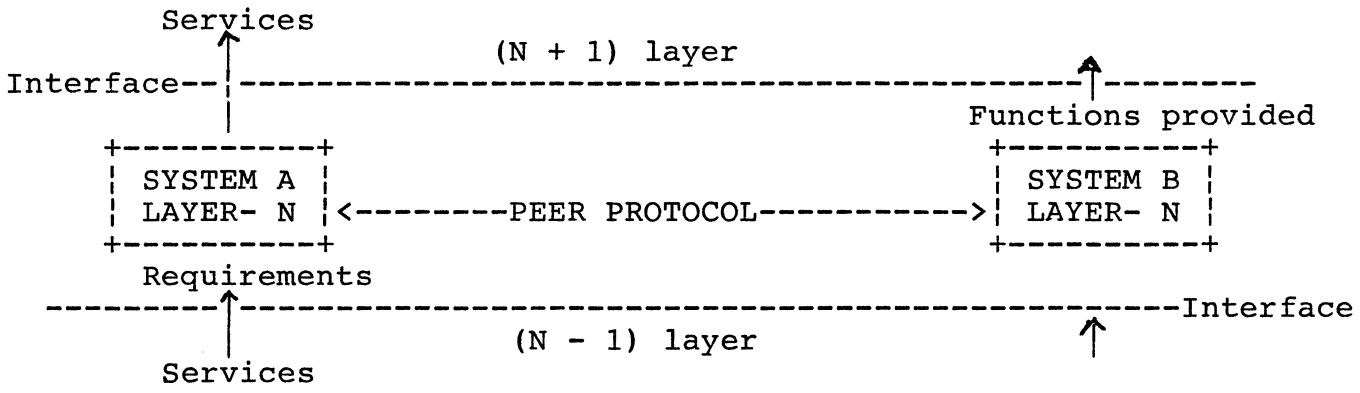
LAYER 6 - PRESENTATION LAYER - An application (layer 7) uses this layer to properly interpret the data being transferred. The Presentation layer will check syntax, format the data and do translation of it if necessary. An example of a function performed by this layer is video screen formatting.

LAYER 7 - APPLICATION - This layer is the window through which the user gains access to the services provided by the other layers. The Application layer provides the user data in a readable form.

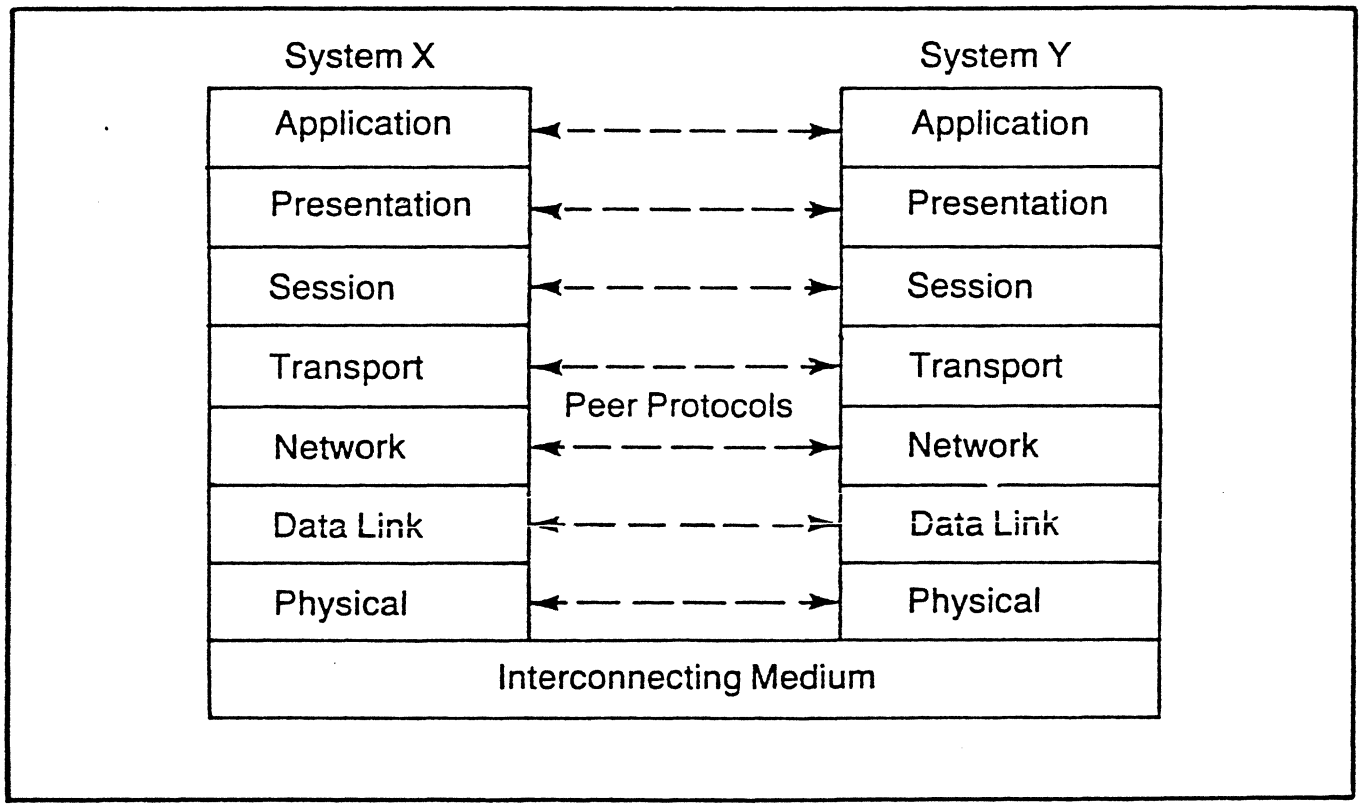
1.2 LAYER CONCEPTS

The use of layers forces a well defined interface between devices. Complex problem are more easily solved using a layered or building block approach. Layering also makes changes easier to implement because only a single layer is changed to implement an enhancement or fix a bug.

A layer will request services from the layer below (these are the requirements of the layer to complete its function). The layer will process the data received from the lower layer and pass to the next higher layer, providing a service to that layer, so it may complete its function. Each layer may add or subtract information from the original user data unit dependent on the operation being performed. During a transmit operation information is added and during a receive operation information is subtracted. The following figure illustrates the relationship between layers in the model:



Each layer in the model physically interfaces only to the level above and the level below. The layers also maintain a virtual peer-to-peer relationship with the corresponding layer in the other connected device. Communications appears to each layer to be directly between it and its associated layer when in actuality the communication path is down through the lower layers, across the physical link and up through the lower layers to its peer layer. The following figure illustrates this concept.

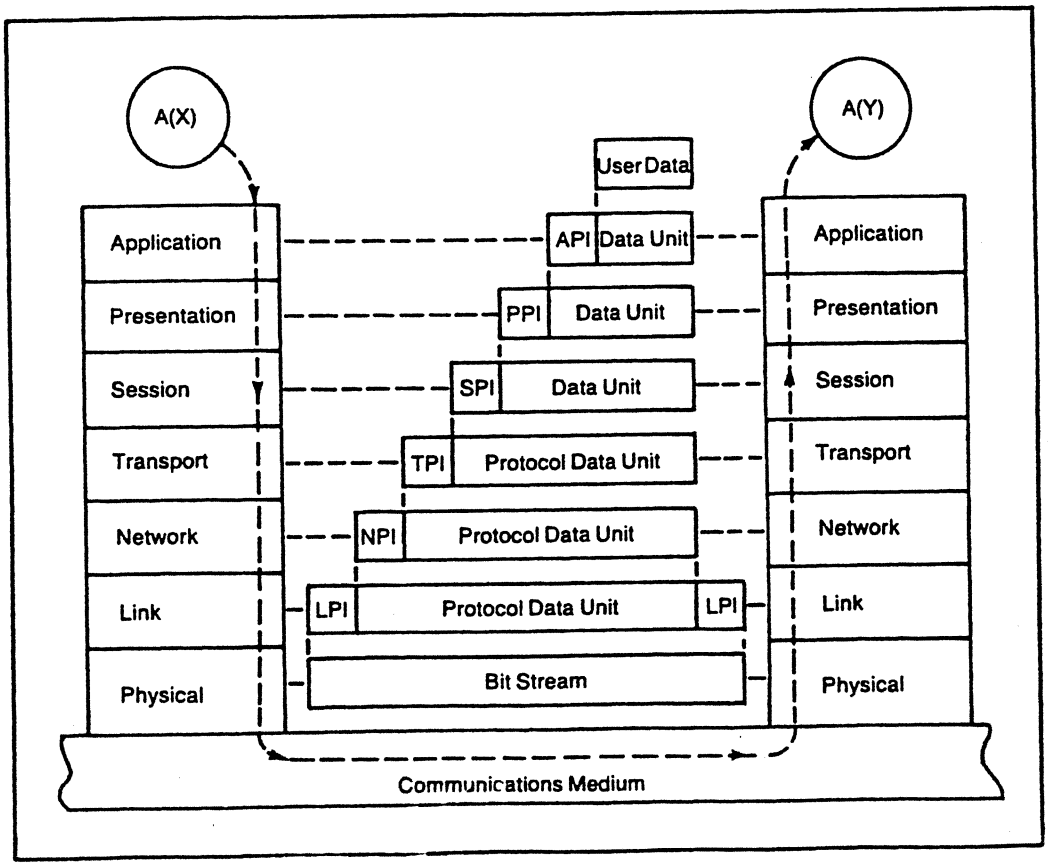


1.2.1 DATA TRANSFER

Communication activity in the OSI model occurs in phases. First a connect phase establishes the connection at the physical and link levels. The data transfer phase then begins to exchange user data. In the local system (X) {see figure below} the user data flows down through the layers. Each layer in turn adds any required protocol information and passes the complete unit to the next lower layer.

The user through layer 7 (Application) wishes to send a screen of data to the remote unit. The presentation layer (layer 6) may add a screen format information header to the user data and send it to layer 5 (Session) where the proper higher level protocol 'call' is established and verified. Layer 4 (Transport) will then ensure the proper host-to-host protocol header has been appended (such as TCP or SNA) and send it to layer 3. The Network layer will packetize the data into an X.25 frame for the level 2 protocol. The Data Link layer will build HDLC frames combining this users data with other users data and passes them to the physical layer (layer 1) for transmission to the remote unit.

The data is then sent across the communications medium and up through the layers of the remote unit (system Y). Each layer in the remote system will strip off its peer protocol information (if any) and send the data unit up to the next layer. This process continues upward until the original user application information is delivered to the remote user in the proper format.



1.3 PROTOCOL DIFFERENCES

This chart shows the differences and between character, character count (DDCMP) and bit oriented protocols.

FEATURE	BISYNC	CHARACTER COUNT	BIT ORIENTED
Format	Sync	Async/sync	Sync
Link modes	Half duplex	Half/full duplex	Half/full duplex
Frame formats	Numerous	1 (3 types)	1 (3 types)
Link control information	Optional header	Required header	1 or 2 octet control field
Station addressing	Point-Point multipoint	Address in header	Single/extended address field
Error checking	Text only	Header and info field separately	All between flags
Error detection/ generation	VRC/LRC-8 VRC/CRC-16 CRC-16	CRC-16	CRC-CCITT V.41 CRC-32
Request for retransmission	Stop and wait	Go back n frames	Go back n or selective reject
Max frames outstanding	1	255	7 or 127
Flow control	Control char. and No-ack	None	RNR frame, window
Character codes	ASCII,EBCDIC SBT	ASCII for SOH,DLE,ENQ only	Any
Data length	n X 8 or 6	n X 8	Unrestricted
Transparency	Via escape mechanism	Character count	Zero insertion/ deletion
Control character bit pattern	Numerous 1 and 2 character sequences	SYN, SOH DLE, ENQ	Abort, idle

BIT ORIENTED PROTOCOL

2.0 INTRODUCTION

Bit Oriented Protocols were developed to overcome the deficiencies encountered in Character Oriented Protocols, i.e. Bisync. These deficiencies are:

- o Lack of transparency (complicated escape mechanism)
- o Large code set required for link control functions
- o Half duplex operation only

Bit oriented protocols overcome these deficiencies and provide the following features:

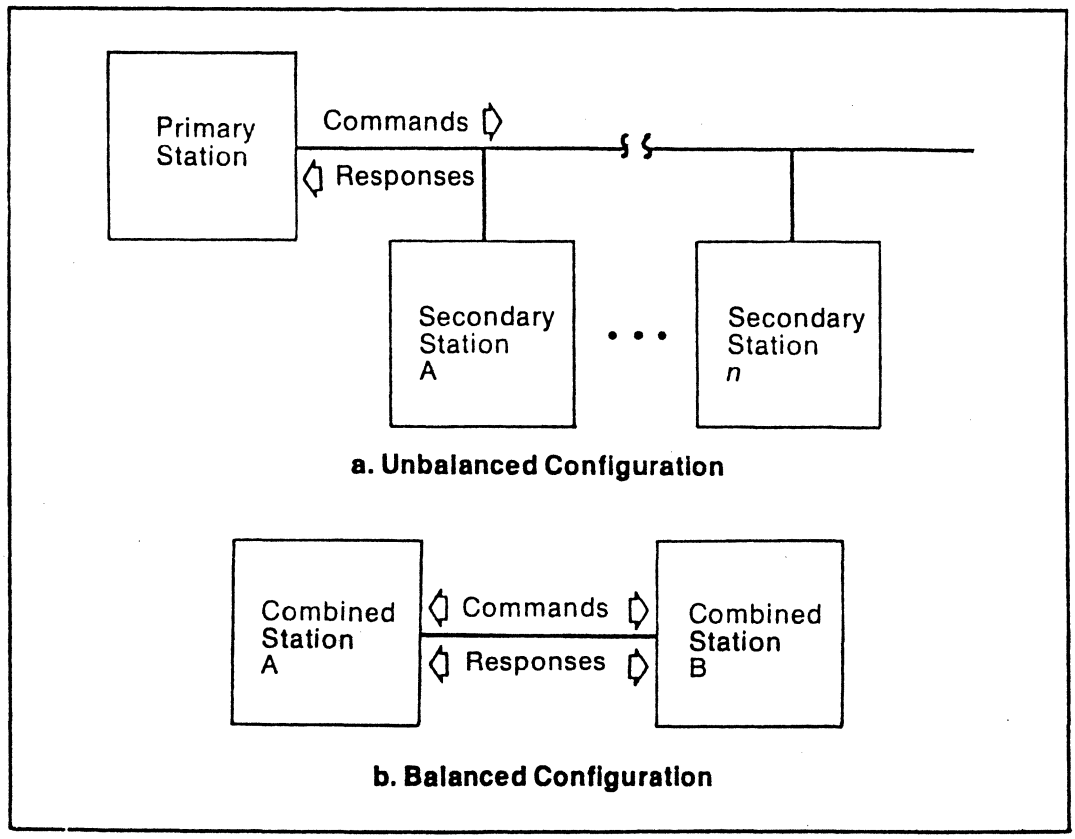
- o Bit orientation, allows use of the complete code set for information transfer.
- o Code independence, because the framing and control fields separate the link functions from the data pattern any character set can be used.
- o Greater efficiency, BOP can run full duplex

Bit Oriented Protocols refer to a serial by bit transfer of information across a communications link. BOP protocols operate in one of two basic configurations; point-to-point or multipoint. In BOP protocols any binary pattern can be transmitted between the opening and closing flags. Data transparency is achieved through the use of a zero insertion technique. The zero insertion technique prevents random bit patterns from appearing as a flag character (01111110 or 7E) and prematurely terminating the frame. This technique is explained in Section 2.7.

2.1 DATA LINK CONFIGURATIONS

The physical line along with the DCE equipment make up the Data Link. The data link determines the non-data transmissions necessary to set up, verify and terminate data transfer. These transmissions are called the Data Link Control.

In HDLC there are two basic data link configurations: unbalanced and balanced. An unbalanced configuration features one primary station and one or more secondary stations connected to a communications link. The configuration is unbalanced because the primary station controls all data flow between stations and initiates error recovery. The secondary station has no direct control of the link but responds to primary command frames. A balanced configuration consists of two combined stations (primary and secondary in one unit) connected point-to-point. Both stations have equal data transfer and link control ability. A combined station transmits commands and responses to, and receives commands and responses from another combined station. The following figure depicts the basic data link configurations:



2.2 OPERATIONAL MODES

The operation of the two data link configurations, balanced and unbalanced, can be two-way alternate or two-way simultaneous using dedicated or switched facilities in any one of three operational modes.

Normal Response Mode (NRM) is used in an unbalanced configuration where the secondary can initiate transmission only after receiving explicit permission from the primary. Permission is obtained when a frame with the Poll bit set is received by the secondary. The secondary will then transmit its traffic, the last frame sent will have the Final bit set.

Asynchronous Response Mode (ARM). Used in an unbalanced configuration in which the secondary station initiates transmission without explicit permission from the primary station. The secondary may send more than one frame when transmitting. This method is used when a primary and a single active secondary wish to exchange data without the overhead of continuous polling sequences.

Asynchronous Balanced Mode (ABM) provides for a symmetrical data transfer between two combined stations. Each station can initialize and disconnect the link. Each station is also responsible for controlling its own data flow and error recovery. This mode is used when the primary/secondary relationship is unacceptable.

2.3 TRANSMISSION STATES

There are three states in which the data link can operate, they are:

- o Transient state
- o Idle state
- o Active state

The transient state exists when the channel is being conditioned before initial transmission and after each transmit or receive reversal (i.e half duplex operation) This is the state during the RTS/CTS delay.

The idle state is entered when the link is operational but there are no frames (information or control) being exchanged. This state is entered by sending at least fifteen one bits across the line.

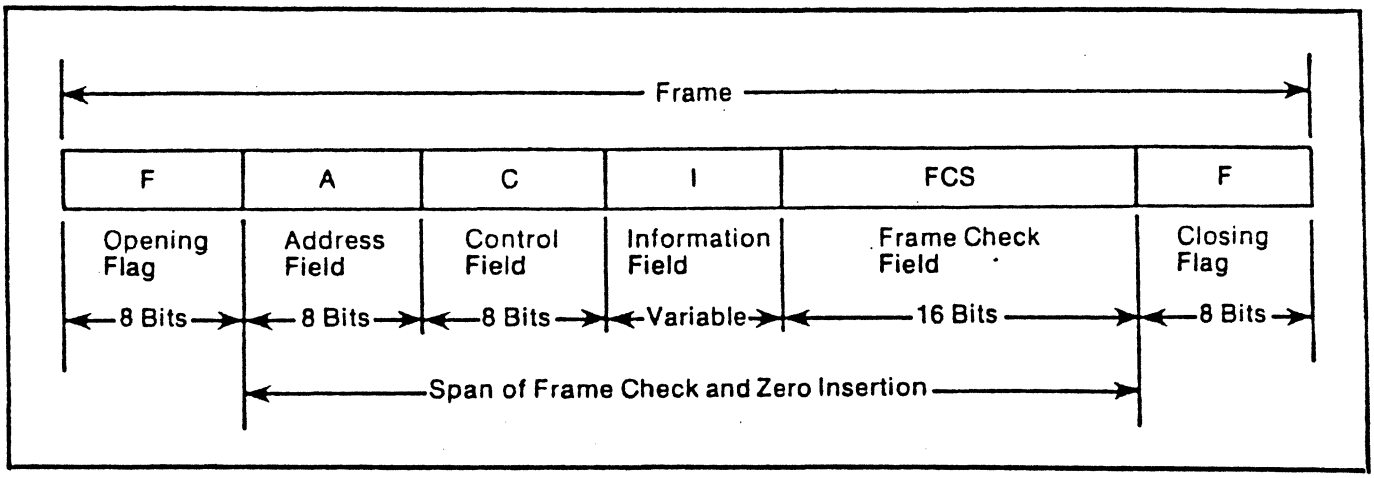
The active state occurs when the units are exchanging control or information frames. Flags are considered an idle condition by the protocol but the link is considered to be in the active state.

2.4 FRAME DEFINITION

A frame is a sequence of contiguous bits bounded by and including the opening and closing flag sequences. The frame structure governs the formatting and significance of the various fields within the frame boundaries. All frames have the same general structure and elements, these elements are:

- o Opening and closing flags
- o Address field
- o Control field
- o Information field (optional)
- o Frame check sequence

These elements are combined into the transmission frame as illustrated below and explained in detail on the next page.



FLAG FIELD (F) - the flag is an eight bit field generated at the transmitter. The provide synchronization at the receiver and acts as a reference point for the other fields in the frame. All frames open and close their transmissions with a flag sequence which has the following binary configuration 011111110 (7E). The flag sequence is prevented from occurring in the other frame elements by the use of a zero insertion technique.

An opening flag can be followed by a frame or by another flag. An ending flag can be followed by another flag, an idle line (fifteen contiguous 1 bits) or can be used as the opening flag of the next frame.

ADDRESS FIELD (A) - The address field follows the opening flag. This field is used to differentiate between secondary stations on the link. The contents of this field can be either a single, group or global address. In unbalanced modes of operation the primary station is never identified. The primary will put the address of the secondary he wishes to poll in the address field. The secondary station will also put its address in this field so the primary knows where the data came from. In balanced operations (i.e ABM) commands will have the address of the Receiving unit while Responses contain the Transmitting unit address.

CONTROL FIELD (C) - This field provides the signals to operate the data link: its bits identify frame type, sequence numbers and command or response codes. The control field may be eight bits (normal) or sixteen bits (extended) depending on the operating mode of the stations. There are three types of frames defined by this field: Information, Supervisory and Unnumbered. If the leading bit in the control field is a "0" the frame is an information frame. The control field functions are explained in detail below.

If the leading bit of the field is a "1" the protocol uses the second bit to distinguish between Supervisory and Unnumbered or Management frames. The supervisory frames start with a "1 0" sequence and the Unnumbered frames start with a "1 1".

Each format contains a Poll/Final bit (P/F). The P/F bit is used like a send/receive indicator: it can be either a poll (P) bit sent by a primary to a secondary to request transmission or it can be a final (F) bit sent by the secondary to mark the end of a transmission. Upon reception of a poll bit the secondary must try to get his response (F bit set) as quickly as possible. The basic format of the control field for extended and non-extended formats is shown below:

NON EXTENDED

FRAME TYPE	8	7	6	5	4	3	2	1
I-FRAME	Nr			P/F	Ns			0
S-FRAMES	Nr			P/F	S	S	0	1
U-FRAMES	M	M	M	P/F	M	M	1	1

LSB

Where:

P/F = Poll/Final Bit
S = Supervisory bit
M = Management bit
Ns = Send count
Nr = Receive count
X = Don't care

EXTENDED

TYPE	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
I-FRAME	Nr							P/F	Ns							0
S-FRAME	Nr							P/F	X	X	S	S	S	S	0	1
U-FRAME	0	0	0	0	0	0	0	P/F	M	M	M	X	M	M	1	1

LSB

Specific commands are shown in the Frame Format section, Section 2.5.

INFORMATION FIELD (I) - This is an optional field and contains the actual user data. The I-field is typically unrestricted in length and may contain any code type.

FRAME CHECK SEQUENCE (FCS) - Each frame contains a 16-bit frame check sequence immediately following the I-field (or C-field if there is no I-field). This field detects errors induced by the transmission link. The FCS is the result of a mathematical calculation (cyclic redundancy check) done on all bits (except inserted zeros) in the address, control and information fields of the frame. The FCS is followed by the closing flag of the frame.

2.5 FRAME FORMATS

2.5.1 UNNUMBERED FORMAT (U-FRAME)

The U-frame provides for the initialization of secondary stations, controlling responses from those stations and reporting certain procedural errors. The following chart shows the U-Frame functions:

FLAG	ADDRESS	CONTROL	FCS	FLAG

Command/Response	8	7	6	5	4	3	2	1	Name
Set Normal Resp. Mode	1	0	0	P	0	0	1	1	SARM
Set Normal Resp. Mode Ext.	1	1	0	P	1	1	1	1	SNRME
Set Async Resp. Mode	0	0	0	P	1	1	1	1	SARM
Set Async Resp. Mode Ext.	0	1	0	P	1	1	1	1	SARME
Set Async. Balanced Mode	0	0	1	P	1	1	1	1	SABM
Set Async Bal. Mode Ext.	0	1	1	P	1	1	1	1	SABME
Disconnect	0	1	0	P	0	0	1	1	DISC
Set Initialize Mode	0	0	0	P	0	1	1	1	SIM
Exchange Station ID's	1	0	1	P/F	1	1	1	1	XID
Unnumbered Acknowledge	0	1	1	F	0	0	1	1	UA
Disconnect Mode	0	0	0	F	1	1	1	1	DM
Request Disconnect	0	1	0	F	0	0	1	1	RD
Request Initialize Mode	0	0	0	F	0	1	1	1	RIM
Command (Frame) Reject	1	0	0	F	0	1	1	1	CMDR
Unnumbered Poll	0	0	1	P	0	0	1	1	UP
Unnumbered Information	0	0	0	P/F	0	0	1	1	UI

2.5.2 U-FRAME DESCRIPTIONS

SNRM - The set normal mode command is used to place the addressed secondary in the normal response mode and to subordinate the receiving station to the transmitting station.

SNRME - Sets normal mode as above but uses an extended control field. The control field is now 16 bits long instead of 8.

SARM - The set asynchronous response mode command places the secondary in the ARM mode.

SARME - Puts the secondary in ARM mode with an extended control field.

SABM - The set asynchronous balanced mode command places the secondary in the ABM of operation.

SABME - Places the secondary in ABM mode with extended control fields.

DISC - The disconnect command informs the receiving station that the sending station is suspending operation. A physical disconnect will also be initiated.

SIM - The set initialize mode command initiates link-level initialization procedures at the secondary.

XID - The exchange identification command causes the addressed station to report its address.

UA - The unnumbered acknowledge response is used to acknowledge the receipt of a unnumbered command (i.e. SABME).

DM - The disconnected mode response is sent to the primary to request a set mode command to initialize the secondary.

RD - The request disconnect response indicates to the primary that the secondary wishes to be placed in the disconnected mode.

RIM - The request for initialization mode is transmitted by a station to notify the primary of the need for a set mode command (SIM).

CMDR - The command reject command is used to report an error condition not recoverable by retransmission. i.e invalid command, invalid Nr, I-field to long.

UP - The unnumbered poll is used to get a single response from the specific secondary (or group of secondary) stations.

UI - The unnumbered information command and response are used to transfer non sequentially numbered information fields across the link. Used for higher level status, link initialization data. These frame are not verified and may be lost.

2.5.3 SUPERVISORY FORMAT (S-FRAME)

The S-frame is used to assist in the transfer of information. They are used to confirm receipt of preceeding frames. The S-frame does not carry data but will acknowledge the reception of data, convey ready or busy conditions and report frame numbering errors. The S-frame formats are shown below.

FLAG	ADDRESS	CONTROL	FCS	FLAG

Command/Response	8	7	6	5	4	3	2	1	Name
Receiver Ready	Nr			P/F	0	0	0	1	RR
Receiver Not Ready	Nr			P/F	0	1	0	1	RNR
Reject	Nr			P/F	1	0	0	1	REJ
Selective Reject	Nr			P/F	1	1	0	1	SREJ

2.5.4 S-FRAME DESCRIPTIONS

RR - The receiver ready frame is used by a station to indicate that it is ready to receive I-Frames and to acknowledge the reception of information frames up to and including Nr -1.

RNR - The receive not ready frame is used to indicate the temporary inability to accept additional information. I-Frames numbered up to and including Nr -1 are acknowledged. A station receiving a RNR frame must stop transmitting at the earliest possible time by completing the or aborting the current frame.

REJ - The reject frame is used to request the retransmission of I-Frames starting with the frame numbered Nr. All I-Frames numbered Nr -1 and below are acknowledged.

SREJ - The selective reject frame is used to request retransmission of a single I-Frame numbered Nr. All frames numbered Nr -1 are acknowledged. Once a SREJ has been transmitted the only I-Frames accepted are those sequentially following the requested I-Frame.

2.5.5 INFORMATION FORMAT (I-FRAME)

These frame are used to transfer data. The I-Frame contain the send (Ns) and receive (Nr) counts. The counts are used to ensure that frames are received in their proper order (Ns count) and to confirm the accepted information (Nr count).

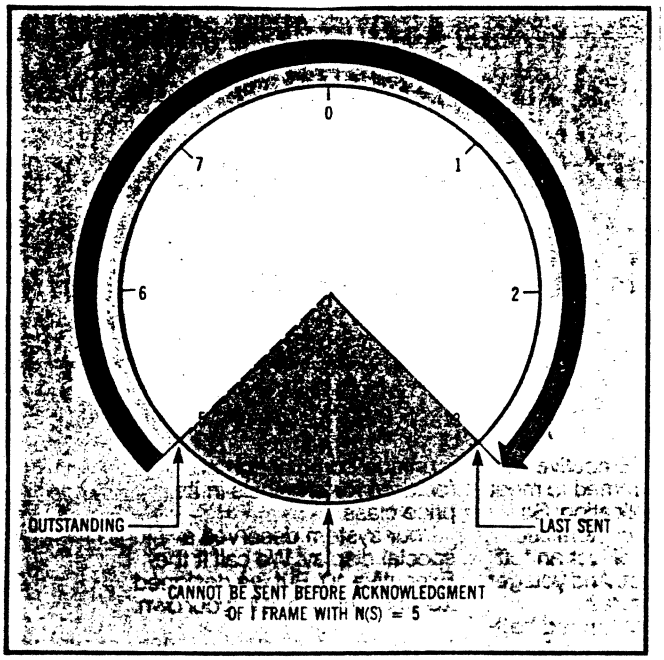
The Ns count indicates the number of the I-Frame within the sequence of data frames transmitted. The Nr count is the number of the frame that the receiver expects to receive next (the Ns of the remote unit). The following diagram shows the I-Frame format:

FLAG	ADDRESS	CONTROL	DATA	FCS	FLAG

Command/Response	8	7	6	5	4	3	2	1	Name
Information	Nr			P/F	Ns			0	I

2.6 FRAME NUMBERING

Frame numbering insures that frames are received in the same order that they were transmitted. Each I-Frame is sequentially numbered and may have the value 0 through modulus -1, where modulus is the moduluo of sequence numbers. The modulus equals 8 for the basic control field and 128 in the extended format. The sequence numbers cycle through the entire range and restart at 0. The maximum number of I-Frames a station may have outstanding, unacknowledged by the remote unit, is called the window size. This number can never exceed one less than the modulus (7 for nonextended or 127 for extended). The window size is determined by the storage capability of the station. This is the maximum number of frames that can be stored for retransmission in the event of an error. The following Figure illustrates window operation:



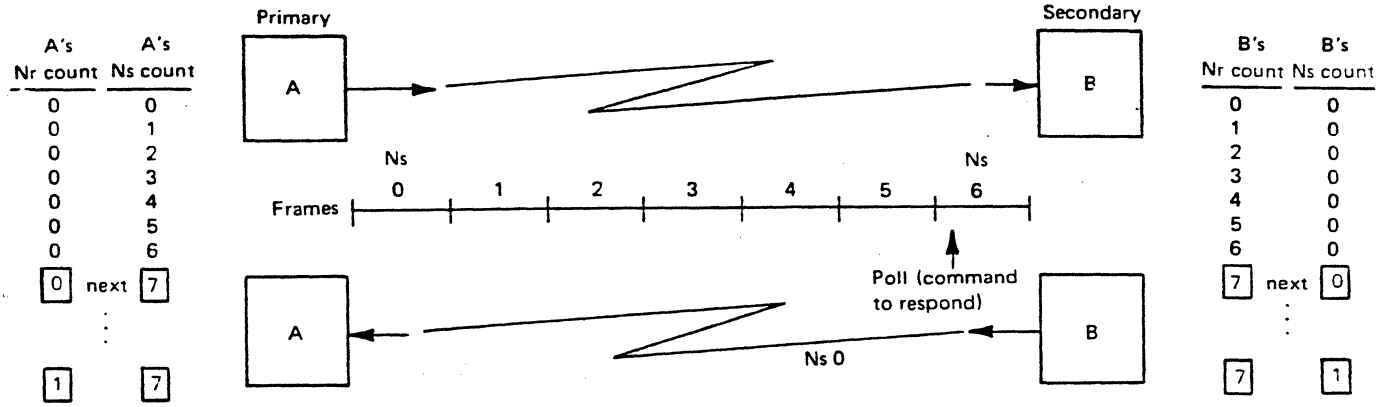
Window size = 7. I-Frames with Ns = 5,6,7,0,1,2,3 have been sent without receiving an ack. The window is now closed (no further I-Frames sent). One or more frames must be acknowledged by the receiver (Nr = 6) before the next I-Frame can be transmitted.

Every station maintains a send variable (Vs) on the I-Frames it transmits and a receive variable (Vr) on the the I-Frames it receives correctly. The send variable (Vs) indicates the number of the NEXT frame to be transmitted. Vs is incremented by one (using moduluo arithmetic) for each I-Frame transmitted. For example if Ns = 6 then Vs = 7. Prior to transmission of an I-Frame the Ns is set equal to Vs after transmission Vs is incremented.

Each station will also maintain a receive variable (Vr), equal to the expected Ns contained in the next received I-Frame. Vr is incremented by one upon reception of an error free I-Frame where Ns = Vr. For example, if the modulus equals 8 and an I-Frame is received with Ns = 7 and Vr currently equals 7 the Vr will be incremented to 0.

All I-Frames and S-Frames contain an Nr count. The Nr count is the expected sequence number of the received I-Frame. Just before transmitting or retransmitting a frame the Nr is set equal to Vr indicating that the station transmitting the Nr has received correctly all I-Frames up to and including Nr -1.

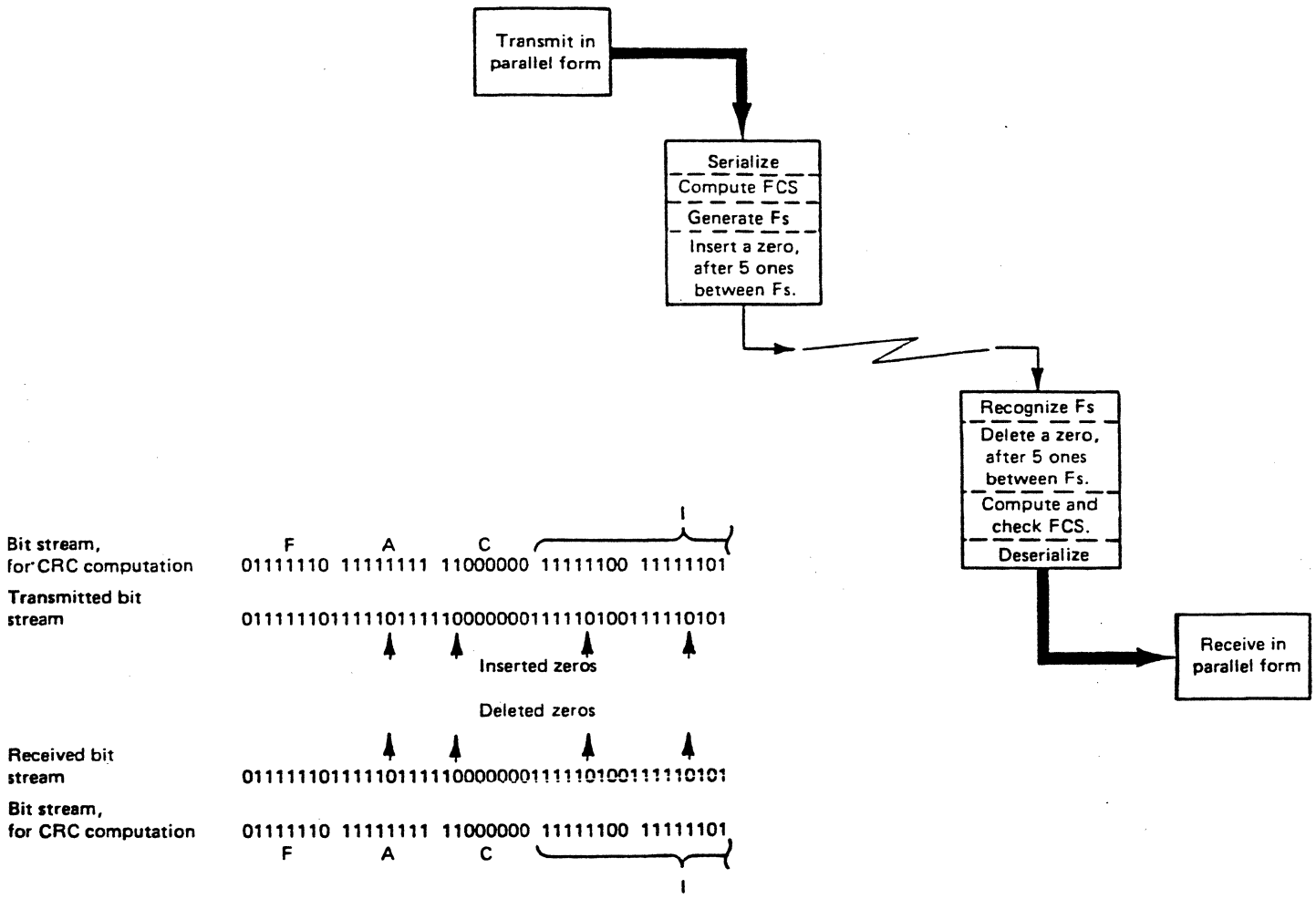
The following figure illustrates the Ns Nr concepts:



If B responds to the poll with Nr =:	A may send Ns frames:
7 (as above, all frames check OK)	7, 0, 1, 2, 3, 4, 5 (continue)
6 (frame 6 discarded because of error)	6, 7, 0, 1, 2, 3, 4 (retransmit and continue)
5 (error on frame 5, 5 and 6 discarded)	5, 6, 7, 0, 1, 2, 3 (retransmit and continue)
4 (error on frame 4, 4-6 discarded)	4, 5, 6, 7, 0, 1, 2 (retransmit and continue)
3 (error on frame 3, 3-6 discarded)	3, 4, 5, 6, 7, 0, 1 (retransmit and continue)
2 (error on frame 2, 2-6 discarded)	2, 3, 4, 5, 6, 7, 0 (retransmit and continue)
1 (error on frame 1, 1-6 discarded)	1, 2, 3, 4, 5, 6, 7 (retransmit and continue)
0 (error on frame 0, no frames accepted)	0, 1, 2, 3, 4, 5, 6 (retransmit)

2.7 ZERO INSERTION (BIT STUFFING)

A frame is identifiable because it begins and ends with a flag and contains only nonflag bit patterns between them. This characteristic does not restrict the contents of a frame because BOP protocols require that a binary 0 be inserted by the transmitter after any succession of five contiguous 1's within the frame. This insures that a flag pattern (01111110) is never transmitted between the beginning and ending flags. The zero insertion is disabled during flag transmission. After testing for flag recognition the receiver removes zero bits after every five 1's. The inserted /removed zero bits are not included in the FCS computations. Reference the following figure for an explanation of the zero insertion technique.



2.8 FRAME ADDRESSING CONVENTIONS

The normal (non-extended) address field is eight bits long. Eight bits gives a possibility of up to 256 different addresses. When using extended mode the address field has the ability to be any length required. If the first bit of the byte contains a '0' this indicates that the byte contains addressing information, if the first bit is a '1' it indicates that this is the last byte of the address field.

An address can be single or broadcast. One station can have several functional addresses, for example, a main address, broadcast or general poll address a functional or group address. The address or addresses the unit responds to is dependent on the protocol conventions being used.

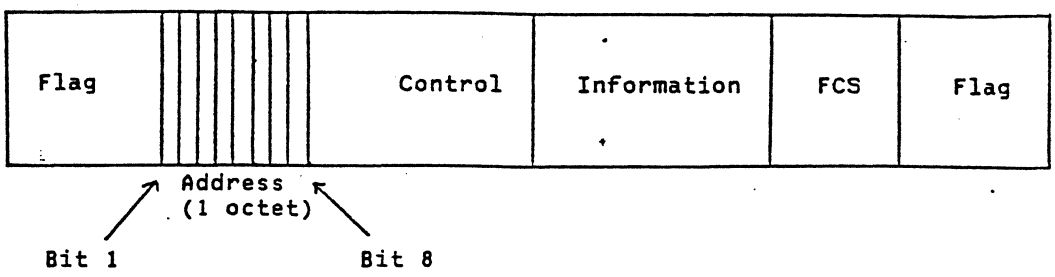
2.8.1 X.25 AND MUXPORT ADDRESSING

The X.25 and Muxport conform to the Link Access Protocol Balanced (LAPB) addressing conventions. In this scheme there are only two addresses 'A' and 'B'. The DTE in X.25 or secondary in Muxport Protocol is given address 'A' while the DCE in X.25 or primary in Muxport uses address 'B'. The primary/secondary relationship in Muxport protocol is for VCTP operations only not link control. The following bit sequences are used for the addressing:

DTE = A = 03 = 00000011 = Muxport secondary = X.25 subscriber

DCE = B = 01 = 00000001 = Muxport primary = X.25 network

Commands FROM the DCE and responses TO the DCE will use address A (03)
Commands FROM the DTE and responses TO the DTE will use address B (01)



Frame Type	Direction of Frame Transmission DTE (A) — DCE (B)	Address Field Format Bit 8 ... Bit 1
Command	→	0 0 0 0 0 0 0 1
Response	←	0 0 0 0 0 0 0 1
Command	←	0 0 0 0 0 0 1 1
Response	→	0 0 0 0 0 0 1 1

6000 MUXPORT PROTOCOL

3.0 INTRODUCTION

A Muxport is the hardware interface for communications between devices in the 6000/6700 product lines. The Muxport is a multithreaded port that passes data for 1-31 devices which may be operating with non-homogenous (different) protocols. The Muxport acts as a gateway for the customers data rather than an endpoint.

The Codex Multiplex Protocol (CMP) is the method used to convey information between Codex multiplexers using the Muxport hardware. Muxport protocol was designed to provide the following features:

- o Layered implementation using standard HDLC hardware
- o High efficiency using stat mux techniques
- o Expandability, supports 1-31 devices
- o Flow Control and error protection
- o X.25 level 2 and HDLC line layer compatibility

Communications over the Muxport is always 2-way-simultaneous (full Duplex). The connection is always leased line and the link channel state is Active (see section 2.3) during normal system operation.

Muxport protocol is divided into four layers: LINE, ARQ, MUX and CONNECTION. Each layer is designed so that processing of data within the layer requires no knowledge of the contents of the data within the other layers. The following table compares the CMP to other layered protocols.

Layer	ISO / X.25	SNA	DECNET	MUXPORT
7	Application	End User	Application	NOT USED BY MUXPORT PROTOCOL
6	Presentation	NAU Services		
5	Session	Data flow Transmission	None	
4	Transport	Ntwrk Services		
		Path Control		
3	Network/Packet		Transport	Connection
				MUX
2	Data Link/HDLC	SDLC	DDCMP	ARQ
				Line
1	Physical	Physical	Physical	Physical

3.1 MUXPORT FRAME

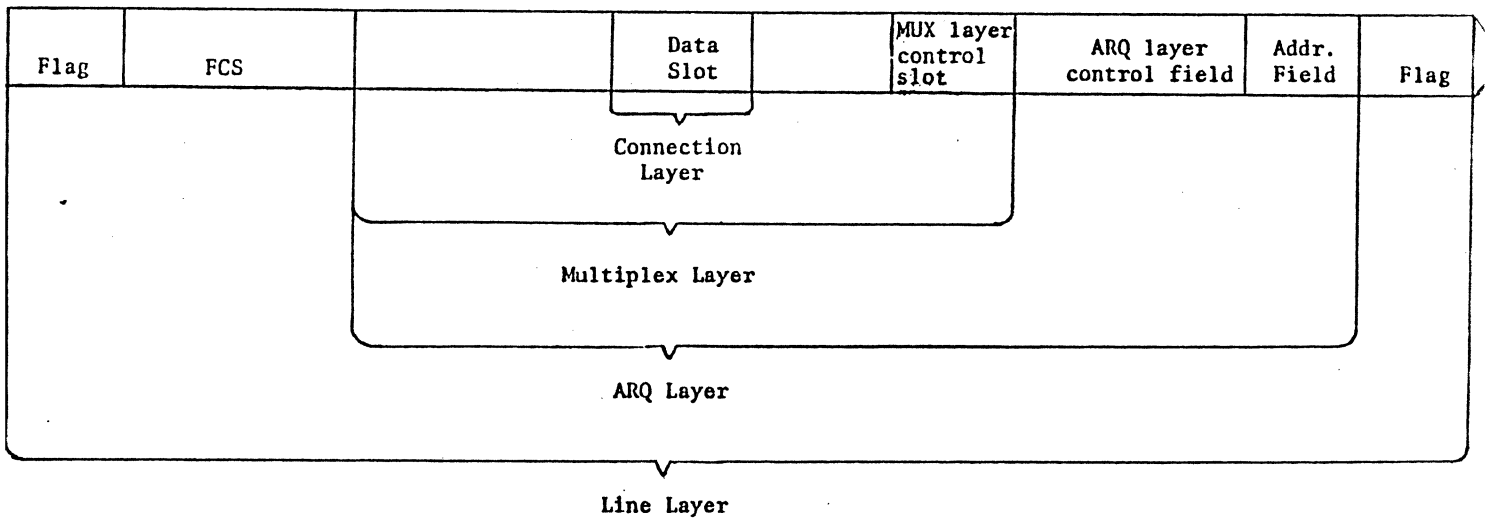
The Muxport protocol layers (reference the figure below) perform the following basic functions:

LINE LAYER - Provides the functionality equal to the HDLC, X.25 and SDLC line layers. It performs data transparency, idle fill, FCS generation, checking and zero insertion functions.

ARQ LAYER - This layer is compatible with the X.25 LAPB(E) Data Link Layer. This layer handles addressing and frame definition, through the control field bits, to define the frame types.

MUX LAYER - This layer multiplexes the customer data and combines it with control information. VCTP operation is handled in this layer.

CONNECTION LAYER - The Connection layer is responsible for end-to-end flow control, In-stream control operations for the ports (DPI, CSU, Autospeed). Customer data is sent to and received from the physical ports via the Connection layer.



3.2 LINE LAYER

The line layer achieves data transparency through the zero insertion technique (section 2.7). FCS calculation and verification are also done in this layer. Frames with a bad FCS are discarded at this layer and not passed to the ARQ layer. The Line layer also performs frame pacing and idle time fill. Idle time fill is done by transmitting flags between frames and by the transmission of empty I-Frames and or Receiver Ready (RR) frames between units. The 6000/6700 series products 'idle' in the following manner:

6005 - Sends RR's when connected to 6050, 6040, 6760. Connected to a 6740 it sends a RR then an empty I-Frame. (RR = 03 01 Nr BCC).
6050 - Sends empty data frames (I-Frames) (DATA = 03 Ns Nr 00 BCC).
6040 - Same as the 6050
6740 - Sends a RR then an empty I-Frame.
6760 - Same as the 6050

3.3 ARQ LAYER

The Muxport protocol ARQ layer contains the Address and Control field functions found in the X.25 and HDLC protocols. This layer is a subset of X.25 level 2. The address field is interpreted in the following manner: Address A = 03 = DTE. Address B = 01 = DCE (X.25 LAPB specifications). COMMANDS contain the address of the receiving unit. RESPONSES contain the address of the transmitting unit, (using Asynchronous Balanced Mode). An exception, the 6050 only sends Address A. The following frame types are used in Muxport Protocol:

control field									
FRAME TYPE	8	7	6	5	4	3	2	1	DESCRIPTION
I-FRAME	Nr			P/F	Ns			0	Information
S-FRAMES	Nr			P/F	0	0	0	1	RR (Receiver ready)
	Nr			P/F	0	1	0	1	RNR (Receiver not ready)
	Nr			P/F	1	0	0	1	REJ (Reject)
U-FRAMES	0	0	1	P	1	1	1	1	SABM (async balanced mode
	0	1	1	P	1	1	1	1	SABME (SABM - extended)
	0	1	1	F	0	0	1	1	UA (unnumbered Ack.)
(see below)	1	0	0	F	0	1	1	1	FRMR (Frame reject)

3.3.1 FRAME REJECT COMMAND

The Frame Reject (FRMR) command is used to report error conditions not recoverable by retransmission of the identical frame. The FRMR is usually caused by one of the following:

- o The reception of a command/response that is invalid or not implemented
- o The reception of an I-Frame with an information field exceeding the maximum limit 255 bytes/port or 16k/frame.
- o The reception of an invalid Nr count (not the next sequential I-Frame or one already acknowledged)

An information field which immediately follows the Frame reject control field provides the reason for the rejected frame.

Normal Mode

FRMR	8	7	6	5	4	3	2	1	
Byte #1	1	0	0	F	0	1	1	1	(FRMR Command)
Byte #2	Rejected Control field								
Byte #3	Vr			C/R			Vs		
Byte #4	0	0	0	0	Z	Y	X	W	

Where:

Byte #1 = The Frame reject command control field.
Byte #2 = The control field of the frame that caused the reject.
Byte #3 Vs = The current value of the XMT state variable
Vr = The current state of the RCV state variable
C/R = Indicates if the frame rejected was a command (0) or a response (1).
W = If set to "1" indicates that the control field received was invalid
Byte #4 X = If set to "1" indicates the control field received is invalid because an I-field (data) was in the frame and the Frame was an S or U-Frame. Bit "W" must be set to 1 in conjunction with this bit being set.
Y = If set to "1" indicates that the I-field received exceeded the maximum allowable limit. 255 bytes per port and 16k per I-Frame
Z = If set to "1", indicates the control field received contained an invalid Nr count. This bit is mutually exclusive with the "W" bit.

Extended Mode

The meaning of the fields is the same as those above (X = undefined).

FRMR	8	7	6	5	4	3	2	1	
Byte #1	1	0	0	X	0	1	1	1	(FRMR Command)
Byte #2	0	0	0	0	0	0	0	F	
Byte #3	First byte of rejected control field								
Byte #4	Second byte of rejected control field								
Byte #5				Vr				0	
Byte #6				Vs				C/R	
Byte #7	0	0	0	0	Z	Y	X	W	

3.3.2 REMOTE RESET (BOOT) SEQUENCE

Link set up is normally achieved through the use of SABM(E) commands. This assumes that each unit is operating and able to receive the command from the remote. At times this is not the case, therefore a mechanism is available to gain control of the remote unit. This mechanism allows a unit to reset the remote hardware and gain control of the unit. The remote reset frame is always discarded by the line layer. If the remote reset circuitry is disabled the frame is totally ignored.

The remote reset is accomplished using a sequence of bits which can be detected by the remote units hardware. The following figure explains the remote reset sequence:

ABORT	FLAG	55	8D	18	CC	B4	3B	1C	12	89	64	2E	C2	32	35	00	XX	YYY	ABORT

Where:

YYY = any number of bytes
XX = command field with the following meaning

Command Field	Unit B	Unit A
Go to loopback	BA	B8
I am in loopback	47	45
Go to normal	46	44

3.3.3 ARQ LAYER SYSTEM STATES

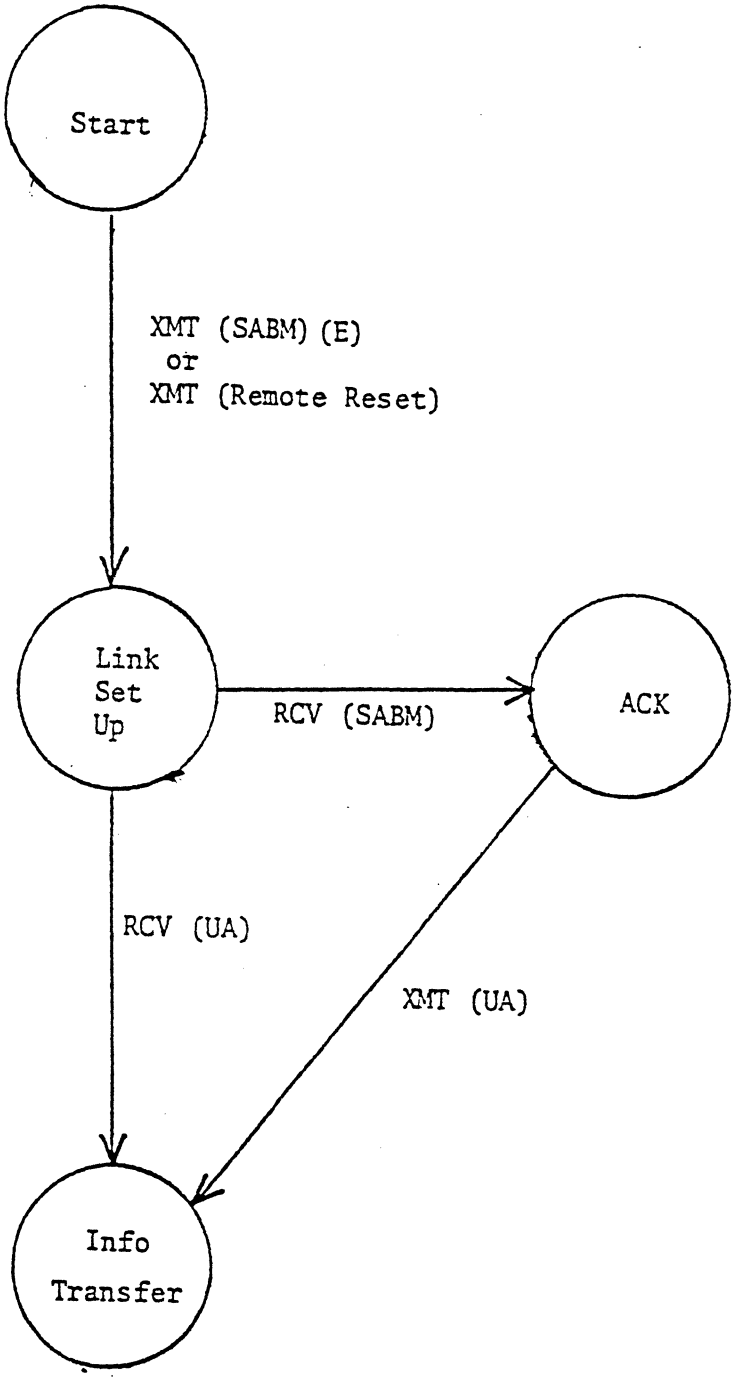
The ARQ layer is organized around two main states: the LINK SET-UP state, which is used when requesting link establishment or reestablishment and the INFORMATION TRANSFER state which is entered after the Link Set-Up state is completed. To control the operation of the layer the CMP maintains several 'system' parameters they are:

- o T1 TIMER - Used during Link Set-Up to check for its completion. If completion is not complete upon T1 expatriation the Link Set-Up procedure (SABM or SABME) is restarted. T1 is a 3 second timer.
- o T2 TIMER - Used during the Information Transfer state as a framing timer. When a new frame is transmitted T2 is started, and is not stopped until a valid acknowledgement for the frame is received. If T2 expires without the ACK, the link set up procedures are started (SABME sent). T2 is set at 20 seconds.
- o N2 COUNTER - Used in the link set up phase. The maximum number of SABM(E)'s sent before transmitting a remote reset. N2 = 20 tries
- o N1 COUNTER - Is the maximum number of bits allowed in a frame. If exceeded the frame is rejected. N1 = 16,0000 bits/frame.
- o RC COUNTER - The retransmission counter, or the number of times to retransmit a frame. RC = 20 retransmits.

3.3.3.1 LINK SET UP STATE

The figure on the next page shows the state diagram for link set up The table below summarizes the the transitions (state changes) that may occur during the Link Set-Up phase.

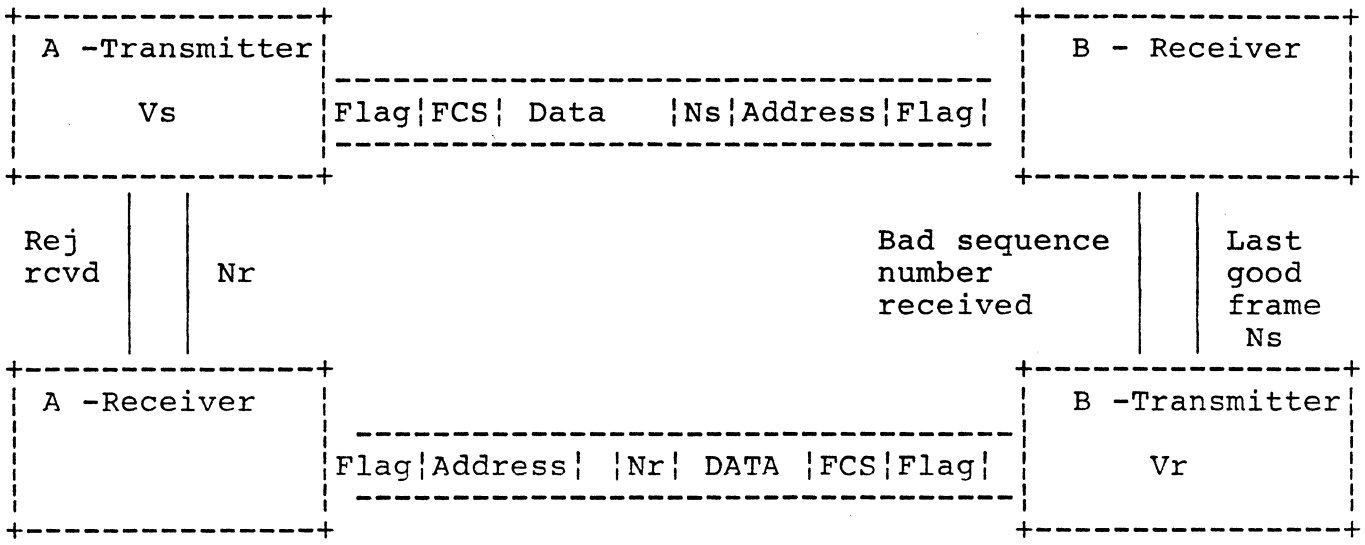
EVENT/RCV	ACTION	NEXT STATE
Any frame other than SABM(E)/UA	Ignore do not stop T1	LINK SET-UP
SABM(E)	Reset T1 Vs & Vr set to 0 Send UA	INFO TRANSFER
UA	Set Vs,Vr = 0 Reset T1	INFO TRANSFER
T1 expires AND RC less than 20 RC not equal to N2	Send SABM(E) Start T1 RC = RC+1	LINK SET-UP
T1 expires AND RC = 20 = N2	Send remote reset Start T1, Reset RC	LINK SET-UP



Normal Link Set Up

3.3.3.2 INFORMATION TRANSFER STATE

There are two sub-states to this state: the Transmission sub-state and the Retransmission sub-state. The figure below illustrates the Muxport operation during both data transfer operations. Unit "A" is the local unit, while unit "B" is the remote unit. The descriptions are from unit A's perspective.



Dialog between units A & B will conform to the following steps:

1. Unit A has a complete I-Frame to transmit.
2. Unit A sets Ns = Vs and Nr = Vr.
3. Frame Ns is stored in the retransmit buffer in unit A. The Nr count and FCS are not stored because they are updated if the frame is retransmitted.
4. Unit A checks to see if Vs is equal to the ARQ number (K) (7 or 127 depending on mode). If it is not equal the frame is transmitted to unit B and Vs is updated. Unit A Unit A then returns to Step #1.

--- EXCEPTIONS ---

4a. If Vs is equal to the maximum number of outstanding frames (K). {last received Nr + K = 7 or 127 depending on mode} Unit A will not transmit any new frames but will start start retransmission, starting with the first unacknowledged frame (at this point the retransmission sub-state is entered).

4b. If the retransmit buffer is full or a Reject frame is received from unit B the Retransmit state is also entered.

5. If unit B correctly receives the I-Frame this is directly communicated to the unit A transmitter via the unit B Nr field of a supervisory frame (RR, RNR or REJ) and is indirectly conveyed to the A transmitter by the Nr field of I-Frames received by unit A.
6. Upon reception of an frame from unit B containing a valid Nr count, unit A will consider the Nr an acknowledgement for all transmitted I-Frames by unit A with a Ns up to and including Nr -1.

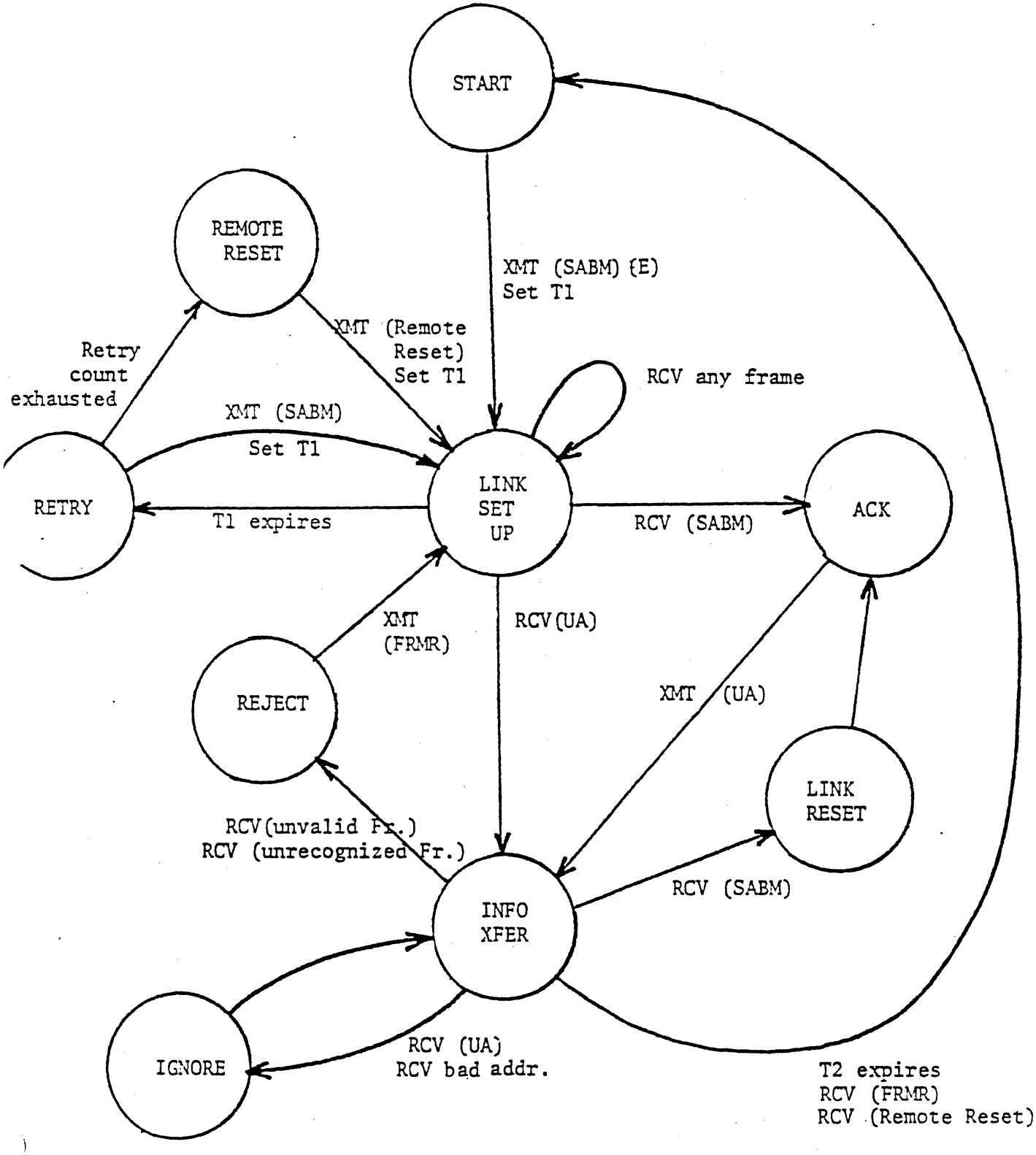
3.3.3.3 RETRANSMISSION SUB-STATE

While transmitting I-Frames any of the following may occur which will cause unit A to enter the retransmission sub-state and begin retransmission of all frames in the retransmit buffer.

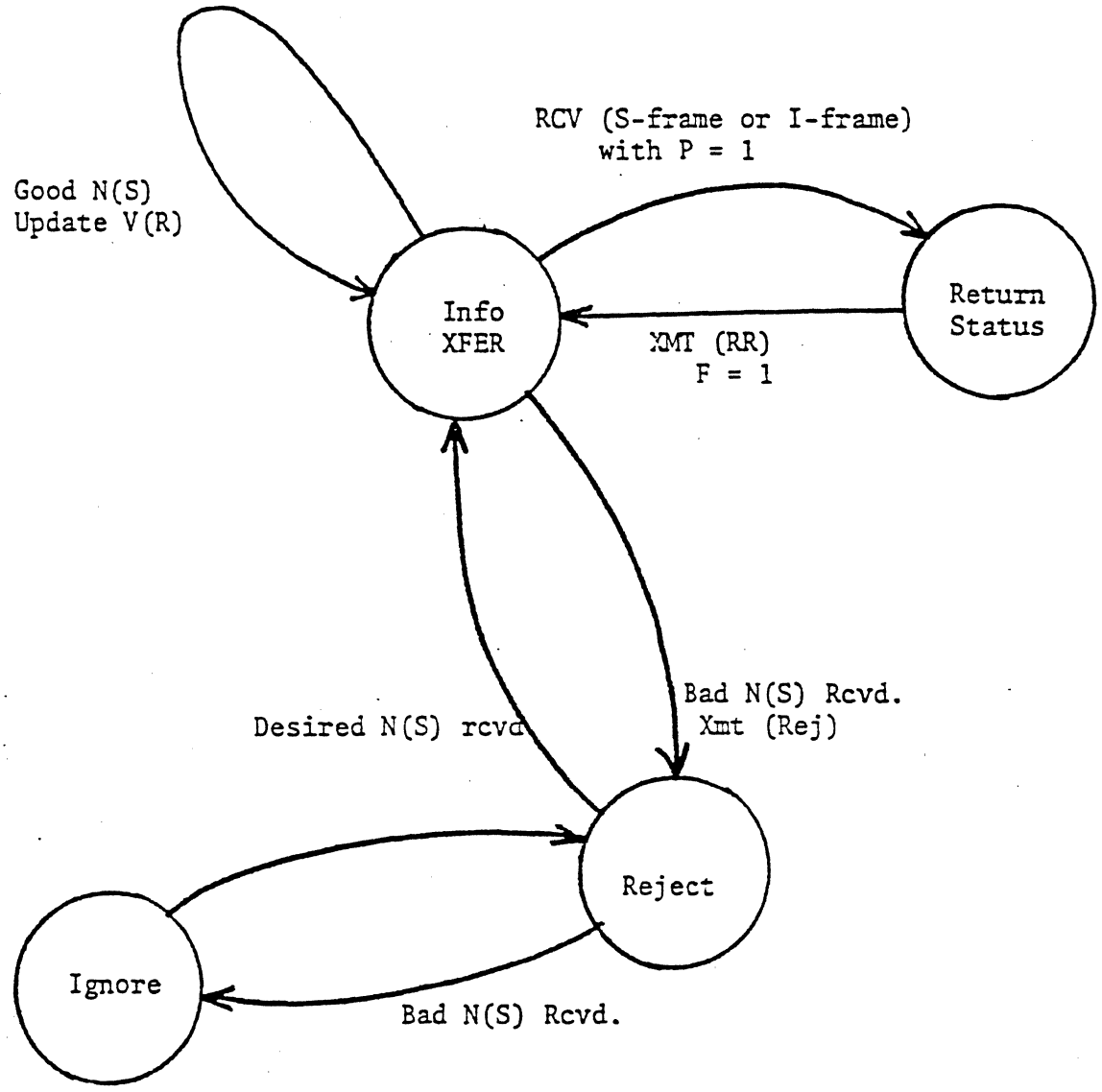
- o The retransmit buffer becomes full.
- o The maximum number of outstanding frames is reached. (many short frames or circuit delay is too long)
- o A reject frame is received from the remote unit.

Exit from the retransmission sub-state (returns to the transmission sub-state) occurs when the end of the retransmit queue is reached (all frames retransmitted) and the frames are acknowledged. Unit A will continue in the transmit sub-state where it left off.

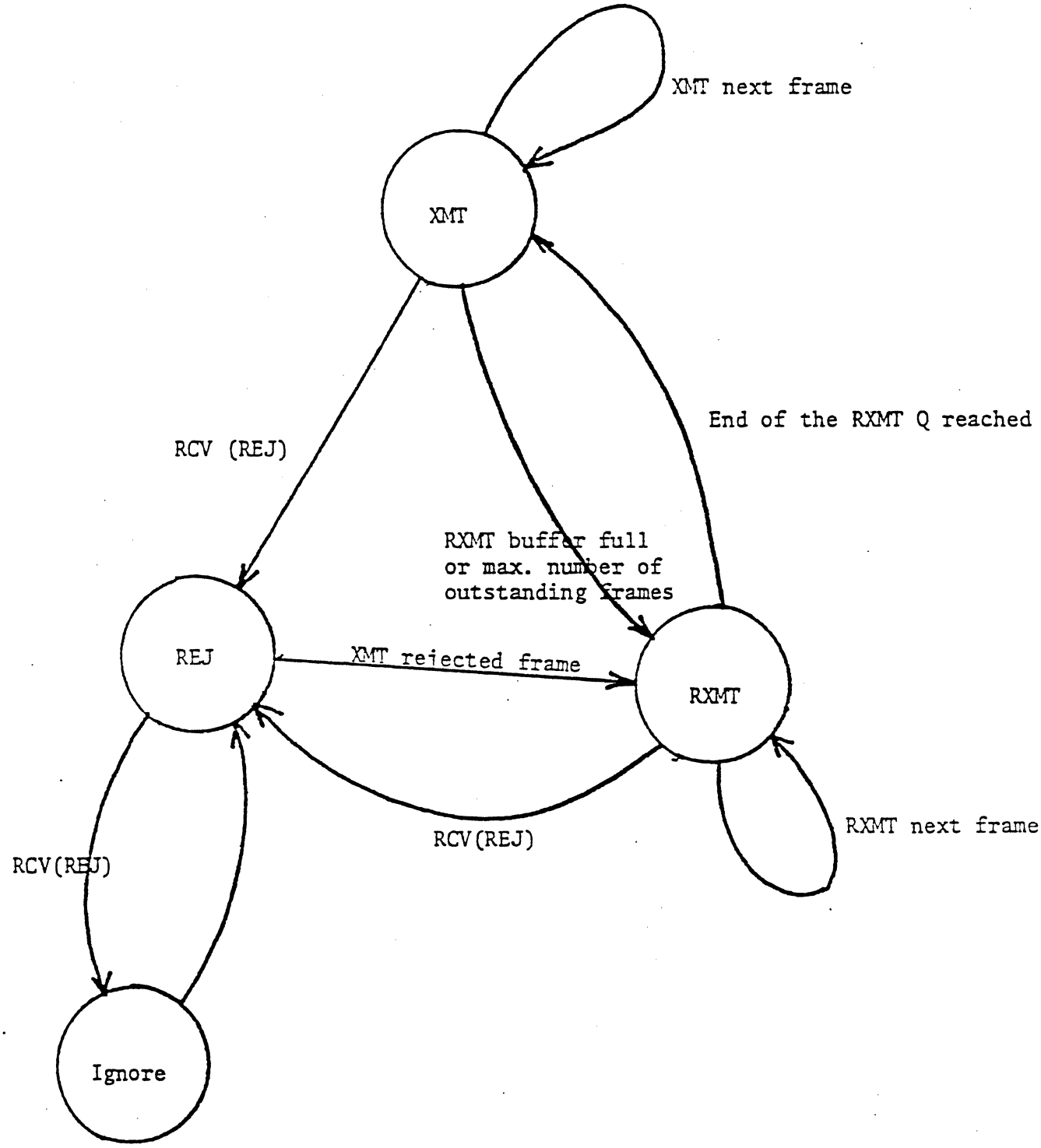
The following state diagrams represent the Information Transfer state (Sending and Receiving) and the Link Set Up state with exception (error) transitions from the Information Transfer state.



LINK SET UP STATE and
EXCEPTION TRANSITIONS FROM INFORMATION TRANSFER STATE



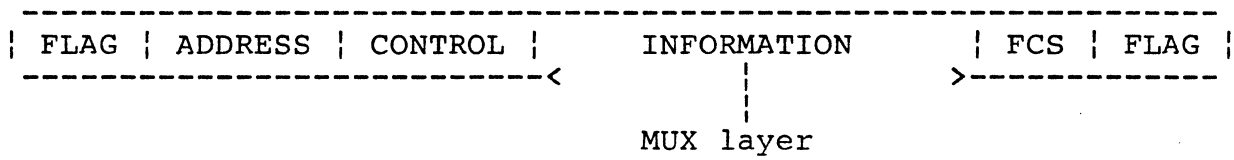
INFORMATION TRANSFER STATE: RESPONDING OPERATION



INFORMATION TRANSFER STATE: INITIATING OPERATION

3.4 MUX LAYER

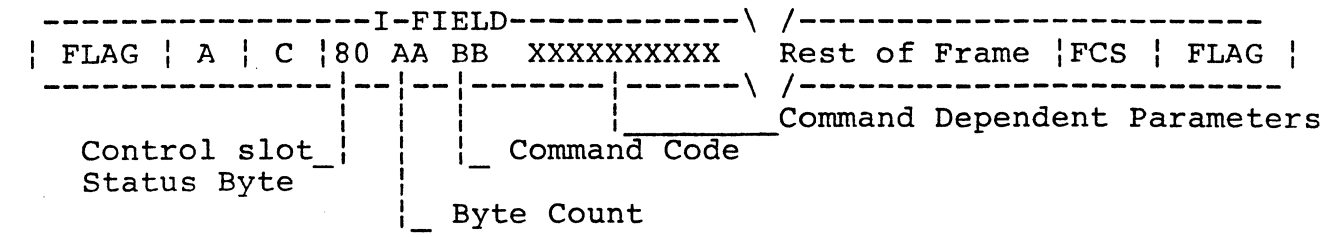
This layer compiles the information field (data) of the Muxport I-Frame, it is responsible for the format of data. The MUX layer combines (multiplexes) the customer data and supervisory information from the Connection layer, adds control information for the remote unit (if any) and sends the complete packet to the ARQ layer for integration into a HDLC frame. Reference the figure below.



The first byte of the information field (MUX layer) is the CONTROL SLOT STATUS BYTE. This byte is used to inform the remote unit if the data following is control information (VCTP, system reports) or customer data. The MSB of this byte determines how the bytes to follow are to be interpreted. If the MSB is set (X'80) a control slot is present. If the MSB is not set (X'00') then customer data will follow.

3.4.1 CONTROL SLOTS

Control slots always begin with a Control Slot Status Byte of X'80'. This informs the remote unit that the data following this byte is 'Higher Level' control information. The byte immediately following the Control Slot Status Byte is the byte count. The byte count contains the count of the number of control bytes to follow. The third byte is the Control Slot Command Code, this byte identifies the type of control slot. The rest of the bytes (according to the byte count) are command code dependent parameters/information. The following figure illustrates a Command Slot lists the Command Codes.



Where:

80 = Control Slot Status Byte (MSB set) indicates control information follows.

AA = Byte Count, the number of control information bytes to follow.

BB = Command Code indicates one of the following commands:

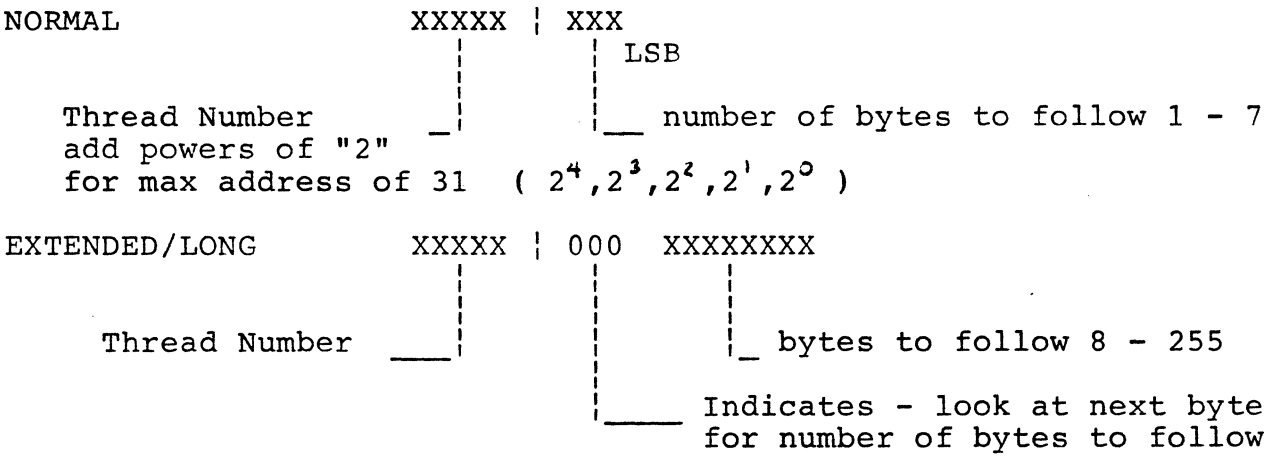
0 Error Message Report
1 Read Port Configuration Parameters
2 Return Port Configuration Parameters
3 Write Port Configuration Parameters
4 Write Acknowledge Port Configuration Parameters
5 Read Port Statistics
6 Return Port Statistics
7 Read Statistics Threshold
8 Return Statistics Threshold
9 Write Statistics Threshold
A Write Acknowledge Statistics Threshold
20 CTP Command
21 CTP Response (continuing)
22 CTP Response (completed)
23 Unsolicited Error Messages and Reports
24 End of VCTP
25 Address Packets
27 6740 Protocol Announcement
29 6005-6005 Move TP Configuration Command

3.4.2 DATA SLOTS

A Data Slot is indicated by a Control Slot Status Byte equal to 00. The data field is subdivided into slot groups which are further divided into slots containing data for specific ports.

There are eight slot groups available, each one is designated by a Slot Group Header (SGH) numbered 00 thru 07. Currently only slot group 00 is used. It is assumed that slot group 00 will immediately follow the Control Slot or Control Slot Status byte if 00, thus the SGH for Group 00 is optional. The SGH may be used as an idle fill marker between data slots within the Slot Group.

Following the SGH (if present) are the individual data slots. Each data slot starts with a one byte Slot Header indicating the thread number and the number of bytes to follow. There are two formats for the Slot Header, normal and extended or long. The normal header is used when there are 1-7 bytes of data for the port, while the long format is used for slots of 8-255 bytes, this format requires two bytes. The Slot Header format is shown below:



Sample Data Slot

Flag	A	C	I - Field	FCS	Flag
7E	03	0Ns Nr	00 77 41 42 43 44 45 46 47	xx xx	7E

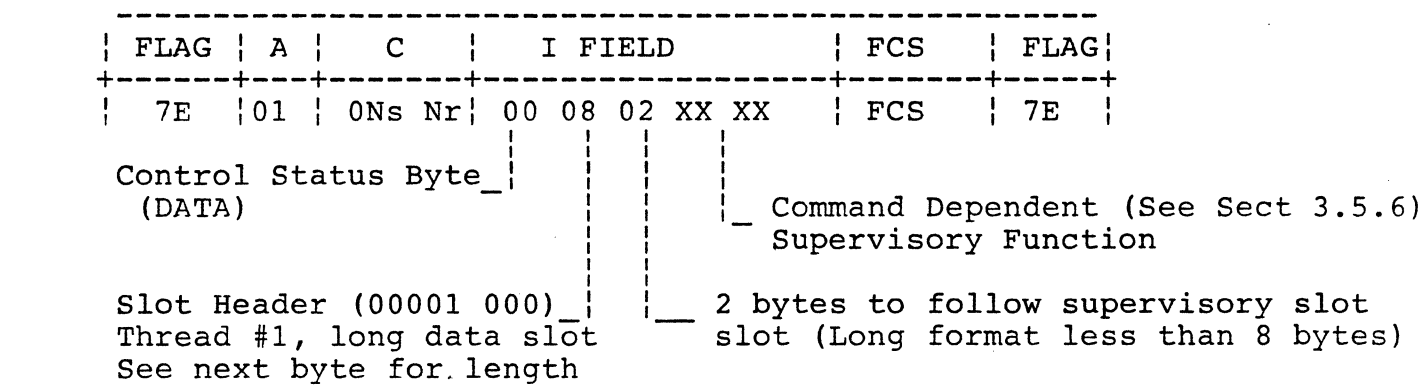
"A" — I-Frame

Control Status Byte — Thread 14, 7 bytes to follow 77 = 01110 111

7 bytes data (ABCDEFG)

3.4.3 SUPERVISORY SLOTS

Supervisory Slots are used to pass end-to-end flow control information for specific ports. The flow control data is generated in the connection layer but formatted and inserted into the frame by the MUX layer. The MUX layer interleaves the supervisory slots with data slots. Supervisory slots consist of a slot header using the extended format followed by 1-7 message bytes. The number of bytes to follow is the only way to distinguish a supervisory slot from a long data slot because by definition a long data slot must contain more than 7 bytes of data. The following figure gives an example of a Supervisory slot:



3.5 CONNECTION LAYER

The Connection layer is the highest layer of the CMP. This layer is responsible for the actual transfer of customer data. In Muxport Protocol, data is transferred as eight bit bytes regardless of its original length. If necessary the high order bits are filled with zeroes. Data is passed from end-to-end without alteration, the one exception is the data sequence X'01' which is passed as a two byte sequence X'01 01'.

The Connection layer also handles the Control Signal Updates, Data Path Initialization, Break, Data Slot Termination, Autospeed and end-to-end Flow Control. These events use a special escape sequence (X'01') to distinguish them from data and are embedded within the data stream. They are known as In Stream Control Codes or ISCC's.

The escape sequence is used to pass control information (ISCC) within the data stream. The (01) sequence indicates that neither it nor the following byte is part of the data. The byte following the escape is to be interpreted as a special function. An ISCC applies only to the port owning the data slot in which the sequence appears.

3.5.1 CONTROL SIGNAL UPDATES (X'01 1n')

Control Signal Updates (CSU) are used to pass the value of the terminal or modem signals, RTS, DTR, MB, across the link whenever a change is detected by the local node. All CSU's begin with the sequence (X'01') followed by a second byte identifying the signal(s) being updated. The local unit may request a CSU from the remote via the Control Signal Update Request (CSUR) command. A CSUR uses the X'01 40' sequence.

Each unit maintains an internal control signal register (the 1n byte). Whenever a change is detected the entire set of signals is sent across the link. A bit set indicates the signal is 'high' while a bit not set (zero) indicates a signal is 'low'. The following table explains the significance of the bits in the X'1n' byte and internal register:

1n where n = 0 MB RTS DTR	
01 10	All signals low
01 11	DTR high (bit 0)
01 12	RTS high (bit 1)
01 13	RTS & DTR high
01 14	MB high (bit 2)
01 15	DTR & MB high
01 16	RTS & MB high
01 17	Not used for CSU's
01 40	CSU request to remote

3.5.2 DATA PATH INITIALIZATION (X'20' and X'30')

Because of software buffers in the system it is possible for user data to become trapped, i.e a dial up user is disconnected abnormally. If another user connects to this port he would get the previous users data. To prevent this two ISCC's are used to flush the virtual circuit. The two functions are X'20" Data Path Initialization (DPI) and X'30' Data Path Initialization Acknowledge (DPIA).

When a high to low transition of DTR is detected, it is assumed that the user is disconnecting. This event causes a DPI (X'01 20') to be sent to the remote unit following a CSU. This sequence implies that no more data will be output for the port, excluding control information. All Data in the local units buffers is flushed. The remote unit must respond with a DPIA (X'01 30') indicating that the virtual circuit is cleared in both directions. If a DPIA is not received within 40 seconds the DPI is reissued by the local unit.

3.5.3 START, STOP BREAK (X'7n' and X'80')

If the local node detects a break condition (constant space) the ISCC functions X'7n' and X'80' are used to relay this information to the remote end. The X'01 7n' function is used to initiate the break. The time period (length) of the break is "n" character times. If n = 0 the break is continuous and should not be stopped until the X'01 80' sequence is received. A received X'01 80' following a X'01 7n' sequence where "n" is non zero is discarded.

3.5.4 DATA SLOT TERMINATION

The Muxport adds a protocol specific terminator to the end of each ports data slot. The terminator signals the end of the ports allocated space in the specific Muxport frame. The ports transmission may be continued in another Muxport frame. The terminator is used between Muxport units only for internal data separation. The Muxport terminates a data slot in one of the following ways:

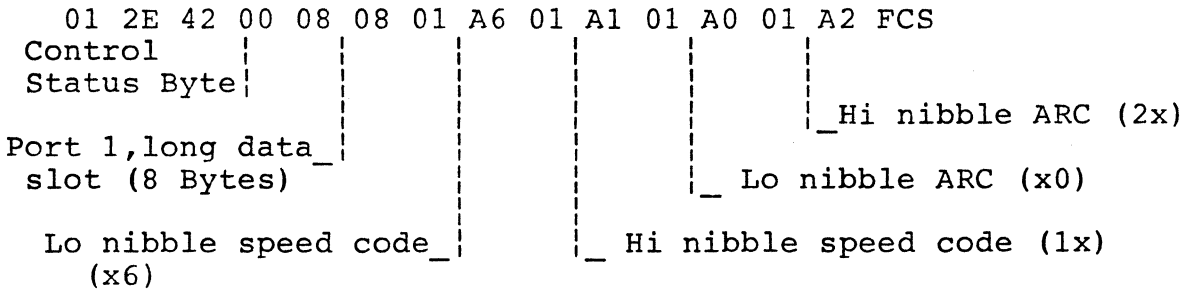
Slot Type	Termination sequence
Asynchronous	No termination required
Synchronous	X'01 17'
BOP (flag idle)	X'01 72'
BOP (mark idle)	X'01 72 01 17'
BOP (abort)	X'01 17'

3.5.5 AUTOSPEED (X'01 An')

When a thread in the local node receives an Autospeed recognition character (ARC), it transmits the data over the link to allow the remote unit to configure its corresponding port. This information is relayed to the remote unit via the ISCC sequence X'01 4n'.

Autospeed information requires 16 bits of information. The local unit must send an Autospeed Recognition Character and speed code (8 bits each). The data is sent four bits at a time, so four X'01 An' sequences are transmitted across the link for each Autospeed initialization. The ARC and speed code are sent over the link in the following order: Low nibble of speed code, High nibble of speed code, Low nibble of ARC and High nibble of speed code. As illustrated in the following figure:

Autospeed at 2400 bps



The speed code (16) corresponds to 2400 bps.
The ARC character (20) is a space.

SPEED CODES

SPEED	CODE	SPEED	CODE
50	01	1800	14
75	06	2000	15
110	08	2400	16
134.5	0A	3600	17
150	0B	4800	18
300	0E	7200	1A
600	10	9600	1C
1200	13	19200	1E

3.5.6 FLOW CONTROL

Flow Control (FC) information between Muxports is transmitted in Supervisory slots within the data stream, see section 3.4.3. Flow Control is used to regulate the flow of data between ports. Flow Control is done on an individual thread basis and is available for all threads connected to the Muxport.

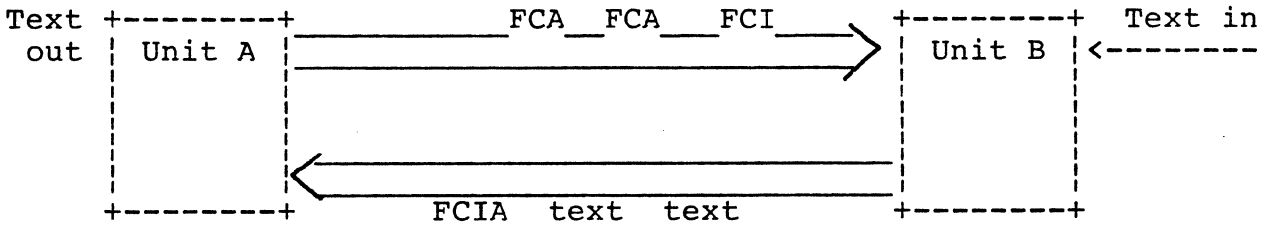
The thread assumes that it is not controlled until it receives the initialization sequence, Flow Control Initialization (X'10). A thread will request flow control status from the remote port during call creation via the Flow Control Initialization Request (FCIR X'12'). Possible responses to the FCIR are the FCI or a Flow Control Disable (FCD X'01 40') indicating that no FC is requested. The following Table summarizes the flow control handshaking sequences:

X'10'	FCI - (Flow Control Initialization) sent by the controlling port to initiate flow control.
X'11'	FCIA - (Flow Control Initialization Acknowledge) Affirmation sent by port to be controlled.
X'2n'	FCA - (Flow Control Authorization) Sent to controlled port (n+1) *16 = number of bytes authorized for transmission.
X'12'	FCIR - (Flow Control Initialization Request) Sent by controlled port asking to send more data.
X'01 40'	FCD - (Flow Control Disable) Signal to controlled port that data flow is not controlled.

3.5.6.1 FLOW CONTROL OPERATION

After the network has been booted and data is flowing between the circuits, a port with FC enabled detects that it must restrain the data coming from the remote port. The port becomes a CONTROLLING port by sending an FCI to the remote port causing it to become a CONTROLLED port.This sets the controlled ports authorization to zero causing data to stop flowing into the local (controlling) port. The controlled port responds with an FCIA. After receiving the FCIA the controlling port will send transmit authorizations via the FCA command. The remote (controlled) port will then be able to send data to the controlling port.

The controlled port will decrement its authorization every time it sends a character. If its authorization falls to zero it stops sending characters. If the port still has data to send (and the authorization count is zero) the port may send an FCIR to the controlled port to request additional authorization to send data. The port will remain controlled until the reception of an FCD from the controlling port. The following figure illustrates the flow control sequences for normal operation:



EVENTS

1. Unit A sends FCI
2. Unit B responds with FCIA
3. Unit A responds with FCA
4. Unit B sends text
5. Unit A outputs text
6. Unit A sends FCA
7. Unit B sends text

3.5.6.2 FC EXCEPTION CONDITIONS

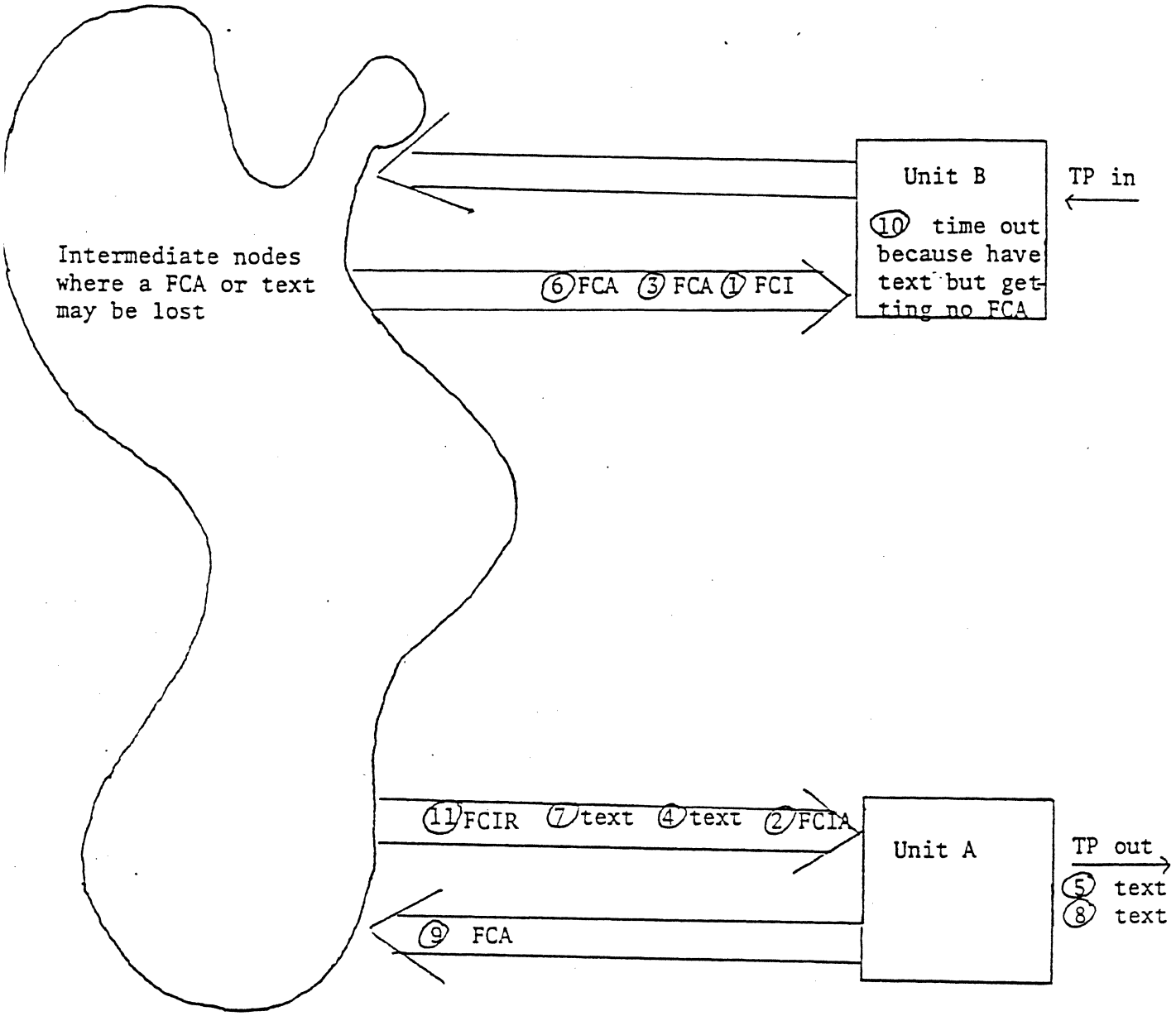
The controlling port should send an FCI after a disconnect has occurred or when a reconnect occurs, because at these times data flowing over the Muxport link is minimal and bandwidth is now available.

The only indication that a FCA's have been lost in the network is that a controlled ports authorization count reaches zero. The port will start a 40 second timer whenever its authorization is zero. When the timer expires the port will send an FCIR. If no FCI is received another FCIR is sent this will continue indefinitely. If the controlling port sends an FCI (in response to the controlled ports FCIR) but is restrained (via data restraint by the output device) it will be unable to send an FCA, the controlled port will then send another FCIR every 80 seconds until the controlling port can accept data.

Data received in excess of authorization sent via FCA's will cause the controlling port to send an FCI. The only external indication of this will be the controlling port buffer overflow.

The following figure illustrates flow control operation with FCA's lost:

Flow Control - A lost FCA resulting in a FCIR



Events

1) Unit A send FCI
2) Unit B respond with FCIA
3) Unit A responds with FCA
4) Unit B sends text
5) Unit A outputs text
6) Unit A sends FCA
7) Unit B sends text
8) Unit A outputs text
9) Unit A sends FCA but it is lost at an intermediate node
10) Unit B times out because its authorization is Ø, it has text and is getting no FCA
11) Unit B send FCIR
12) Restart at 1)

3.6 TYPICAL BOOT SEQUENCE

1. Local unit sends boot (Remote Reset) to remote unit. CC is the command code, see section 3.3.2.

ABORT FLAG 55 8D 18 CC B4 3B 1C 12 89 64 2E C2 32 35 00 CC nnnn ABORT

2. Local unit sends SABME's, as per T1 Timer, see section 3.3.3.

   03 6F 00 35 (6F is the SABME sequence)

3. Remote responds with SABME. A "UA" command is also valid here.

   01 6F 00 8D

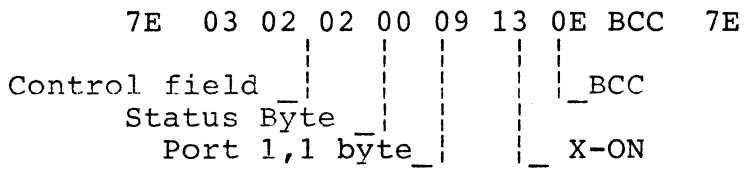
4. Local sends a UA to acknowledge the SABME and initiate data transfer phase.

   01 63 00 2D

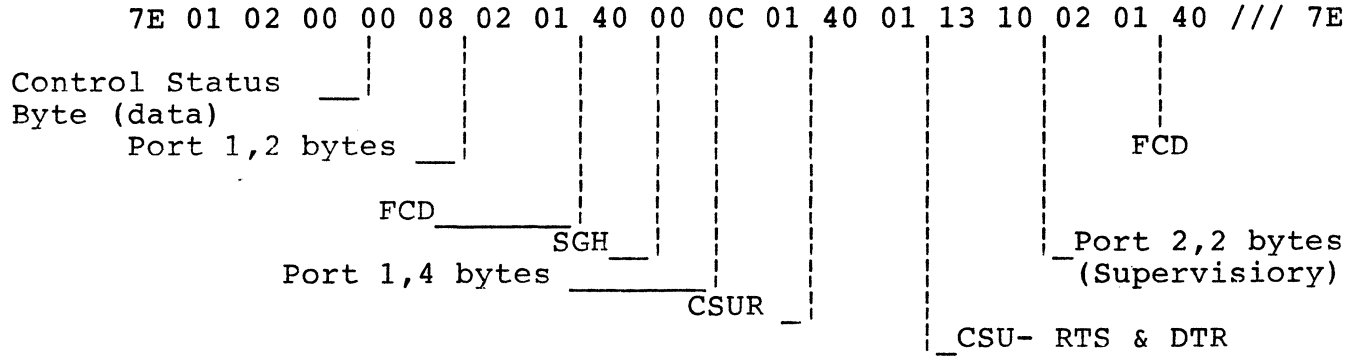
5. The local unit then sends a CSUR and updates the signal status for all threads configured on the Muxport (or NP for 6005)

   7E 03 00 00 00 08 0B 01 01 01 40 etc...

6. Local units async. threads on Muxport send XON sequence.



7. The remote responds with a flow control information for each port (in a supervisory slot). Also a CSUR and control signal state for each port is sent.



3.7 NOTES

- Address A = 03 = DTE - Address B = 01 = DCE per X.25 LAPB specifications. COMMANDS have the address of the Receiving unit and Responses have the address of the Transmitting unit, when using Asynchronous Balanced Mode (i.e. Muxport) EXCEPT for the 6050 who always acts like a master station and always sends Address A.

- Slot Group Header 00 is optional. The 6005 always sends it.

- Poll/Final bit formalities are sent by the Muxport protocol to comply with X.25 specifications, BUT the bits are not acted upon (used) by the Muxport.

- T1 Timer (link set-up timer) 3 seconds with carrier high and no response to link set-up request (SABME). The local unit will continue to transmit an SABM(E) every three seconds for twenty tries, will then issue a remote reset (Boot) sequence.

- T2 Timer (ARQ timer) after 20 seconds without an acknowledgement for transmitted frames the Muxport declares the link down and starts recovery procedures (sends SABME and starts T1 timer).

- The Ns counts will always be even for I-Frames, and odd for S and U frames.

JUPITER PROTOCOL

4.0 INTRODUCTION

Jupiter Protocol is an enhanced version of the Codex Multiplex Protocol. The Jupiter Protocol supports most of the facilities provided by CMP and features several extensions to the original protocol. The extensions are provided in order to support a 6740 Delta network configuration with each device having no more than two links running the Jupiter protocol, these extensions are:

- o Multiple path support for data transfer between nodes
- o Frame routing at the ARQ layer
- o Flow control on a per path basis

4.1 NPP OVERVIEW

The Jupiter protocol runs on the Network Port Processor (NPP) in the 6740. The purpose of the NPP is to transfer data between the Master processor (switch) and the configured links. There are four links per NPP over which frames can be sent or received. These links may be configured to use either the Jupiter or Muxport protocol. Other functions provided by the NPP include: link connection establishment, thread and link rerouting and acting as an endpoint for Muxport thread connections.

Additionally the NPP invokes the 'Extra Functions' software modules for threads configured on Muxport links. These Extra Functions perform loopback, data monitoring, fox diagnostics, datagram services and UDR conversation sensing. These software modules are configurable on a thread by thread basis and only for Muxport threads.

4.1.1 NPP RECEIVE DATA FLOW

The NPP receives the user data from the link and must pass it to the switch processor for distribution to the TP's.

Data reception begins when a valid HDLC address is received by the NPP. If the frame is destined for another node (Jupiter protocol only) the frame is received, held in a buffer until it is complete, (no error checking is done) then sent over the appropriate link to its final destination.

If the frame is for the local node it is scanned by the NPP for frame type (I, S or U). U and S-Frames are processed immediately by the protocol link layer, then discarded. I-Frames are passed to the Data level (also within the NPP). The I-Frames are demultiplexed, flow control information reformatted, extra functions invoked for Muxport threads and sent to the Switch processor for distribution to the TP's.

4.1.2 NPP TRANSMIT DATA FLOW

The transmit data flow occurs in the following sequence:

1. NPP obtains data for the specified link and path.
2. The data is 'packed' (stored) for the frame.
3. Extra functions are performed for Mux links.
4. The data is multiplexed into data slots.
5. Flow control information is added, (supervisory slots).
6. Control slots are added (if required).
7. The HDLC address and sequence numbers are added.
8. The frame is transmitted over the link.

Sequence numbers are added to frames corresponding to the path number taken. This allows the NPP to verify, at the frame level, the sequence numbers of the frames being passed. In the case of path 0 frames (point-to-point or Muxport) the sequence number is equal to the final sequence number. See Section 4.2.2.2.

4.2 PROTOCOL DESCRIPTION

4.2.1 LINE LAYER

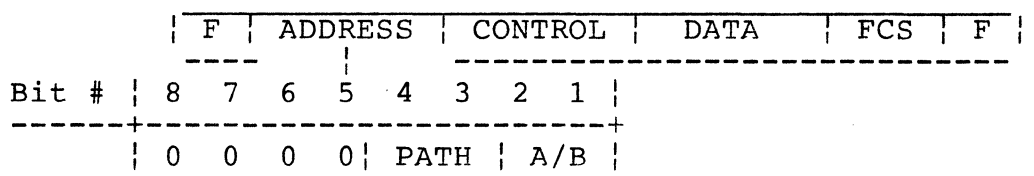
The line layer in the Jupiter protocol is identical to the line layer in the Codex Muxport protocol, see Section 3.2 for a description of this layer.

4.2.2 ARQ LAYER

The Jupiter protocol retains the main features of the CMP ARQ layer (Section 3.3) but only the Extended mode of operation is supported. There are enhancements to the ARQ layer for the Jupiter protocol, they are: Paths, frame routing and link failure recovery. The following sections describe the changes that Jupiter protocol makes to the ARQ layer of CMP.

4.2.2.1 ADDRESS FIELD

The Jupiter Protocol address field has the following format:



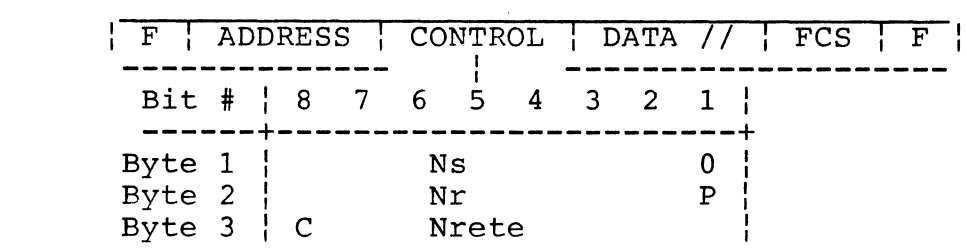
Where:

Bits 8 - 5 = 0 and are unused
Bits 4 & 3 = Path number in the range of 0 - 2.
Bits 2 & 1 = Address A or B (same as CMP)

The Path number field is a new addition to the Address field. Path 0 is used for frames sent directly to the destination node without passing through an intermediate node (standard Muxport). Path 1 is used for frames sent from the source node to the intermediate node. Path 2 is for the frames sent from the intermediate to the destination node. Frames cannot travel over more than two links to reach their destination.

4.2.2.2 CONTROL FIELD

Only the extended mode of control field operation is supported in Jupiter Protocol. The control field for Path 0 is the same as in Muxport protocol. For Paths 1 & 2 the control field has the following format:

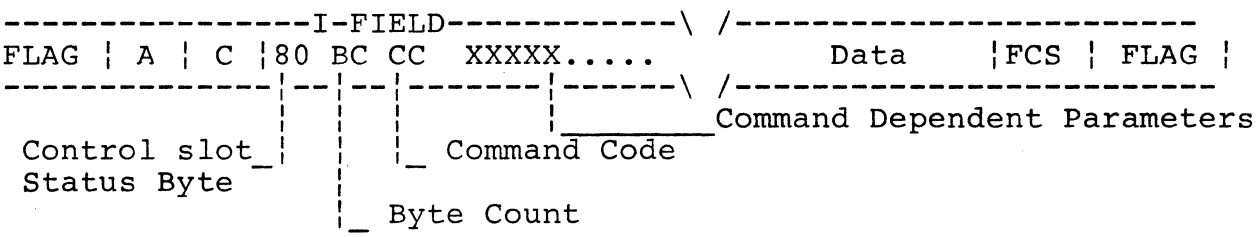


Where:

Ns = The send sequence number
Nr = The receive sequence number
P = The poll/final bit
Nrete = The end-to-end sequence number (Section 4.3)
C = If set a control slot follows

4.2.3 MUX LAYER

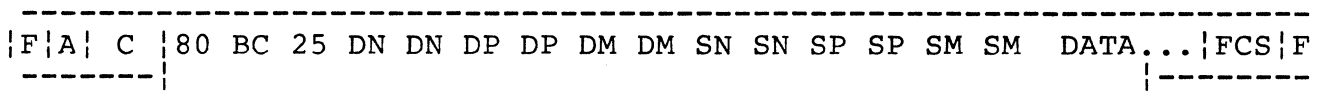
The Mux layer applies to I-Frames only. This layer differs from CMP in that there are two new control slot frames, defined below. The control slot frames conform to the Muxport protocol format defined in section 3.4.1. The following figure illustrates the control slot format in the Jupiter protocol and lists the valid command codes:



Command Code	Definition
20	VCTP command
21	VCTP response
23	Unsolicited report
24	Quit VCTP mode
25	Jupiter address packet
27	Jupiter protocol announcement

4.2.3.1 JUPITER ADDRESSED PACKET (CC = 25)

The Jupiter Addressed Packet is used to perform the control functions of the NPP these include: booting the processor, link or thread, statistics, connection establishment, error reports, diagnostics and link failure recovery. The Addressed Packet control slot uses the following format:



Where:

80 = Control status byte, indicating control information
BC = Byte count (not including self)
25 = Command code for addressed packets
DN = Destination node (2 bytes) If the MSB is set an addressed packet error has occurred.
DP = Destination port (2 bytes)
DM = Destination software module (2 bytes)
SN = Source node (2 bytes). MSB = 1 for response, 0 for command
SP = Source port (2 bytes)
SM = Source software module
Data = Up to 242 bytes of command dependent data

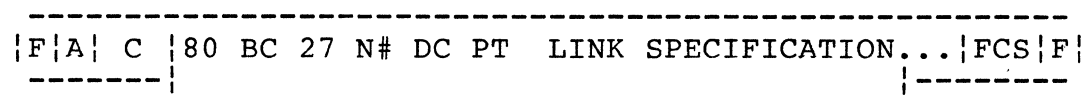
4.2.3.2 LINK SET-UP PROCEDURE (CC = 27)

When an NPP link is brought up the #27 control slot or protocol announcement (PA) is sent to the remote, indicating the senders identity and defining the link parameters. The NPP always sends a PA upon link initialization even if it is configured as a Muxport device. Upon reception of the protocol announcement the remote unit will do one of the following:

1. If the receiver is a Jupiter node, the control slot is handled by the CPS (switch) which will decode the slot into link status information and pass the data back to the NPP. The remote will respond with its own control slot of 27 and the units will enter Jupiter protocol data transfer.

2. In the case of a non-Jupiter device receiving a 27 control slot, the device will reject the control slot. The control slot will be returned to the original device with the command code set to A7 (this is a 27 and the MSB of the byte is set). The rejected control slot is sent to the switch for decoding. The switch informs the NPP to enter Muxport mode of operation (if the NPP is configured as a Muxport).

The protocol announcement command slot has the following format:

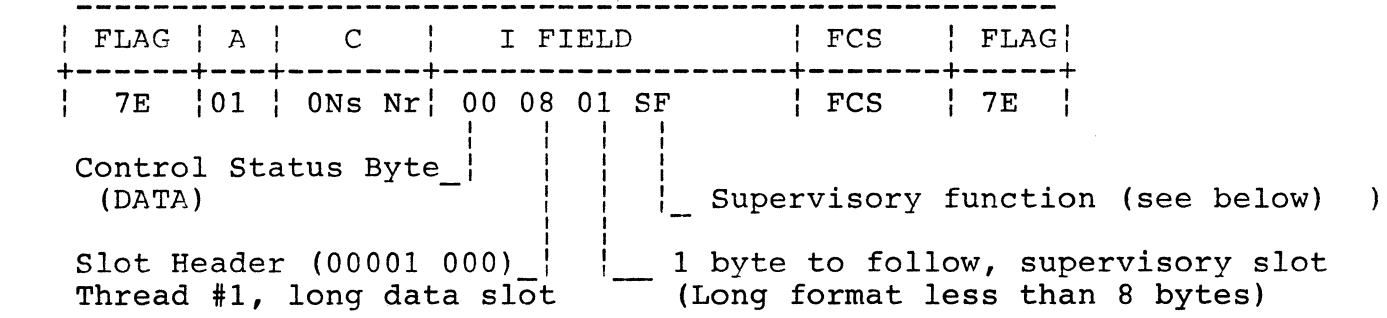


Where:

80 = Status control byte, indicating control information
BC = Byte count (not including self)
27 = Command code for protocol announcement
N# = Source node number
DC = Device code (internal Jupiter use)
PT = Protocol type 0 = Empty
1 = IPL
2 = Muxport
3 = Jupiter
Link Specification = data about the source processor.

4.2.4 CONNECTION LAYER

The Connection Layer is the innermost layer of the I-Frame in the Jupiter protocol. This layer handles the actual data and control information pertaining to specific terminal ports. This layer is equivalent to the Muxport protocol Connection Layer in Section 3.5. Jupiter Protocol contains additional supervisory command functions. The new commands are used for thread rerouting and link cancelation in Jupiter delta networks. The commands use the following format:



Jupiter Supervisory Functions

80	Clear Reroute - reroute a thread going to the remote unit
81	Clear cancel - request cancelation of reroute (DPI)
82	Clear previous - data preceeding this slot to be flushed.
85	Clear previous acknowledge
86	Clear cancel acknowledge (DPIA)
87	Clear reroute acknowledge
FF	NP data loss

4.3 PATH DEFINITION

Paths are unique to the Jupiter protocol. A path defines the specific routing information for traffic between nodes in a delta network. A path serves to identify the route a frame travels between nodes. End-to-end acknowledgement of traffic is available on a per path basis via the Nrete variable. This Nrete also provides for path flow control via a window function, see section 2.6. Paths have the following characteristics:

- o Each path is uni-directional.
- o Two paths are required for two-way traffic.
- o There may be more than one path between endpoint nodes.
- o Multiple slots may travel on the same path.

4.3.1 PATH NUMBERING

Path numbers are used to identify paths. A path number of '0' is used on frames sent directly from source node to destination node. Path '1' is used for frames sent to an intermediate node for retransmission to the final destination. Path '2' is used for frames that are transmitted from the intermediate node to the destination node.

4.3.2 FRAME ROUTING

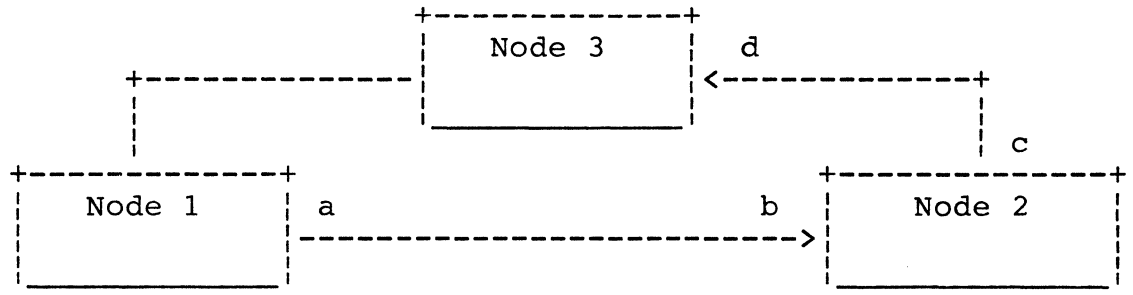
Upon arrival at a node, frames on paths 0 and 2 do not require further routing. These frame have arrived at the final destination node. Path 0 and 2 frames are removed by the NPP for processing and distribution to the TP's. When data on path 1 is received the intermediate node will do the following:

1. A new local path number of '2' will replace the path number of '1' on the incoming frame (address field).
2. The frame will be transmitted on the outgoing link designated for this transmit path.

4.3.3 PATH ENABLING

At start up time all paths going through a link are in a disabled state. A disabled path is one where data flow is inhibited. When the Jupiter link comes up, path '0' (in both directions) is automatically enabled by the NPP. The enabling of paths 1 & 2 require explicit addressed packets from the routing manager (in the switch), these packets are called Enable Path packets. When the Enable Path packet is received all the variables for the path concerned are reinitialized. Data flow is permitted on an enabled path only.

The interaction to enable paths 0, 1 and 2 is shown below. A delta network is assumed. Link 'a' of Node 1 is connected to Link 'b' of Node 2 and Link 'c' of Node 2 is connected to Link 'd' of Node 3. The topology is illustrated below:



The interaction to enable the paths from node 1 to node 3 is:

```
|------- Node 1 -------><------- Node 2 -------><------ Node 3 ------|

            Jupiter link set-up is complete for all nodes

Link 'a' path 0 enabled

Link status broadcast ----->

 Enable path AP (addressed packet)
 link 1 path 'b' --------->
                    <---------Path Enabled

                              Enable Path AP --------->
                              Link 'c'path 2
                                          <----------Path Enabled

                              Enable Path AP ---------->
                              link 'd' path 2

                                           <----------Path Enabled
                  <----------- Path enabled
```

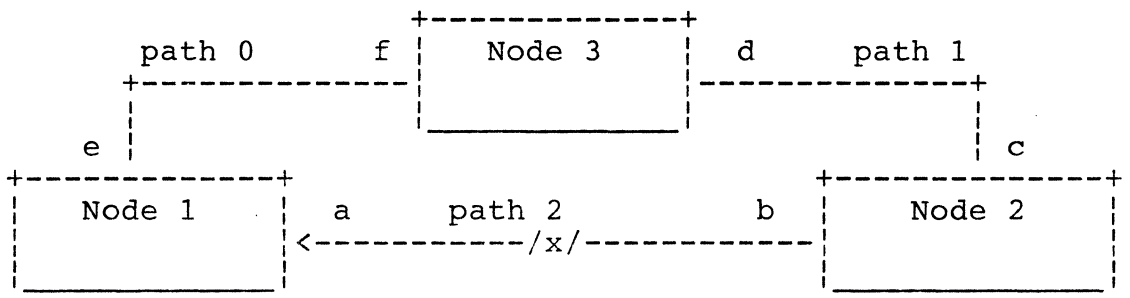
Note :

- The assignment of threads to the enabled path is done by the source node.
- The enabling of the path from Node 3 to Node 1 is done in the same way with the enabling starting at the source node.

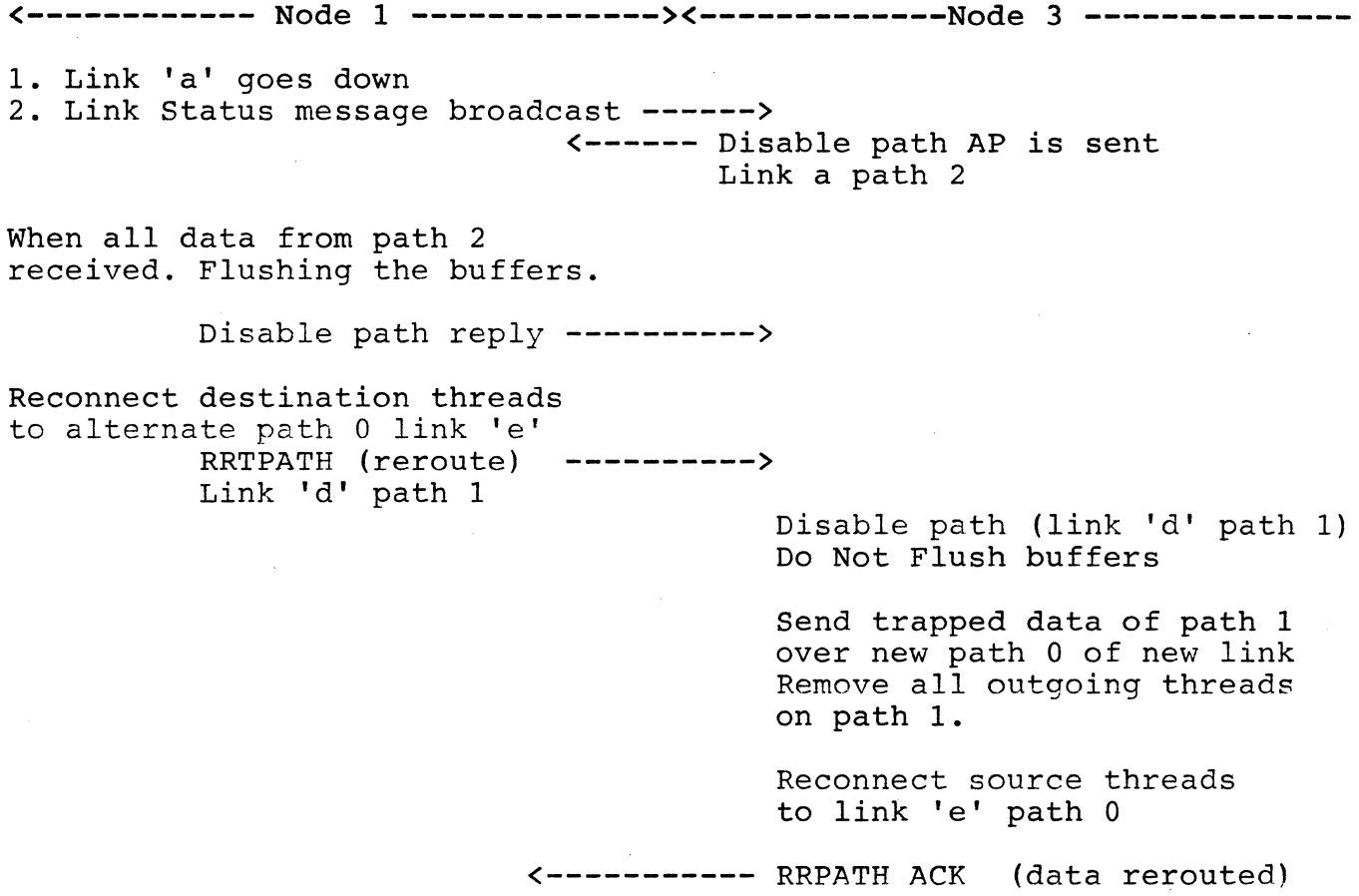
4.3.4 PATH REROUTING

Path rerouting is performed by the routing manager in a node. The reroute is accomplished via address packets, there are two packets used: a Disable Path Packet (DSBLPATH) and a Reroute Thread Path Packet (RRTPATH). The following example illustrates a 6740 path reroute.

Node 1 is connected to Link 'b' of Node 2 and Link 'c' of Node 2 is connected to Link 'd' of Node 3. A path from Node 3 to Node 1 currently exists on these links. The data flow is from Node 3 to Node 1 using Node 2 as an intermediate node. There is also Link 'e' on Node 1 connected to Link 'f' on Node 3. An alternate path 0 from Node 3 to Node 1 goes through this link. The network topology is shown below:



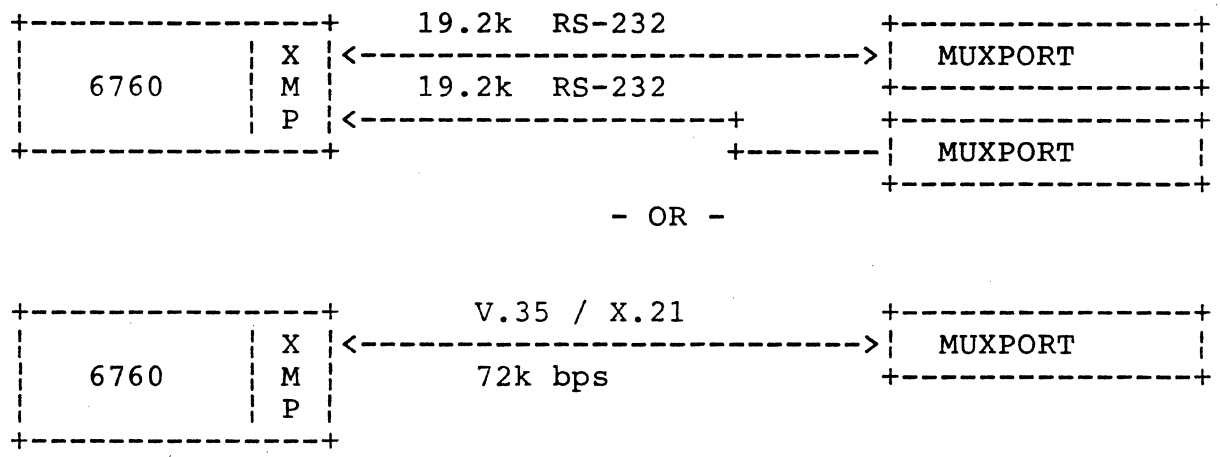
When the physical link joining Link 'a' of Node 1 and Link 'b' of Node 2 goes down the following reroute takes place on the path that goes from Node 3 to Node 1.



6760 EXTENDED MUXPORT

5.0 OVERVIEW

The 6760 Extended Muxport (XMP) consists of a controller (X1 Engine) and a Communications Interface Daughter card (CIC). The XMP will provide communications between the 6760 and 6000 6700 Muxport devices. The XMP supports one group band connection (V.35 or X.21) or two RS-232 devices at speeds up to 19.2k bps. The following figure illustrates the typical system configurations.



5.1 PERFORMANCE

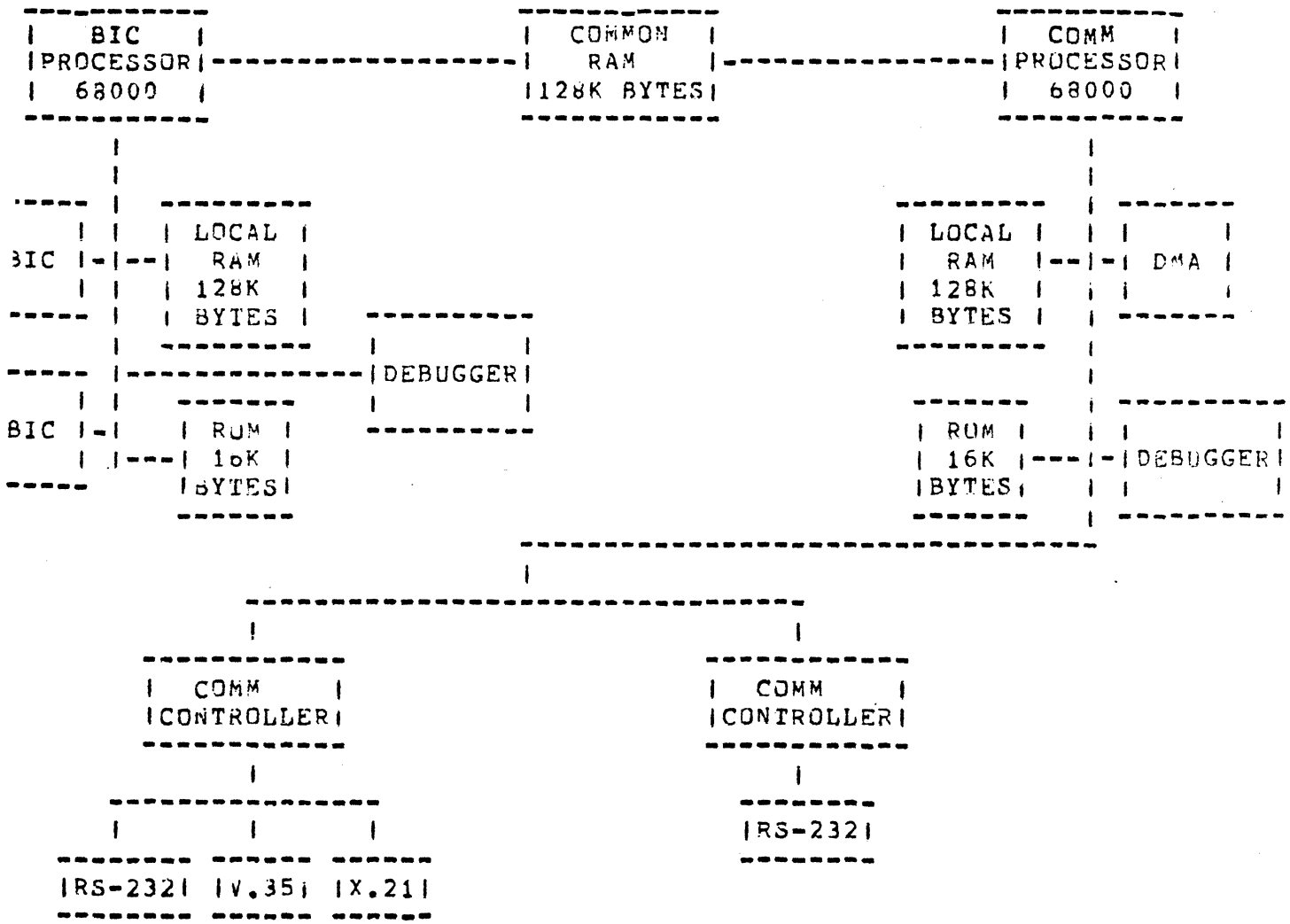
The XMP will support up to 128 threads. Depending on user population (port types) the actual number supported will be less than 128, i.e 90 - 120 for single port configuration and 50 - 70 threads for dual port configurations.

5.2 COMPATIBILITY

The XMP is a direct replacement for the 6050 Muxport. Standard Muxport protocol is supported (section 3). The Jupiter protocol, (section 4) including routing and congestion control, is also supported.

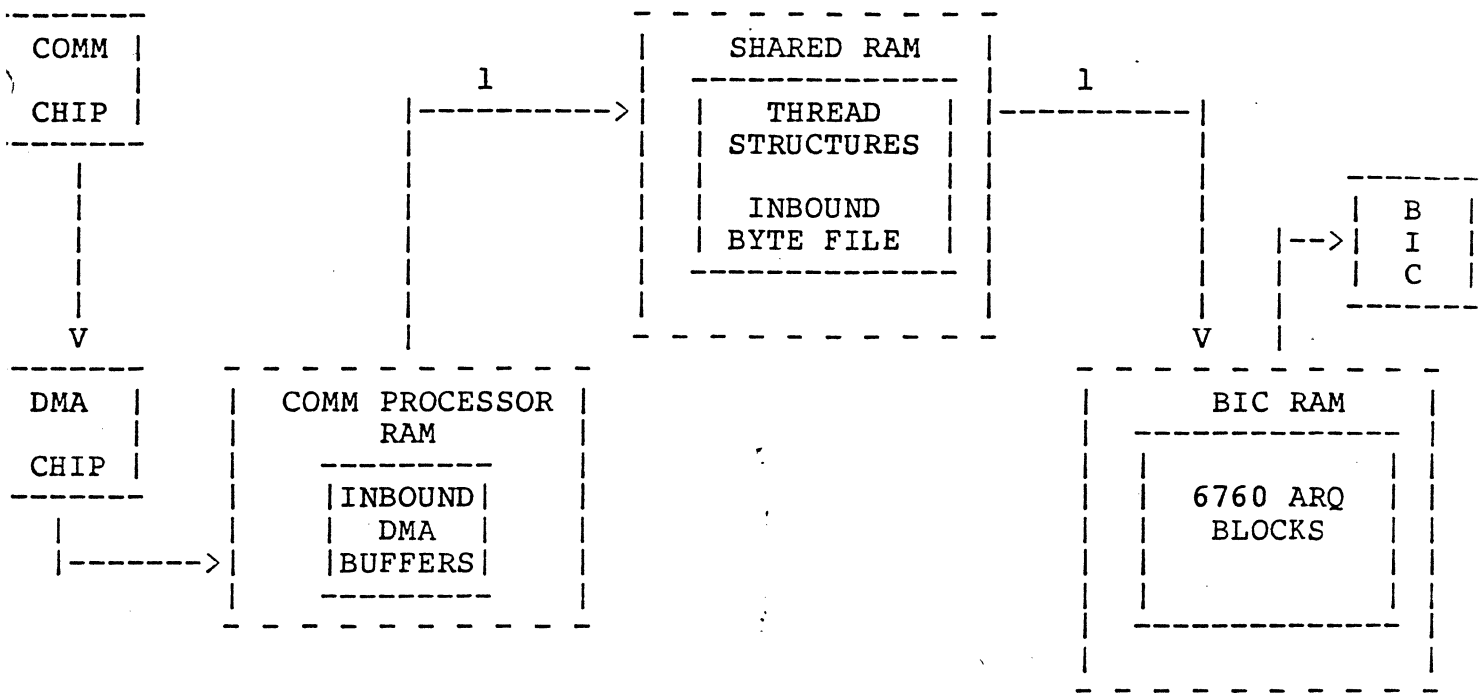
5.3 BLOCK DIAGRAM

The following figure illustrates the functional block diagram for the XMP.

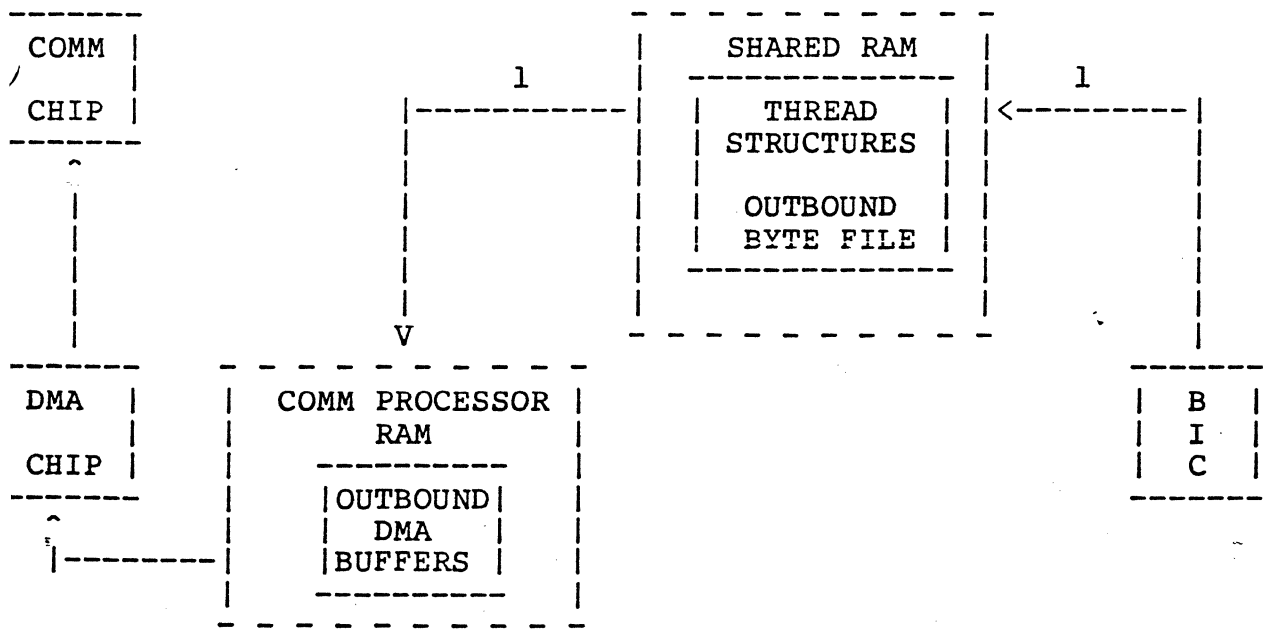


5.4 DATA FLOW

The following figures illustrate the data flow through the XMP hardware.



A.) INBOUND



B.) OUTBOUND

X.25 OVERVIEW

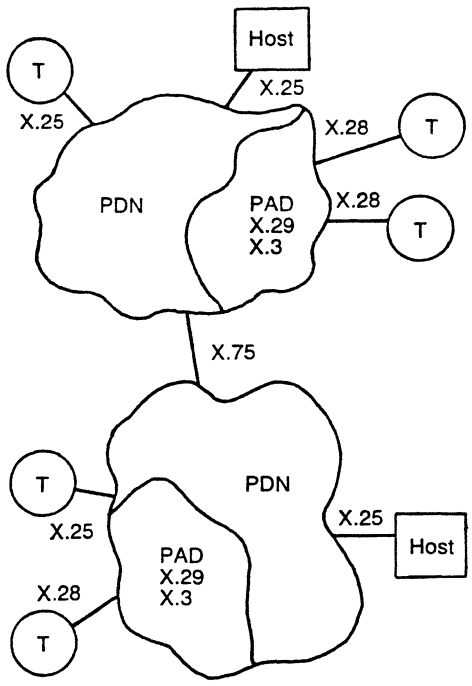
6.0 INTRODUCTION

X.25 is defined as the interface between a DTE (terminal/host) and a DCE (network node) operating in the packet mode on a Public Data Network (PDN). The protocol defines the procedures for accessing the network but not how the network functions internally. This section is meant as an introduction and overview only.

6.1 PACKET SWITCHING CONCEPTS

Packet switching is the transmission of data by means of addressed packets. A packet is a group of binary digits (1s and 0s) which are switched as a complete unit. In a packet switching network all user data is formed into discrete variable length entities called packets. Each packet contains a header specifying address and control information. The header enables the network to route a call from source to destination over shared transmission facilities.

As packet switching became popular the need developed for a common protocol that would enable users to interface their private networks with Public Data Networks or PDN's. In 1976 the CCITT formally adopted Recommendation X.25 as the packet switching protocol. A PDN is illustrated below.



6.1.1 VIRTUAL CIRCUITS

X.25 protocol uses the concept of virtual calls and virtual circuits much like the 6050/6760 uses them. When a request to enter packets into the network is received (call request) a virtual call is established between the source and destination. A virtual call exists only for the duration of the session. During the calling period the network acts as if a fixed path exists between the two calling end points. In reality each packet may travel a different route between the source and destination.

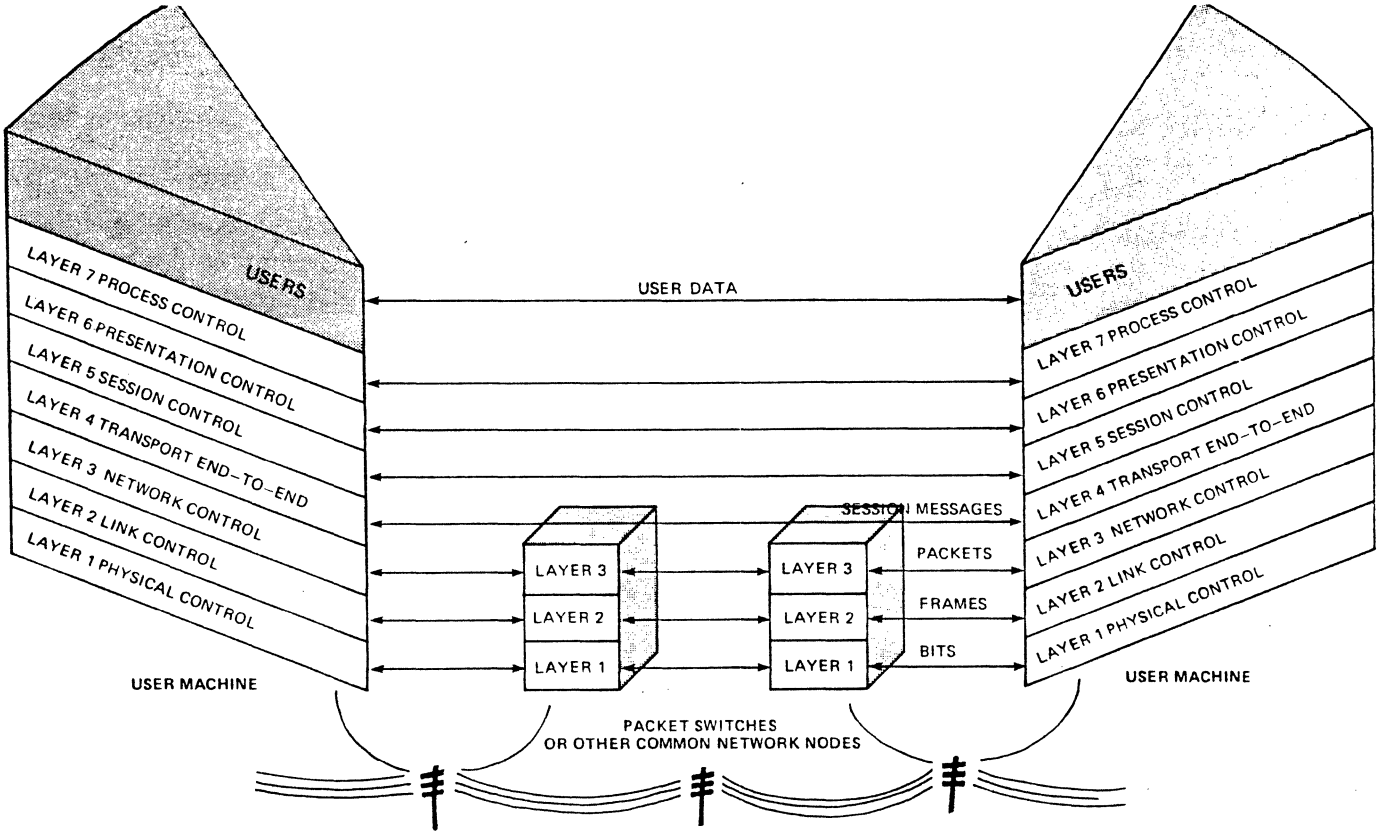
6.1.2 LOGICAL CHANNELS

The virtual circuit works by using logical channel numbers (LCN's). Each packet entering the network is assigned a logical channel number, during the call set-up phase, that indicates the session to which it belongs. The network can then associate LCN's with the DTE's at the source and destination to establish the virtual call. LCN's are assigned by both the DTE and DCE in a packet network. The DTE assigns LCN's starting with a PDN defined upper limit and working down. The DCE assigns LCN's working from the bottom up.

6.2 X.25 STRUCTURE

X.25 is structured to span the three lower levels (layers) of the ISO Reference Model, see figure below. The three levels are: Physical, Frame and Packet. The Physical level deals with how 1s and 0s are represented, how contact is established and timing considerations.

The Frame level (ISO Data Link) provides for reliable communications between a DTE (terminal) and a DCE (PDN). The protocol used at this layer are LAP or LAPB (subsets of HDLC). The third or Packet layer (ISO Network) is concerned with the format and meaning of the data field (packet) within each frame. The Packet layer provides for routing and virtual circuit management.



6.2.1 PHYSICAL LEVEL (LAYER 1)

This layer defines the requirements for functional, mechanical and electrical interface to the modem, communications facility, or DCE. This layer is responsible for establishing and maintaining the physical link between user machines. X.25 level 1 specifies the requirements of Recommendation X.21 (X.21 is RS-232C compatible).

6.2.2 FRAME LEVEL (LAYER 2)

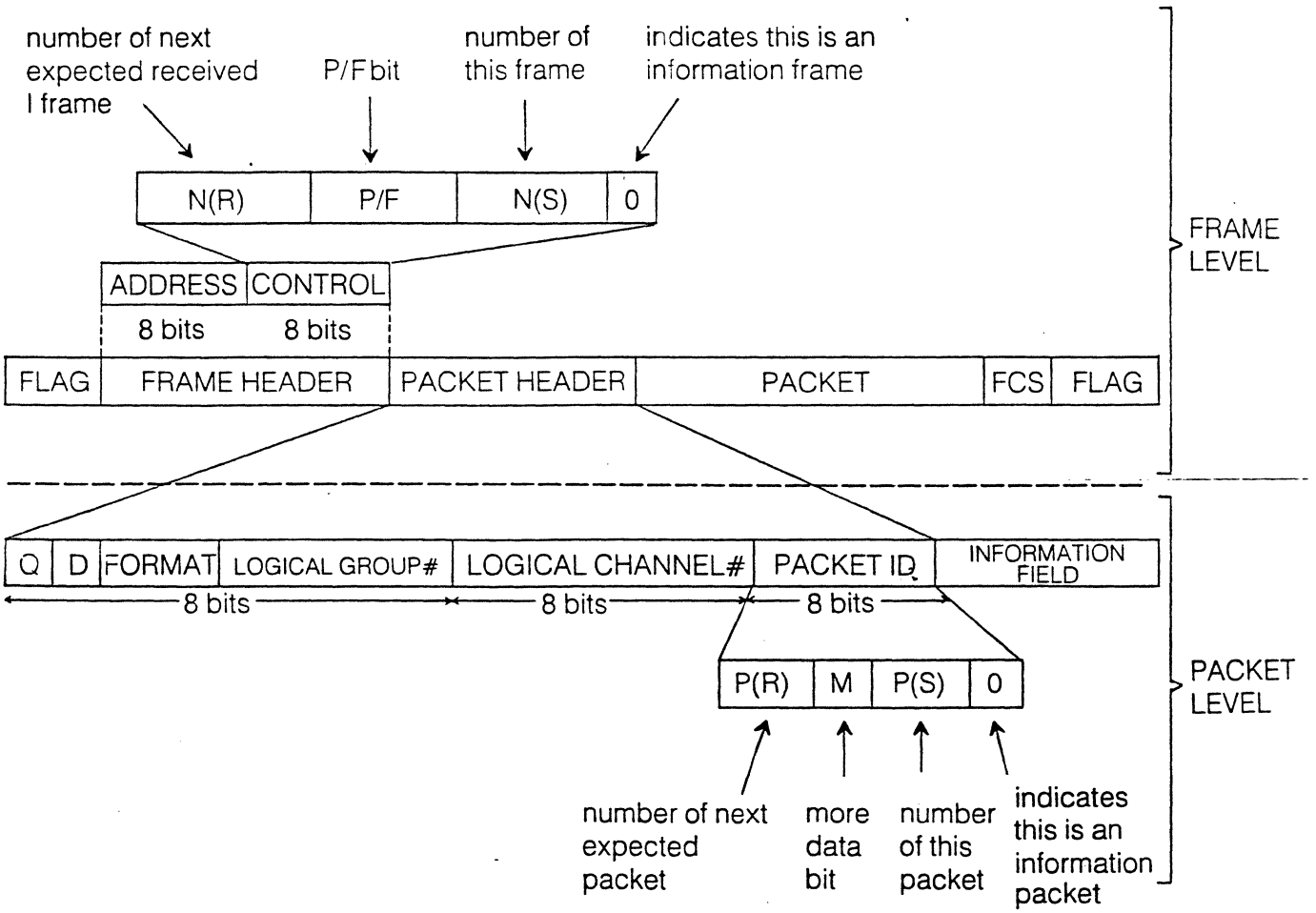
Layer 2 protocols guarantee an error free transmission across the communications link, ensuring no lost or damaged frames. The purpose of the frame layer is to provide a controlled pipeline for transmitting packets (level 3 data) from DTE to DCE. X.25 currently defines two protocols at this level: LAP and LAPB (Link Access Procedure Balanced). LAPB is the preferred protocol and is full compatible with HDLC procedures.

Level 2 protocols exchange data in frames, which contain delimiter (flags), address, control, error checking (FCS), and information fields. The information field contains the X.25 packet. For a complete discussion of HDLC protocol see section 2.

6.2.3 PACKET LEVEL (LAYER 3)

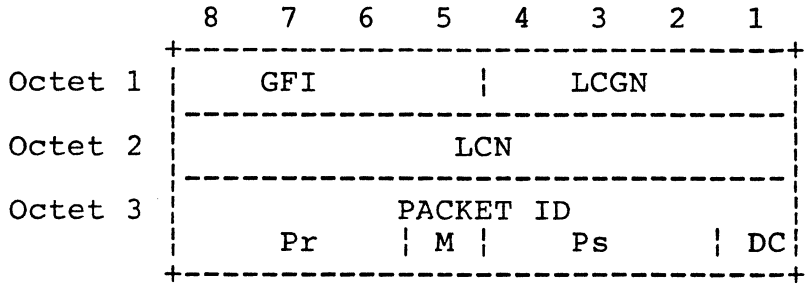
This layer defines the procedures for establishing and clearing virtual calls between the DTE (subscriber) and the DCE (network). The Packet level describes the formats and procedures for data transfer, flow control and error recovery in a packet switched network.

The X.25 packet layer is sent in the information field of an HDLC frame each level 2 I-Frame will contain only one level 3 packet. While the frame level interface procedures (level 2) apply to all data flow on the link between the DTE and DCE the packet level procedures apply to each end of a virtual call. The X.25 Layer 2 and 3 relationship is illustrated below:



6.3 X.25 PACKET FORMAT

Each packet contains a common three octet header (an octet is eight bits). The first four bits form the general format identifier field (GFI), which indicates the sequence numbering scheme. The next twelve bits identify the logical group and channel numbers. The last octet identifies the packet type. See figure below.



Where:

GFI = General Format Identifier
LCGN = Logical Channel Group Number
LCN = Logical Channel Number
Packet ID = Type of Packet
Pr = Receive Sequence Number (next expected receive packet)
Ps = Packet send sequence number
M = More Data Bit (if set data is continued in next packet)
DC = Data or Control Bit. If 0 data follows, if set (1) control information follows

6.3.1 GENERAL FORMAT IDENTIFIER (GFI)

The first four bits (high order) of the first header octet contain the General Format Identifier (GFI). The GFI determines the format of the remainder of the packet header. It also informs the DTE/DCE if the packet contains control information and/or requires end-to-end acknowledgement. The GFI can have one of the following formats:

8	7	6	5	Packet Type
Q	D			
0	0	0	1	Clearing, Flow Control, Reset, Restart, Interrupt Packets.
0	X	0	1	Call Setup Packet
X	X	0	1	Data Packet

Where:

Bit 8 = Qualifier Bit (Q-Bit). If set (1) this bit signals that the data packet contains control information.

Bit 7 = Delivery Confirmation Bit (D-Bit). If this bit is set the acknowledgement is from end-to-end. If the bit is zero (not set) the local DCE (network node) will acknowledge the packet.

Bits 6 & 5 = This bit pair indicates the modulo sequencing to be used. The valid settings are:

01	Packet sequencing is modulo 8
10	Packet sequencing modulo 128
11	Applies to datagrams

6.3.2 LOGICAL CHANNEL GROUP NUMBER (LCGN)

The Logical Channel Group Number (LCGN). Bits 1, 2, 3, and 4 of the first octet indicates groups of logical channel numbers designated for particular type of access (e.g. accept calls only). The LCGN extends the addressing range of the LCN to twelve bits.

6.3.3 LOGICAL CHANNEL NUMBER (LCN)

The second octet of the header contains the Logical Channel Number or LCN. Each packet entering the network is assigned a LCN at call establishment the LCN is similar to a Dynamic Transfer Port in the 6050. The LCN can range from 0 to 255, but 0 is reserved. The LCN is assigned by the DTE during call request starting with an agreed upon upper limit and working down. The DCE assigns LCN's working from the bottom of the range upwards. Logical Channels are assigned by category each call falls into one of four categories, they are:

1. Permanent Virtual Circuits (leased line) no call set up phase exists. the LCN is permanently in data transfer state.

2. One-Way Incoming Calls (contention) only the DCE can initiate call set up via incoming call packets.

3. Two-Way Switched Virtual Calls (dial up) either the DTE or DCE may initiate call establishment procedures.

4. One-Way Outgoing Calls Only the DTE may initiate call setup attempts.

6.3.4 PACKET TYPE IDENTIFIER

The third octet of the header identifies the type of packet. This byte is similar to the control field in an HDLC frame. The Packet ID contains send and receive counts (Ps & Pr) along with the More data Bit (M Bit). The M bit indicates that the data in the packet is not complete and is continued in another packet. The table below defines the packet types:

Data	P(R)			M	P(S)			0
Call request	0	0	0	0	1	0	1	1
Call accepted	0	0	0	0	1	1	1	1
Clear request	0	0	0	1	0	0	1	1
Clear confirmation	0	0	0	1	0	1	1	1
Interrupt	0	0	1	0	0	0	1	1
Interrupt confirmation	0	0	1	0	0	1	1	1
Receive ready	P(R)			0	0	0	0	1
Receive not ready	P(R)			0	0	1	0	1
Reject	P(R)			0	1	0	0	1
Reset request	0	0	0	1	1	0	1	1
Reset confirmation	0	0	0	1	1	1	1	1
Restart request	1	1	1	1	1	0	1	1
Restart confirmation	1	1	1	1	1	1	1	1
Diagnostic	1	1	1	1	0	0	0	1

6.4 CALL ESTABLISHMENT

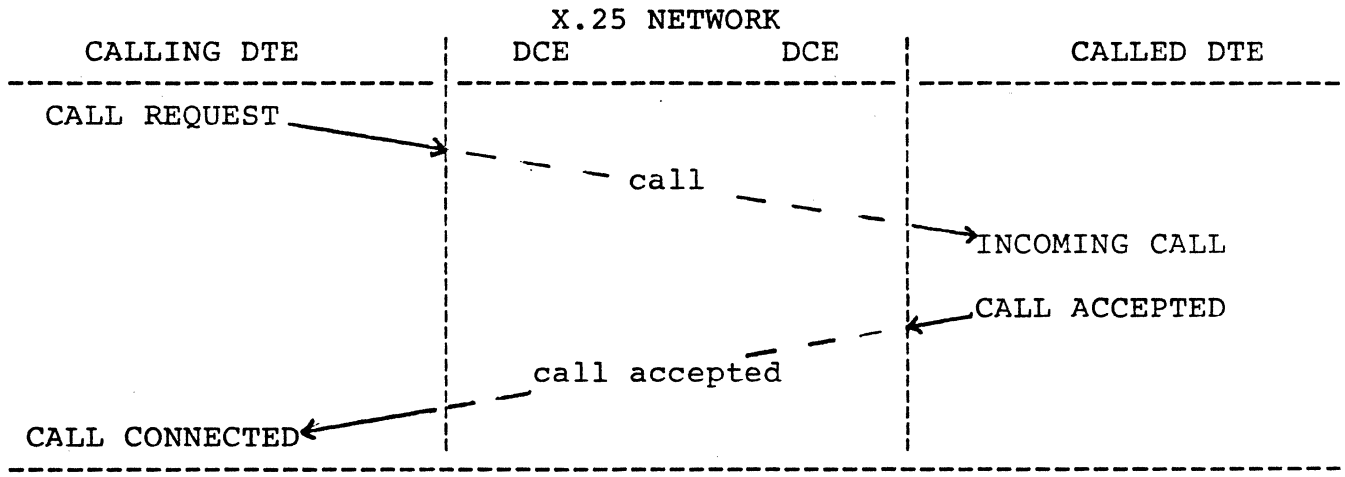
Packet level end to end communications is accomplished in the following manner: (reference Figure below).

1. The calling DTE sends a CALL REQUEST packet to the network, after selecting an LCN. This packet specifies the address of the called DCE.

2. The network routes the packet to the destination, where the remote DCE selects an LCN and transmits a CALL PACKET to the called DTE.

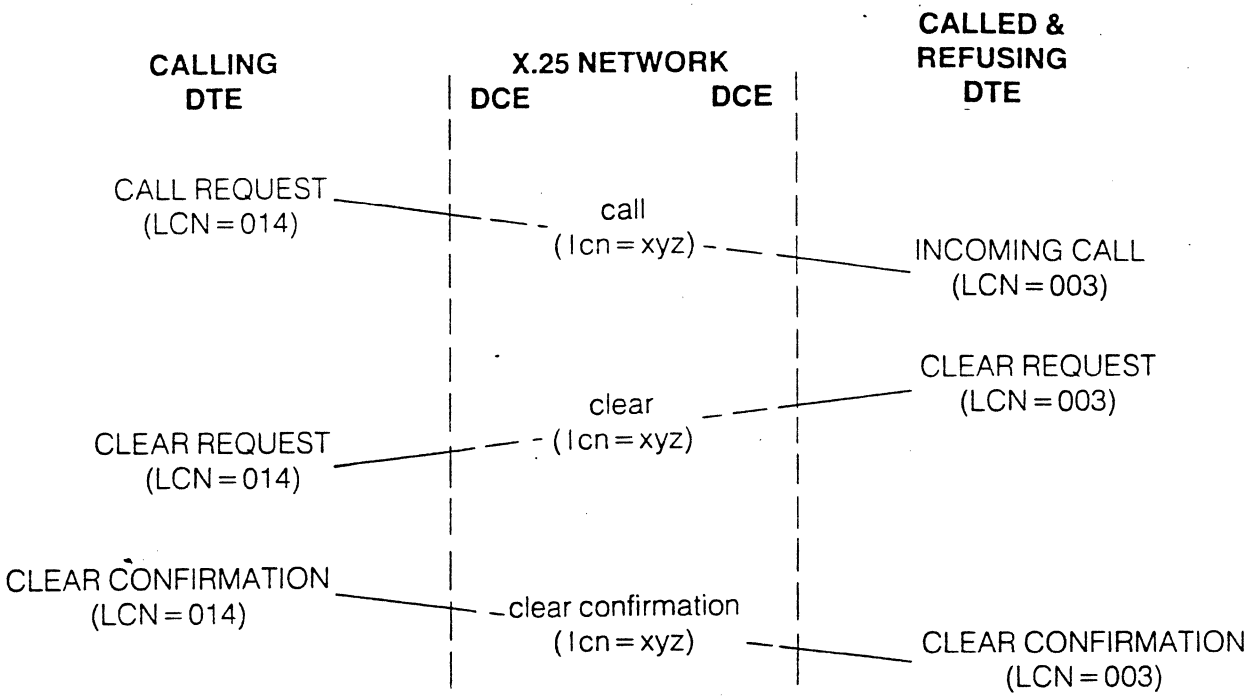
3. If the remote DTE chooses to accept the call it transmits a CALL ACCEPT packet to the remote DCE.

4. The network will send a CALL CONNECTED packet to the calling DTE. The virtual call is now established and data packets may be exchanged.



6.5 CALL CLEARING

The called DTE may refuse the call. In this case it will generate a CLEAR REQUEST packet. The calling DTE will then receive a CLEAR INDICATION packet and will respond with a CLEAR CONFIRMATION packet thus freeing the LCN for another user. The following figure depicts the remote DTE refusing the call and clearing the line:

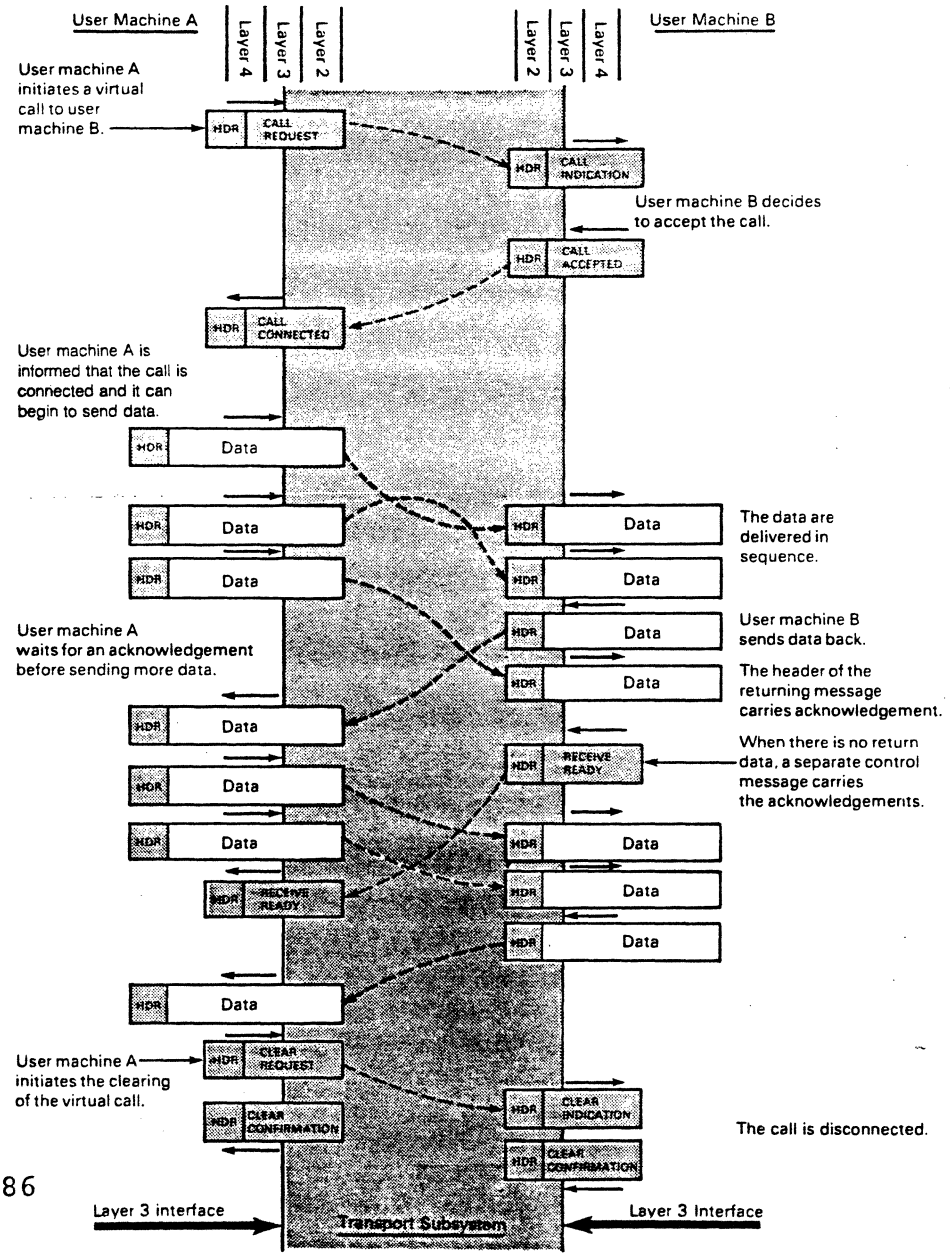


Call Clearing Procedure

6.6 DATA TRANSFER

Once a virtual circuit has been established, data packets may be exchanged. All data packets are assigned a send and receive count number (Ps & Pr). These numbers are equivalent in function to the Ns and Nr numbers in an HDLC frame.

Other bits used in data transfer are: the Q-bit, D-bit and M-bit. The Q-bit indicates that control information follows. The Q-bit is analogous to a control slot in Muxport protocol. The D-bit determines whether packets are to be acknowledged by the remote (D = 1) or acknowledged by the local DCE (D = 0). The M-bit, if set indicates that there is a logical continuation of data in the next data packet. The figure below illustrates call establishment, data transfer and call clearing using X.25 protocol packets.

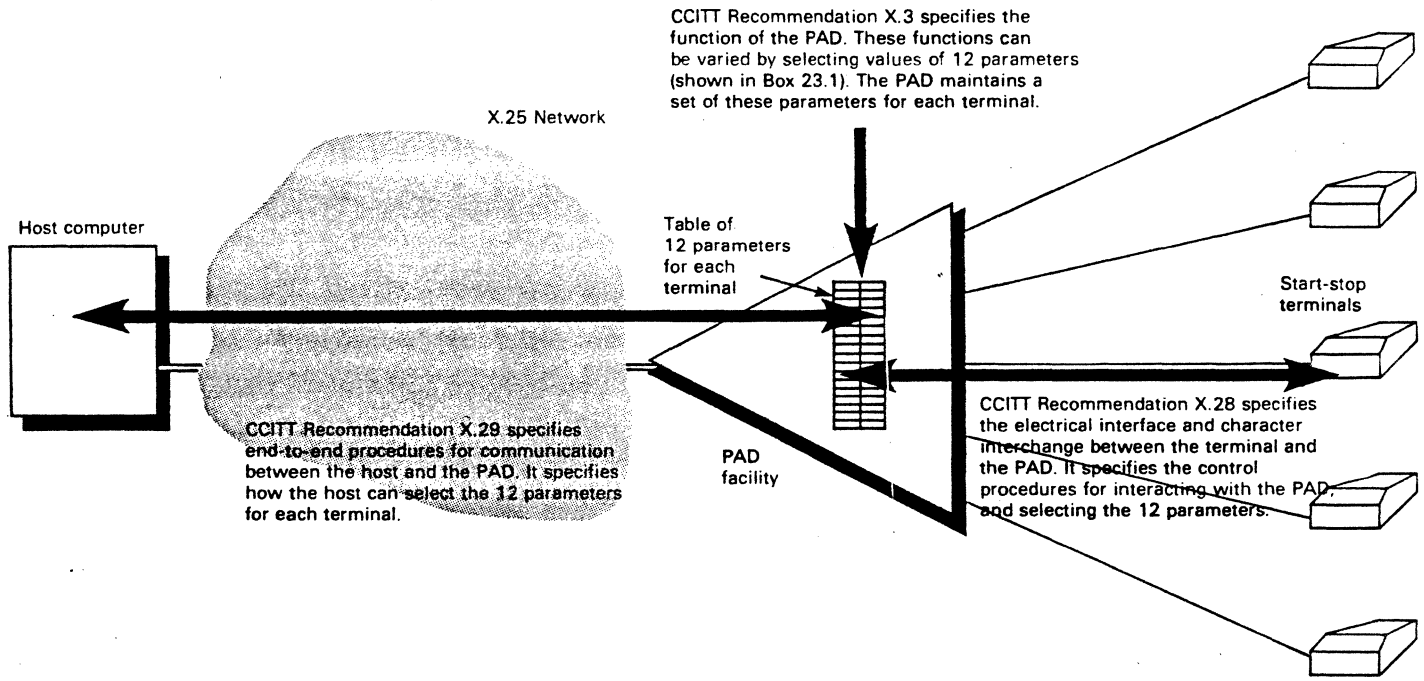


6.7 PACKET ASSEMBLER/DISASSEMBLER (PAD)

A PAD is a device (hardware or software) that accepts strings of characters from non-X.25 terminals and assembles them into packets for transmission through the network. Packets that arrive at a PAD are disassembled and delivered to the user terminal one character at a time. The following CCITT standards govern the use of the PAD:

X.3 - Describes the PAD functions
X.28 - Defines the interface between the terminal and the PAD
X.29 - Specifies the procedures governing the exchange of data between the PAD and a packet mode (X.25) terminal.

The figure below illustrates the function of the PAD:



TERMS AND DEFINITIONS

Abort	In HDLC protocols, the act of prematurely terminating a frame by the transmission of seven contiguous "1" bits with no zero insertion.
ADCCP	Advanced Data Communications Control Procedure The ANSI bit oriented protocol, similar to HDLC.
Address Field	The eight bit sequence immediately following the opening flag used to identify stations on the link.
ANSI	American National Standards Institute.
BOP	Bit Oriented Protocol.
Control Field	The eight bit (16 if extended) following the address field that identifies the frame type.
Data Link Control Procedure	(DLC) - A set of rules and procedures governing the exchange of data between machines over a communications link.
FCS	Frame Check Sequence - a 16 bit mathematical computation, generated at the transmitting station and checked at the receiving station. If a difference occurs the frame is in error.
Flag	A unique eight bit sequence generated at the transmitter, indicating the beginning and end of a Frame. Used to provide synchronization.
Frame	A group of bits (32 min.) sent serially over a communications link. A logical transmission unit sent between data link layer entities.
FRMR	Frame Reject. A link level response generated upon receipt of an undefined or to large frame.
HDLC	High Level Data Link Control - The ISO bit oriented protocol standard.
Idle	In HDLC protocols, the transmission of at least 15 consecutive "1" bits or the transmission of continuous flag characters.

Information Field	The sequence of bits occurring between the last control field bit and first bit of the FCS.
Invalid Frame	A sequence of bits, following an opening flag that is terminated in an abort sequence or contains less than 32 bits before the end flag.
ISO	International Standards Organization. A global standards setting body.
ITI	Interactive Terminal Interface. A PAD for supporting asynchronous terminal access to an X.25 network.
Logical Channel	The part of an X.25 interface that supports a logical connection for a call. X.25 allows up to 4095 logical channels.
Modulo 8	A cyclical numbering system used in HDLC/X.25 protocols in which sequenced frames or packets assume the same value as the integral remainder of division by 8. A frame can have a sequence value of 0 thru 7.
Nr	Receive Sequence Number. A link level variable that is transmitted with I-frames and S-frames indicating the sequence number of the next expected I-frame and acknowledging all frames up to Nr-1.
NRZ	Non Return to Zero - a binary encoding method where "1"'s and "0"'s are represented by opposite and alternating high and low voltages. There is no return to zero between bits.
NRZI	Non Return to Zero Inverted - an encoding method that inverts the signal on a "0" and leaves the signal unchanged on a "1". A change in the voltage represents a "0" and the absence of a voltage change denotes a "1".
Ns	Send Sequence Number. A link Level variable sent in I-frames only indicating the 'address' in sequence, using Modulo 8, of the frame.

OSI	Open System Interconnect. A seven layer model used to development an international standard for data exchange.
Octet	An eight bit group of information, a byte.
PAD	Packet Assembler/Dissembler. A device that converts a native mode data stream into X.25 packets. Typically for async devices.
PDN	Public Data Network. A packet switched network
Pr	Packet Receive Sequence. In X.25 packets the next packet expected to be received.
Protocol	A formal set of rules governing the format of data and the relative timing of message exchange between two devices.
Ps	Packet Send Sequence. In X.25 packet level the number of this packet.
SDLC	Synchronous Data Link Control - IBM's BOP
STE	Station Terminal Equipment. An X.75 node.
Two-Way Alternate	Half Duplex communications.
Two-Way Simultaneous	Full duplex communications.
Vr	Receive State Variable. A link level variable whose value indicates the sequence number of the next I-Frame to be received.
Vs	Send State Variable. A link level variable whose value indicates the sequence number of the next I-frame to be transmitted.
Window	The maximum number of frames or packets that may be outstanding without acknowledgement from the remote unit.
Zero Insertion	The technique used by BOP protocols to achieve data transparency. The transmitter inserts a 0 bit after every five contiguous "1" bits. The receiver strips the zero bits to recover the original data.