Exhibit C

X.25 EXPLAINED:

Protocols for Packet Switching Networks

Second Edition

R. J. DEASINGTON
IBM (UK) Laboratories Ltd.

ELLIS HORWOOD LIMITED
Publishers · Chichester
Halsted Press: a division of
JOHN WILEY & SONS
Chichester · New York · Ontario · Brisbane

Second Edition published in 1986 by

ELLIS HORWOOD LIMITED

Market Cross House, Cooper Street,
Chichester, West Sussex, PO19 1EB, England

First Edition published by **Ellis Horwood Limited** in 1985

The publisher's colophon is reproduced from James Gillison's drawing of the ancient Market Cross, Chichester.

Distributors:

Australia and New Zealand:
JACARANDA WILEY LIMITED
GPO Box 859, Brisbane, Queensland 4001, Australia

Canada:
JOHN WILEY & SONS CANADA LIMITED
22 Worcester Road, Rexdale, Ontario, Canada

Europe and Africa:
JOHN WILEY & SONS LIMITED
Baffins Lane, Chichester, West Sussex, England

North and South America and the rest of the world:
Halsted Press: a division of
JOHN WILEY & SONS
605 Third Avenue, New York, NY 10158, USA

British Library Cataloguing in Publication Data
Deasington, R. J.
X.25 explained: protocols for packet switching networks. — 2nd ed. —
(Ellis Horwood series in computers and their applications)
1. Computer networks 2. Data transmissions systems
3. Packet switching (Data transmissions) I. Title
004.6'6 TK 5105.5

ISBN 0–7458–0110–2 (Ellis Horwood Limited)
ISBN 0–470–20731–0 (Halsted Press)

Typeset by Ellis Horwood Limited
Printed in Great Britain by R.J. Acford, Chichester

Note that in class D facilities the second octet indicates the number of octets which will follow rather than the total number of octets in the facility. In the case of the Segment Count facility the number of octets following the two header octets is eight. In this case the number of octets transferred in each direction is given as a 32-bit integer with the most significant octet first in each case. The coding of the Segment Count is now given:

Bits:	8	7	6	5	4	3	2	1	
Oct 1	1	1	0	0	0	0	1	0	Class D and Facility code
2	0	0	0	0	1	0	0	0	Length count field of 8
3	Most significant octet of								
4	Received Segment Count								
5									Received segments
6	Least significant octet								
7	Most significant octet of								
8	Sent Segment Count								
9									Sent segments
10	Least significant octet								

Call Duration and Segment Count facilities are often used by gateways which allow access to users of expensive networks such as IPSS (International Packet Switched Stream) by British Telecom, where access to a host in another continent can cost many pounds sterling per hour of connection or per thousand segments sent, a segment being 64 octets or a part of 64 octets if a partly empty packet is sent. Comparisons are performed between the locally calculated cost and that billed by the network provider. In the case of PSS and IPSS this comparison is particularly easy to make as one of the nationally defined extensions allows the cost of a call to be returned as part of a Fast Select Clear Indication packet.

4.6 DATA TRANSFER PACKETS

The data transfer phase of either a PVC or an SVC makes use of four packet types, Data, Receiver Ready (RR), Receiver Not Ready (RNR), and Reject (REJ). These packets, used in conjunction with each other, allow the receiver of a stream of data to control the speed at which it is sent to it by the network; this is performed by the use of 'flow control' which will be described after the basic formats of the packets have been given. As the name implies the Data packet is the packet type used to send user data from one DTE across the network to another DTE. The format of the Data packet is now given:

Bits:	8	7	6	5	4	3	2	1	
Oct 1	Q	0	0	1	LGN				Qualifier, GFI and LGN
2	LCN								LCN
3	P(R)			M	P(S)			0	Sequence counts and More bit
4	User data								0–128 octets of user data

The Data packet above is shown using the same notation as before. Up to 128 octets of user data may follow the three-octet header, or up to 2048 if the packet size has been successfully negotiated to this size. The X.25 protocol is slightly inconsistent in its use of the packet header for data packets in that the General Format Identifier is used to carry the 'Qualifier' bit which is a user-settable bit which has significance from end to end only. This bit is typically used by a higher level protocol to differentiate between data which is intended for a user process and data which forms part of the high level protocol itself. The other user-defined bit is the 'More' bit, this is used to indicate that a packet is part of a series of packets which together make up a single logical message. The More bit may only be set on if the packet is full, i.e. contains the same number of octets of user data as was negotiated at call set-up time; if a partially filled packet is sent with the More bit set on the network will clear the call since a procedure error will have been detected. If the More bit is not set on, the packet may contain any number of data octets, some networks, however, prohibit the transmission of packets with zero octets of user data. The two sequence counts in octet three are used to provide the flow control mechanism as we shall now describe. It should be remembered that flow control at the packet level only has significance for the SVC or PVC using the Logical Group and Channel number given in the header octets, stopping the flow of data on one virtual circuit does not affect the flow of data on any other. This property should be considered in comparison with the flow control exerted at the Link level where the blocking of further transmission will halt the flow of data for all calls in progress. It is obviously mandatory to permit users of X.25 Network level interfaces only to control data at the Network level, so as to avoid interference with other users. The flow control mechanism used is very similar to that employed by LAPB, except that here the fact that the Network level makes use of a virtual error-free channel provided by the Link level can be used to render unnecessary any error detection or correction procedures. The other packet types involved with the transmission of data are now detailed. The Receiver Ready (RR) is shown first:

Bits:	8	7	6	5	4	3	2	1	
Oct 1	0	0	0	1	LGN				GFI and LGN
2	LCN								LCN
3	P(R)			0	0	0	0	1	Sequence count and PTI

The Receiver Not Ready (RNR) packet:

Bits:	8	7	6	5	4	3	2	1	
Oct 1	0	0	0	1	LGN				GFI and LGN
2	LCN								LCN
3	P(R)			0	0	1	0	1	Sequence count and PTI

The Reject (REJ) packet:

Bits:	8	7	6	5	4	3	2	1	
Oct 1	0	0	0	1	LGN				GFI and LGN
2	LCN								LCN
3	P(R)			0	1	0	0	1	Sequence count and PTI

As will be recalled from the previous section on the negotiation of facilities, the range of window sizes used at the Network level of X.25 is the same as at the Link level, i.e. up to seven. It can be seen seen from the formats of the Data, RR, RNR, and REJ packets that they all contain three-bit counter fields, termed P(R) and P(S), which can count between zero and seven. The P(R) field is termed the Received sequence count and the P(S) the Sent sequence count. Each time a new packet is sent to the Link level for transport to the network node to which the DTE is connected, the Sent sequence count is incremented, except that if its current value is seven it will be set back to zero. If the difference between the last Received sequence count and the Sent sequence count is greater than the value agreed for this virtual circuit, then the data is not sent and the flow in that direction is blocked until a packet is received which updates the Received sequence count. The Received sequence count can be sent on any of the four data transfer type packets, i.e. Data, Receiver Ready, Receiver Not Ready or Reject. If data is flowing in both directions at a rate which both ends of the link can deal with satisfatorily then no other packet types need to be sent, if there is no data to be sent outwards from a DTE which is receiving Data packets from some remote host then either the Receiver Ready or Receiver Not Ready packet type may be used to carry the latest Received sequence count. The RR type packet is used to indicate that the DTE has sufficient buffer space and resources to continue to receive packets of data, the RNR type packet would be used to inform the remote DTE that, owing to resource constraints, no further packets of data should be forwarded on this virtual channel until a Receiver Ready packet is sent to indicate that the congestion situation is over. It may be that a DTE receiving data will run out of buffer space for further packets and send a Receiver Not Ready packet at the same time as further Data packets are being sent to it, the link protocol being truly full duplex in nature. If this situation occurs and the receiving DTE is forced to throw away a packet of data any subsequent packets would have a sequence count which is not in sequence. To indicate that packets need to be retransmitted the Reject packet type is used, in the same way as at Level 2, to indicate the next sequence count it expects to receive from the network; thus the remote DTE must restransmit this packet and any it has been subsequently. The sequence counters both start at zero when the call is set-up, and are reset to zero should the call be Reset, a procedure which will be decribed shortly. The response to a data packet transmitted to the network, i.e. the reception of a P(R) greater than or equal to the P(S) of the Data packet, must be received within a certain time, termed the Level 3 time-out period; failure to receive the necessary confirmation of receipt will cause the call to be reset. Hence the RNR packet should be used to hold off data rather than indefinitely holding back the acknowledgement.

We will now show a session during which a call is received by a DTE, a Call Accept is sent followed by the transfer of data and clear-down. This example will first illustrate only the Network level frame transfer, but will then be repeated to show the operation of both the Link and Network levels for this transaction. The transaction shown is that of an interactive terminal calling a host computer using a Packet Assembler/Disassembler (PAD) using the X.29 protocol which will be described in Chapter 6. This protocol makes use of the Qualified data bit to signify messages destined for the control of the PAD itself rather than for the terminal. After the connection has been established the host system sends a message to the PAD to set it into a mode suitable for working with this machine's operating system, the PAD replies showing the parameters set up as the host requested; the host then resets the PADs mode this time not asking for a reply. The remaining data is not Qualified and is destined for the terminal or host, depending on the direction, without alteration. All the frames have the same logical group and channel number so this is not shown here.

			Received		Transmitted	
Direction	Q bit	Packet type	P(R)	P(S)	P(R)	P(S)
Rx→		Call Req.				
←Tx		Call Acc.				
←Tx	Q	Data			0	0
Rx→	Q	Data	1	0		
←tx		RR			1	
←Tx	Q	Data			1	1
←Tx		Data			1	2
Rx→		RR	2			
Rx→		RR	3			
Rx→		Data	3	1		
←Tx		RR			2	
←Tx		Data			2	3
←Tx		Data			2	4
Rx→		RR	4			
Rx→		RR	5			
Rx→		Clr Req.				
←Tx		Clr Conf.				

It can be seen from this example that the holdback timer on the DTE, which controls how long the DTE should wait to see if a Data packet from the system itself comes along to be sent out rather than sending out an RR frame when an acknowledgement is required, is set to a short interval so that the Data packets rarely carry acknowledgements with them. The adjustment of the holdback timer is generally a compromise between the *Scylla* of reduced response time and the *Charybdis* of increased utilisation of the channel by packets which are strictly redundant. For a host handling mostly interactive work, the increased response of immediate acknowledgment may be advantageous, whereas for a host handling mostly bulk data transfer the increased overall throughput of the line will probably be of greater importance. the same set of Network level transactions shown above is now shown but with the operation of the Link level also given:

	Link level						Network level				
Direction	Frame type	Received N(R)	Received N(S)	Transmitted N(R)	Transmitted N(S)	Q bit	Packet type	Received P(R)	Received P(S)	Transmitted P(R)	Transmitted P(S)
Rx→	I	1	5				Call Req.				
←Tx	RR			6							
←Tx	I			6	1		Call Acc.				
Rx→	RR	2									
←Tx	I			6	2	Q	Data			0	0
Rx→	I	3	6			Q	Data	1	0		
←Tx	RR			7							
←Tx	I			7	3		RR			1	
←Tx	I			7	4	Q	Data			1	1
←Tx	I			7	5		Data			1	2
Rx→	I	6	7				RR	2			
←Tx	RR			0							
Rx→	I	6	0				RR	3			
←Tx	RR			1							
←Tx	RR(P)			1							
Rx→	RR(F)	6									
Rx→	I	6	1				Data	3	1		
←Tx	RR			2							
←Tx	I			2	6		RR			2	
←Tx	I			2	7		Data			2	3
←Tx	I			2	0		Data			2	4
Rx→	I	1	2				RR	4			
←Tx	RR			3							
Rx→	I	1	3				RR	5			
Rx→	I	1	4				Clr Req.				
←Tx	I			5	1		Clr Conf.				

In the preceding example the similarity between the Network and Link level is highlighted. This is a rather simple example in which only a single virtual call is operating at the Network level, obviously it is much harder to follow when several calls are proceeding at the same time. With the aid of a reasonably intelligent line-monitor it is possible to watch the progress of the Network and Link level in real-time, which will give a better idea of the timings involved. On a two-dimensional diagram it is difficult to show the occurrence of time-outs etc. and the real flavour of the protocol is best seen as it happens in a live situation. The above example shows the difference between the Link level and Network level holdback timers, the piggy-backing of acknowledgements at the Link level is quite common in this example.

Having shown the basic data transfer mechanism at the Network level the other facilities are now described, such as Call Reset and Interrupt data.

4.7 INTERRUPT PACKETS

A frequent requirement when communicating between an interactive terminal and a host computer is the provision of a facility to allow the user to inform the computer that it should stop what it is at present doing and allow the user to regain control of the system. Depending on the computer system in use it may or may not be possible to inform the host machine of such a request in the ordinary flow of data from the terminal to the host. In the situation where the host has stopped receiving input for flow control reasons it would be impossible to receive the user's request for attention. This problem is overcome by the provision in the X.25 protocol of an Interrupt facility which operates outside the normal flow control constraints of Data packet transmissions. Since this means of sending data has no flow control in it, a restriction is imposed that a DTE cannot have more than one outstanding Interrupt on any virtual call in each direction. Thus the receiving software need only reserve one byte of buffer space for each logical channel in use in case an Interrupt is received. Higher level protocols for handling interactive terminals generally make use of the Interrupt facility to provide a standardised way of causing an attention signal to be raised in a host. The format of the Interrupt packet is now given:

Bits:	8	7	6	5	4	3	2	1	
Oct 1	0	0	0	1	LGN				GFI and LGN
2	LCN								LCN
3	0	0	1	0	0	0	1	1	PTI
4	User data								