SYSTEMS & CONTROL ENCYCLOPEDIA

Theory, Technology, Applications

Editor-in-Chief

Madan G Singh

UMIST, MANCHESTER, UK

Volume 6
P-Sim

SYSTEMS & CONTROL ENCYCLOPEDIA

Theory, Technology, Applications

VOLUME 6
P–Sim

Editor-in-Chief

Madan G Singh

University of Manchester
Institute of Science and Technology,
Manchester, UK

PERGAMON PRESS

OXFORD · NEW YORK · BEIJING · FRANKFURT
SÃO PAULO · SYDNEY · TOKYO · TORONTO

U.K.	Pergamon Press, Headington Hill Hall, Oxford, OX3 0BW, England
U.S.A.	Pergamon Press Inc., Maxwell House, Fairview Park, Elmsford, New York 10523, U.S.A.
PEOPLE'S REPUBLIC OF CHINA	Pergamon Press, Qianmen Hotel, Beijing, People's Republic of China
FEDERAL REPUBLIC OF GERMANY	Pergamon Press, Hammerweg 6, D-6242 Kronberg, Federal Republic of Germany
BRAZIL	Pergamon Editora, Rua Eça de Queiros, 346, CEP 04011, São Paulo, Brazil
AUSTRALIA	Pergamon Press Australia, P.O. Box 544, Potts Point, N.S.W. 2011, Australia
JAPAN	Pergamon Press, 8th Floor, Matsuoka Central Building, 1–7–1 Nishishinjuku, Shinjuku-ku, Tokyo 160, Japan
CANADA	Pergamon Press Canada, Suite 104, 150 Consumers Road, Willowdale, Ontario M2J 1P9, Canada

First edition 1987

Library of Congress Cataloging in Publication Data
Main entry under title:
Systems and control encyclopedia.
Includes bibliographies.
1. System analysis—Dictionaries.
2. Control theory—Dictionaries.
3. Systems engineering—Dictionaries.
I. Singh, Madan G.
QA402.S968 1987 003'.03'21 86-15085

British Library Cataloguing in Publication Data
Systems and control encyclopedia: theory, technology, applications.
1. Control theory—Dictionaries I. Singh, Madan G.
003'.03'21 QA402

ISBN 0–08–028709–3 (set)

Printed in Great Britain by A. Wheaton & Co. Ltd, Exeter

SYSTEMS & CONTROL ENCYCLOPEDIA

Theory, Technology, Applications

EDITORIAL ADVISORY BOARD

CHAIRMAN

HONORARY EDITORIAL ADVISORY BOARD

SUBJECT EDITORS

Applications of Control to Aerospace and Aeronautics

M Pélegrin
Centre d'Etudes et de Recherches de Toulouse
Toulouse, France

Applied Robotics, Control Applications to Manufacturing Industries, Flexible Manufacturing and Ergonomics

P Drazan
University of Wales
Institute of Science and Technology
Cardiff, UK

Building Systems

D J Fisk
Department of the Environment
London, UK

Classical Control, CAD of Control Systems, Multivariable Control

N Munro
University of Manchester
Institute of Science and Technology
Manchester, UK

Database Management, Theory and Applications of Distributed Computing for Control

E Gelenbe
Université Paris-Sud
Paris, France

Distributed Parameter Systems, Discrete Systems

J L Lions
Centre National d'Etudes Spatiales
Paris, France

Environmental Measurements, Measurements in Chemistry

Y Sawaragi
Japan Institute of Systems Research
Kyoto, Japan

Environmental Systems and Agricultural Systems

E Halfon
National Water Research Institute
Burlington, Ontario, Canada

Hierarchical Control, Decentralized Control, Model Simplification for Complex Systems Team Theory, Robustness and Sensitivity

A Titli
Centre National de la Recherche Scientifique
Toulouse, France

History of Systems and Control

S Bennett
University of Sheffield
Sheffield, UK

Management Systems, Educational Systems, Health Care Systems, Social Effects of Automation

A P Sage
George Mason University
Fairfax, Virginia, USA

Measurements in the Biological Area

P A Payne
University of Manchester
Institute of Science and Technology
Manchester, UK

Measurements in the Control Area including Intelligent Instrumentation

M G Mylroi
University of Bradford
Bradford, UK

Modelling and Control of Biological Systems, Biomedical Engineering

D A Linkens
University of Sheffield
Sheffield, UK

Pattern Recognition, Fuzzy Systems

M M Gupta
University of Saskatchewan
Saskatoon, Saskatchewan, Canada

Physical Systems Modelling, Identification, Estimation Theory and Applications, Signal Processing

T Kailath
Stanford University
Stanford, California, USA

CONTENTS

P

Packet-Switching Networks

Since the early 1960s, packet switching has evolved from a concept outlined in a technical report to the principal technique by which the worldwide public data network is being expanded. Early international agreement on service and interface specifications has greatly contributed to the success of packet switching.

1. *Milestones of Packet-Switching History*

The concept of packet switching, which consists of formatting data into a series of small packets and transmitting them through a network by means of a store-and-forward procedure from node to node, appeared in a series of Rand Corporation reports in 1960. The first specification for actual implementation was made in 1964 by SITA (Société Internationale de Télécommunications Aéronautiques), when it announced its project for a fast message-transmission network for airline companies.

It was not until between 1966 and 1970, however, with the Arpanet experiment of the Advanced Research Project Agency in the USA, that packet switching was studied and publicized beyond a small circle of early advocates. The years 1971–74 brought refinements of concepts and new experiments like Cyclades and RCP in France and EIN in Europe. During the same period, standardization committees on packet switching were created within CEPT (European Commission of Post and Telecommunications) and CCITT (International Telegraph and Telephone Consultative Committee).

During the second half of the 1970s, the first public packet-switching networks were open to service, with Telenet in the USA (1975), Datapac in Canada (1977) and Transpac in France (1978). In parallel, official endorsement of commonly agreed interfaces took place in CCITT: the X.25 recommendation (1976) for packet-mode terminals, and the X.3, X.28 and X.29 recommendations (1978) for character-mode terminals. In the early 1980s most industrialized countries had opened or had plans to open public packet-switching data networks. The interconnection of these networks provides for the third worldwide public communications service, after telephone and telegraph services.

After X.25 and related recommendations were refined in 1980, computer and terminal manufacturers progressively announced support of packet-switching interfaces in their systems. In particular, IBM gave official support to X.25 in 1980, whereas it had expressed the view in 1976 that the technology was premature.

In addition to public networks, a number of private companies and international organizations have set up their own packet-switching networks.

2. *X.25 Interface and Virtual-Circuit Principles*

Figure 1 presents the interfaces and procedures involved when two sets of data terminal equipment (DTE) communicate with one another by means of a virtual circuit. Both DTEs are assumed to operate in pocket mode and therefore to have interfaces with the network conforming to the X.25 CCITT recommendation. Between DTE_x and DTE_y the following equipment is identified.

(a) DCE_x, the data-circuit terminating equipment; this is part of the network and is situated immediately next to DTE_x.

(b) PSE_1, the packet-switching exchange to which DEC_x is connected; the line between DCE_x and PSE_1 is typically subject to a not negligible transmission error rate, for example 10^{-5} on transmitted bits, a rate which is too high for data-processing applications.

(c) PSE_2 and DCE_y, symmetrical to PSE_1 and DCE_x on the DTE_y side.

Between PSE_1 and PSE_2 there may be a complex network of nodes and lines. Alternatively, PSE_1 and PSE_2 can be one and the same. What X.25 defines is the two DTE–DCE interfaces and the service which is available between them, independent of any implementation within the network itself.

The specification of the X.25 interface is divided into three parts which correspond to the three lowest layers of the ISO model for open-system interconnection: the physical level of X.25 for layer 1, the link level of X.25 for layer 2, and the packet level of X.25 for layer 3. The physical level concerns the interface between the DTE and its local DCE. It specifies the physical connector, the number of interface circuits with their pin assignments, electrical characteristics of circuits and interface supervision signals over these circuits. Detailed specifications of the two permitted alternatives are given in the X.21 and X.21*bis* recommendations. X.21*bis* is compatible with the interfaces of modems which are used for transmission on telephone lines. X.21 is a new interface with improved performance.

The link level concerns the link control procedure which is executed across the transmission line between a DCE and its PSE. Its functions are the following:

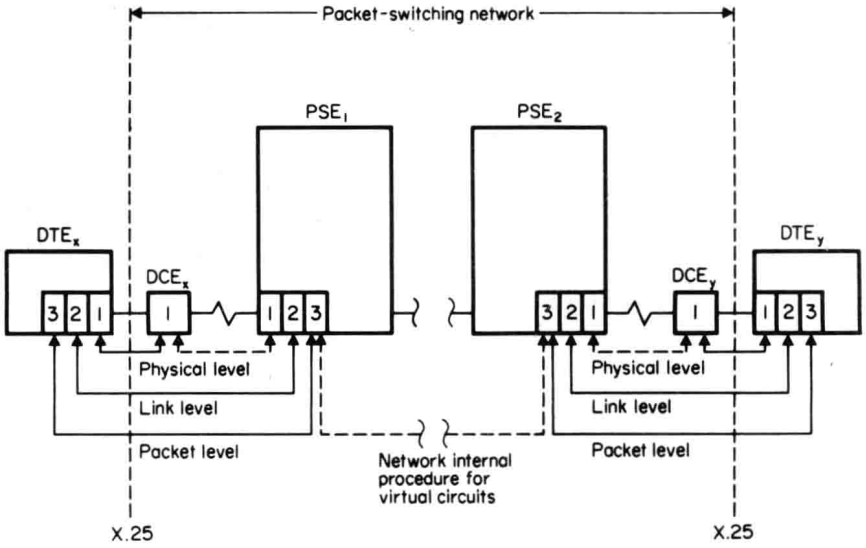

Figure 1
Interfaces and procedures of a virtual circuit between two DTEs

(a) to exercise error correction so that, starting from the not negligible bit error rate, data are transmitted with a negligible residual error rate;

(b) to provide for a common initialization mechanism between DTE and PSE; and

(c) to permit flow control by the DTE and by the PSE so that they can limit the rate at which they receive blocks of information. This flow control may prevent data loss in case of temporary overload of either a processor or its buffer space.

The packet level is the level concerned with virtual circuits. Each DTE can have several simultaneous virtual circuits across its X.25 interface. Each virtual circuit establishes a logical flow of packets between two DTEs. At a given time, a DTE can have virtual circuits with several other DTEs, and several virtual circuits are possible between two DTEs. Each DTE–DCE interface is considered to have a predefined number of logical channels, and every virtual circuit established across that interface is uniquely assigned to one of them. Logical channels at both ends of a virtual circuit are independent (Fig. 2). The DTE which takes the initiative of establishing a virtual circuit—the caller—selects the logical channel for its DTE–DCE interface. The logical channel at the interface of the other DTE—the callee—is selected by the packet-switching exchange. Since a DTE can act both as a caller and as a callee, a rule provides for minimizing the likelihood that both the DTE and the packet-switching exchange will select the same logical channel at the same time, a situation which is called a call collision on the logical channel concerned: packet-switching exchanges select logical channels in increasing order; DTEs must therefore select them in decreasing order.

Logical channel zero is not available for virtual circuits. It can be used locally between a DTE and its PSE for exchanging packets which concern all the logical channels simultaneously. A restart packet on logical channel zero clears all the virtual circuits which have been established or for which an establishment procedure is in progress. If issued by the DTE, the restart packet can contain a diagnostic code. This code is conveyed to the other end of the cleared virtual circuits, as an explanation of why the given DTE has executed a restart procedure. A restart-confirmation packet must be returned in answer to a restart packet by the other side of the DTE–DCE interface. Thus packets which follow the restart procedure are not confused with those

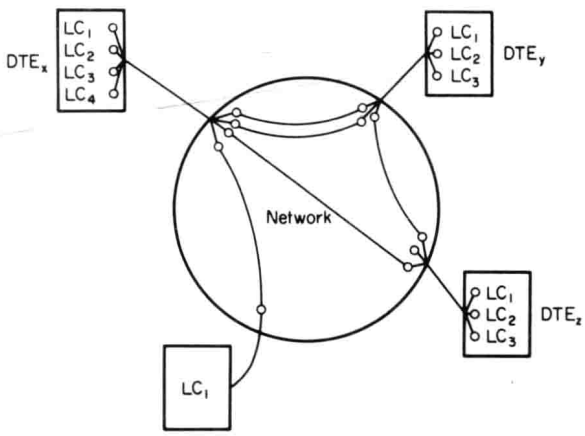

Figure 2
Example of virtual circuits between logical channels of DTEs

which were transmitted, or prepared for transmission, before treatment of the restart packet.

The procedure for establishing, clearing and using virtual circuits applies independently to each logical channel. It is described in Sect. 3 in the simple case where no optional facility is used. Additional elements of procedures apply when some of the optional facilities listed in CCITT recommendations X.2 are used. Between the establishment phase and the clearing phase of a virtual circuit, a logical channel is in the data-transfer state. In this state, the virtual circuit can be used for transmitting data packets and interrupt-packets in both directions. These packets are submitted to a flow-control procedure so that each DTE can control the rate at which it accepts them and so that the network can put an upper limit to the number of packets that it has accepted from a sender and not yet delivered to the corresponding receiver.

Successive data packets are delivered to their destination DTE in the same order as they have been sent by their source DTE, with neither omission nor duplication. If, owing to an internal incident, the network is unable to do so, it either resets or clears the virtual circuit. Resetting consists in indicating that data may have been lost and in resynchronizing both ends; normal service resumes immediately after the reset procedure, on the virtual circuit which is still identified by the same logical channel numbers. Clearing destroys the virtual circuit and returns both logical channels to the state where they are free for the establishment of a new virtual circuit.

3. Logical Channel Procedure for Virtual-Circuit Establishment and Clearing

Figure 3 presents the virtual-circuit establishment procedure in three cases:

(a) the callee accepts the call;

(b) the callee immediately clears the call; and

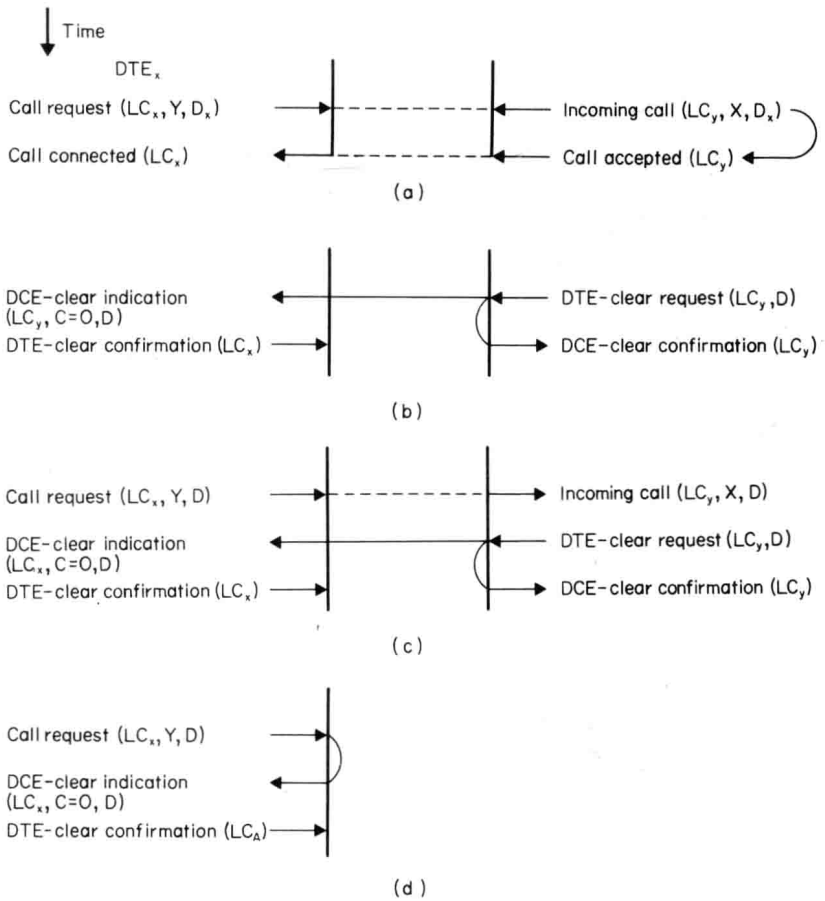

Figure 3
Typical sequences of packets for virtual-circuit establishment and clearing: (a) virtual-circuit establishment; (b) virtual-circuit clearing; (c) immediate clearing by the callee; (d) call rejection by the network

(c) the network is not able to forward the call to the callee.

The call-request packet is issued by DTE_x, the caller, with three arguments: LC_x, the logical channel selected by DTE_x for the virtual circuit; Y, the address of the callee; and D, the call data field, which can have any number of bits up to 16 octets.

If the call is successful, which implies in particular that at least one of the logical channels of the callee is not already engaged in a virtual circuit, DTE_y, the callee, receives an incoming-call packet with the following arguments: LC_y, the logical channel selected by the destination PSE for the virtual circuit; X, the address of the caller; and D, the data field as it was in the call request. Some networks may add the address of the callee to its incoming-call packet, but processing this field is not required. If DTE_y decides, on the basis of the X and D arguments, that a virtual circuit can be established (Fig. 3a), it returns a call-accepted packet with LC_y as an argument. The network then sends a call-connected packet to DTE_x with LC_x as an argument.

If, alternatively, DTE_y decides, on the basis of X and D arguments, that no virtual circuit need be established (Fig. 3b), it answers by a clearing procedure. The clearing procedure, when initiated by the DTE, is executed as shown in Fig. 3b, be it before completion of the virtual-circuit establishment procedure as in Fig. 3c or after the data phase has been entered. The clearing DTE, for example DTE_y, issues a DTE clear-request packet with two arguments: LC_y and D, a diagnostic code of one octet intended for the other DTE. DTE_y then waits to receive a DCE clear-confirmation packet from its PSE before it may issue a new call request on the same logical channel. (This rule ensures that there is no ambiguity about which call is cleared, even in worst-case situations.) As a consequence of receiving the DTE clear-request packet, the network issues to DTE_x a DCE clear-request packet with the following arguments: LC_x; C, a cause field which is set to zero to

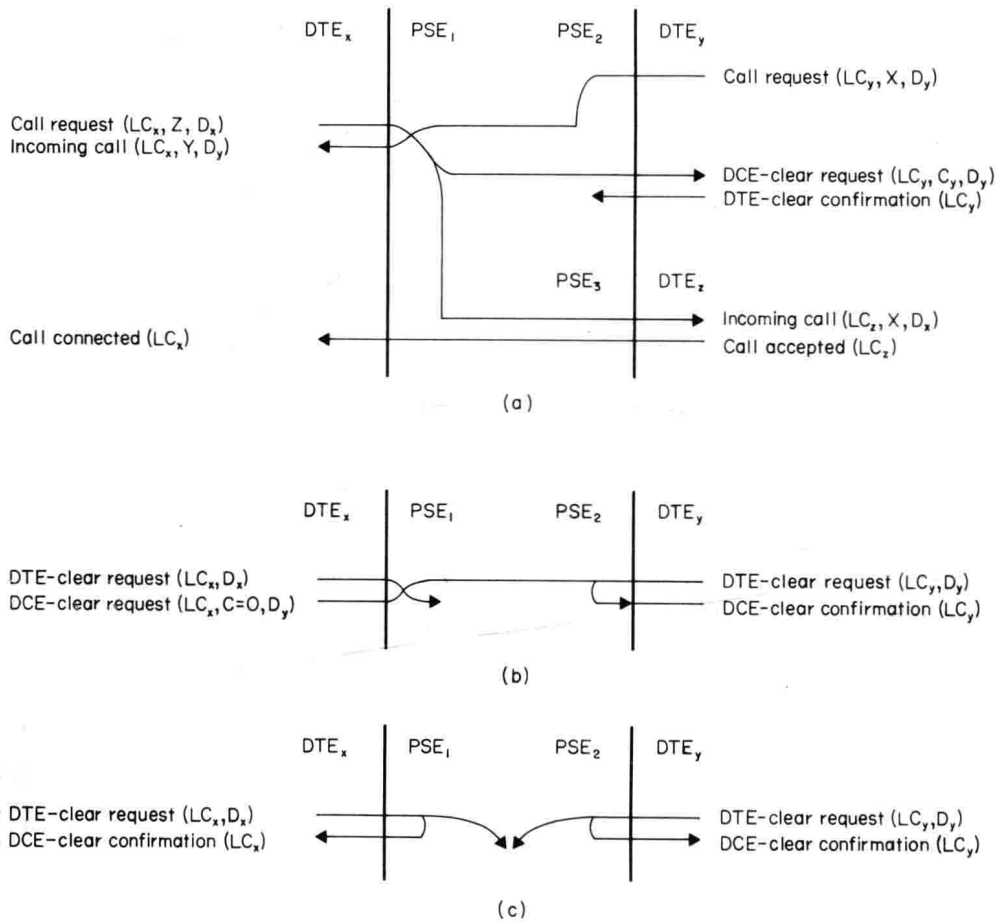

Figure 4
Sequences of packets in some cases of event concurrence: (a) call collision; (b) clear-request collision on a logical channel; (c) clear collision in the network

indicate to DTE_x that the call has been cleared by the other DTE; and d, as it has been provided in the DTE clear-request packet. Logical channel LC_x is considered by the network as ready for a possible incoming call only after a clear-confirmation packet has been transmitted by DTE_x.

The call issued by DTE_x may be unsuccessful for reasons such as called address Y corresponding to no DTE, DTE_y out of order, all logical channels of DTE_y busy (with virtual circuits) or network congestion. In any of these cases, the network answers the call request by a clearing procedure (Fig. 3d). The C and D arguments of the DCE clear-request packet specify the cause, and possibly the error condition, which led to clearing. In this case C is other than zero.

X.25 specifies not only sequences of packets for the above straightforward cases but also for all exceptional occurrences. In particular it specifies that if an incoming call and a call request cross each other over a logical channel (a situation which is called call collision), priority is given to the call from the DTE (Fig. 4a). Another case of simple collision is when a DTE clear-request packet and a DCE clear-request packet cross each other on a logical channel; each one is then considered a confirmation to the other one, so that no confirmation is needed (Fig. 4b). Yet another situation occurs when two DTE clear-request packets on a virtual circuit cross each other within the network. Figure 4c presents such an example. Other cases of special events would go beyond the scope of this article.

4. Logical-Channel Procedure for the Transfer of Data Packets and for Flow Control

For a given virtual circuit, four flows of data cross interface boundaries: one for each direction at each DTE–DCE interface. The procedure is the same for each flow. We will therefore describe it for one of them, one side of the interface being referred to as the sender and the other side as the receiver.

The flow of data packets from sender to receiver is controlled by a so-called window mechanism: the receiver indicates its receive window position, that is, the sequence numbers of the packets that it is ready to receive. This window is defined by its lower edge, PR, and by its size, which is fixed at two packets. The sender is free to send a new data packet if the sequence number of this packet is within the window range; otherwise the packet must wait until a new window position is received. The receiver may increase the lower edge of its receive window up to one beyond the sequence number of the last received packet. When it is ready to accept a new packet, the receiver must transmit an updated window position to the sender without waiting for the receipt of additional data packets. Otherwise deadlock may result.

A new window position can be indicated by means of an ad hoc packet called an RR packet. It can also be indicated by means of an appropriate field in a data packet if one flows in the appropriate direction. (RR packets are the only ones to be used for conveying window positions of a virtual circuit if data flow is in one direction only on that virtual circuit.) When a virtual circuit is established, receive windows are initialized with PR = 0. This means that, for each of the four data flows which cross interface boundaries, data packets numbered 0 and 1 may be transmitted.

Sequence numbers of packets and window positions are treated in modulo 8. Table 1 presents an example of the sequence of data packets and PR progressions across an interface. The chosen sequence is rather an illustration of possible situations than a typical case.

Each data packet carries a delivery confirmation bit or D bit. If this bit is not set, the network can update its receive window with local significance, that is, without waiting for the data packet to be received and the window to be updated at the other end of the virtual circuit. Thus continuous flow of data can be achieved without being constrained by the end-to-end propagation delay across the network. However, if a data

Table 1
Example of flow control by a modulo 8 window position

Packet transmission (data packets and PR progressions)	PR	Window	Next transmitted packet	Permitted packet transmissions
	5	5,6	5	5,6
D5 →	5	5,6	6	6
D6 →	5	5,6	7	
PR6 ←	6	6,7	7	7
PR6 ←	6	6,7	7	7
PR7 ←	7	7,0	7	7,0
D7 →	7	7,0	0	0
PR0 ←	0	0,1	0	0,1
D0 →	0	0,1	1	1
D1 →	0	0,1	2	
PR1 ←	1	1,2	2	2,3

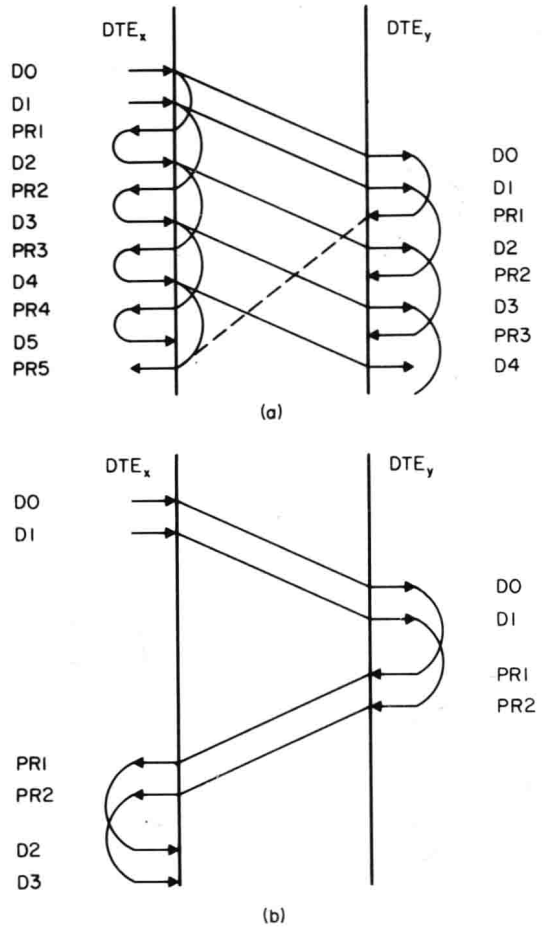

Figure 5
Illustration of the influence of the D bit: (a) continuous flow of data with local significance (D bit set to 0); (b) end-to-end delivery confirmation with D bit set to 1

packet is received by the network with the D bit set to 1, a window update beyond the number of this packet will be returned by the network with end-to-end significance, that is, only after the DTE at the other end of the virtual circuit has indicated a lower window edge beyond the number of this packet. This constrains the rate of data flow to be at most two data packets per end-to-end round-trip delay. On the other hand, this procedure may be used as a means of conveying end-to-end acknowledgement from DTE to DTE without using a data packet for this purpose (Fig. 5).

If the D-bit procedure is used, that is, if D is sometimes set to 1, a receiving DTE may have to update its window to confirm delivery at a time when it is not ready to accept new data packets. For this it can use an RNR packet which gives the new window position but also requests that the transmission of data packets be stopped. Data-packet transmission is permitted again when the receiver transmits an RR packet. The D-bit part of the procedure has been officially part of X.25 since January 1982. However, as of mid-1982, local significance of window updates remains the only universal procedure for international operation.

The rate at which data packets flow across interfaces and across the network depends on the performances of the two DTEs concerned and on the performance of the network. A throughput class is assigned to each direction of a virtual circuit when it is established. A throughput class specifies the nominal number of full data packets per second which can flow across the network if these packets have the D bit set to zero. Because of statistical sharing of transmission and switch resources, the actual data transmission rate is not guaranteed to reach the throughput class 100% of the time; it may also exceed it in some cases. But networks, on the basis of assigned throughput classes, allocate internal resources so that the nominal throughput can be reached most of the time (when sender and receiver introduce no bottleneck themselves). If the receiver is slower than the sender, the sender is flow-controlled by the network so that the number of packets which are in transit within the network remains within controlled limits.

The maximum number of packets in transit varies with the implementation but must at least be as large as the number of packets per second, specified by the throughput class, multiplied by the end-to-end round-trip delay from DTE to DTE. For example, a continuous transmission at 10 data packets per second requires at least a capacity of 10 data packets in transit if the end-to-end round-trip delay from DTE to DTE is of the order of 1 s. Unless specific throughput classes are explicitly requested using an optional facility, default throughput classes which are assigned are the highest ones which are compatible with interface and network performances. For example if a DTE with a DTE–DCE interface at 4800 bits s^{-1} communicates with a DTE with an interface at 9600 bits s^{-1} and if the network supports virtual circuits up to 19200 bits s^{-1}, both directions of the virtual circuits will have a throughput class of 4800 bits s^{-1}, which corresponds roughly to 4.7 data packets per second.

5. Miscellaneous Features of the Logical-Channel Procedure

The user data field of a data packet may contain any number of bits up to a standard maximum length of 1024 bits (128 octets). Some networks impose data fields which contain a multiple of 8 bits. The only universal procedure for international operation is therefore based on octet structured data fields.

A provision has been included in X.25 so that a nonstandard maximum data field length, other than 128 octets, may be selected locally as an optional user

facility. For example, a sender operating with the standard maximum of 128 octets may send data to a receiver which operates, with its local PSE, with a maximum of 256 octets. In such a case it is up to the sender to indicate which data packets can be combined so that large packets can be received. It does so by sending units of data which have natural lengths larger than 128 octets as sequences of one or more full data packets marked with an M bit set to 1, followed by a last data packet marked with an M bit set to 0. By analyzing the sequence of M bits of the packets that it has to deliver, the PSE of the receiver can know which data fields belong to the same unit of information and can thus determine which ones can be regrouped into larger packets. The same procedure makes it possible for a DTE which operates with a nonstandard maximum size smaller than 128 data octets to communicate with a standard DTE and yet to transmit to it full-size 128 octet data fields.

A sequence of full data packets with M = 1 followed by a data packet with M = 0 is called a complete packet sequence. A complete packet sequence can be accompanied by one bit of information called the qualifier bit, or Q bit. The sender must give the same value to the Q bit of all the data packets of a complete packet sequence. The Q bit can be used to multiplex two different data streams on a virtual circuit, for example an ordinary data stream and a stream of supervisory information. This possibility is used in the PAD, described in Sect. 7.

Each DTE can transmit interrupt-packets in parallel with the flow of data and without being constrained by the flow control on these data. An interrupt-packet contains a user data field of one octet. It may be transmitted by a DTE only if none has been previously transmitted, or if an interrupt confirmation packet has been received from the other DTE since the last interrupt-packet has been sent. Within the network an interrupt-packet may overpass a data packet, while the reverse never occurs.

Each of the two DTEs concerned with a virtual circuit may decide at any time to reset it. The effect is that flow control, packet sequence numbering and the interrupt procedure all return to the same state as after virtual-circuit establishment. The reset procedure is similar to the clearing procedure, except for the fact that it does not modify virtual-circuit establishment. The network itself may initiate a reset procedure in case of an internal incident. A cause and diagnostic mechanism allows for distinguishing among the various possible origins of the reset procedure. A reset confirmation indicates that the procedure is completed locally. A collision of a DTE reset request and a DCE reset indication is considered as completing the procedure, obviating the need for reset confirmations. Figure 6 illustrates a reset procedure when a DTE takes the initiative.

In addition to virtual circuits which are established and cleared as described above, X.25 specifies that networks may offer permanent virtual circuits. For these, establishment is requested by an administrative procedure, no clearing is possible and internal incidents are signalled by resets. As of mid-1982, CCITT has yet to specify internetwork procedures extending permanent virtual circuit across several networks. Switched virtual circuits (also referred to as virtual calls) are therefore the only ones available for worldwide communication.

Various optional user facilities are specified by CCITT for the packet mode of operation. CCITT recommendation X.2 gives a list of these facilities, indicating which ones are mandatory on every network which complies with CCITT recommendations. One of the mandatory facilities, called closed user group, allows

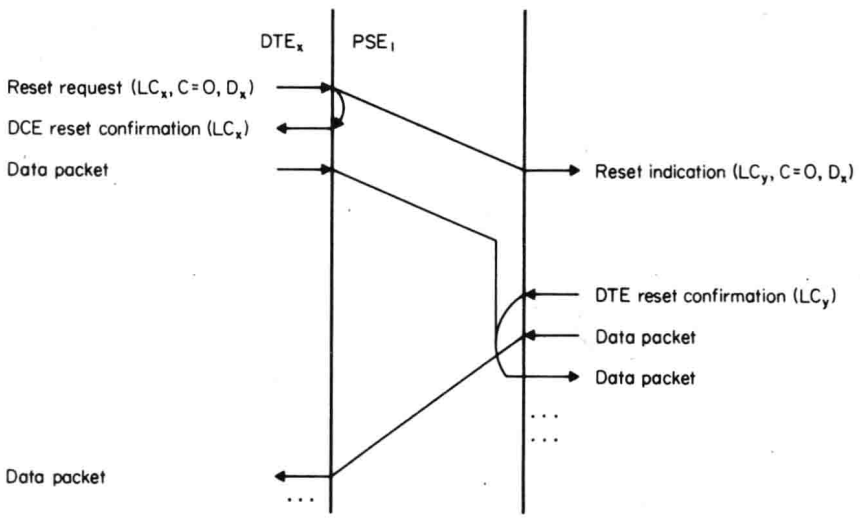

Figure 6
Example of virtual-circuit reset by one DTE

a group of DTEs to have the same level of privacy as they would on a private network: no call can reach one of these DTEs if it does not come from one member of the closed user group, and no call from a member can reach a DTE which does not belong to the group. This facility works the same in circuit switching and in the packet mode.

X.25 also mentions that some networks could offer, in addition to virtual circuits, a datagram service. Detailed specifications are given for this service, but at the time of writing, no service conforming to these specifications has been offered or even announced. Some private networks, in particular local area networks, have been based on mechanisms also referred to as datagram but largely different from the CCITT specification.

6. *Link-Level LAPB Procedure*

The link-level procedure of X.25 conforms to a particular version of the HDLC procedure defined by ISO. The B in LAPB refers to the fact that the first issue of X.25 in 1976 had a link access procedure (LAP) which, having been designed by CCITT before ISO had finalized the HDLC standard, did not conform fully with this standard. The first X.25 networks maintain compatibility with all existing X.25 DTEs by offering both LAP and LAPB procedures.

The link-level procedure can be viewed as subdivided into two sublevels: the frame structure sublevel and the link procedure sublevel. The frame structure sublevel specifies how the flow of bits in each direction is structured into consecutive frames separated by idle signals. Each frame is delimited by an initial flag and a closing flag of eight bits each. Confusion between information contained in a frame and enclosing flags is prevented by modifying every transmitted pattern which would imitate a flag (a 0 is inserted after any sequence of five consecutive 1s, so a 01111110 flag is never imitated). A frame contains, immediately before the closing flag, 16 bits of redundant information: the frame check sequence (FCS). By verifying the value of the FCS, the receiver can ascertain that the contents of a frame have been received without transmission error. Any frame which does not have a complete frame check sequence or which has one which does not match that computed from other bits of information is ignored by the receiver. Thus the frame structure sublevel transforms a channel which has a significant bit error rate (typically 10^{-4}–10^{-6}) into a channel which transfers frames with a not negligible probability of losing frames but with negligible probability of accepting frames which differ from the transmitted ones.

The link procedure sublevel itself can be viewed as having two parts: the link setup and disconnection procedure and the information transfer procedure. Link setup and disconnection normally take place only after the DTE or its PSE is reinitialized. It typically involves the exchange of an SABM frame in one direction and a UA frame in the reverse direction. Special provisions are specified for all cases of transmission errors and special occurrences. During the information transfer phase, flow control is exercised by a window mechanism similar to the packet-level one, but with a window size of seven.

Error correction is achieved by a combination of positive acknowledgements (window progression), negative acknowledgements (REJECT frames, which are transmitted when out of sequence frames are received) and time-out control on acknowledgements. Packets are transmitted in frames of a particular type, I frames. Window progressions are signalled either in I frames from the reverse direction, in RR frames or in RNR frames. RR frames are similar to RR packets of the packet level. RNR frames are used to convey acknowledgements without opening the flow-control window.

A complete specification of the procedure can be found in the X.25 recommendation itself. Its complexity is somewhat greater than this brief introduction would suggest. In particular, the proper handling of all error situations requires careful network design.

7. *Interworking between Start–Stop-Mode and Packet-Mode DTEs by means of a PAD*

In packet switching, the speeds of operation of two communicating DTEs do not need to be the same. As a consequence, it is possible to insert various conversion functions into the path between two DTEs. These may take varying processing times without preventing the orderly exchange of information. This possibility spurred specification of a standard procedure for interworking between start–stop-mode DTEs, (which communicate character by character) and packet-mode DTEs. The conversion functions are conceptually grouped into the packet assembly–disassembly (PAD) facility, which is specified by CCITT recommendations X.3, X.28 and X.29.

X.28 specifies the interface between a start–stop-mode DTE and a PAD, in particular how a virtual circuit, outgoing or incoming, can be established and how parameters which govern PAD operation can be manipulated. X.29 specifies how a packet-mode DTE communicates with a PAD and how characters towards or from the start–stop-mode DTE are formed into packets. It also defines how PAD parameters can be manipulated by means of ad hoc PAD messages (PAD messages are recognized by a Q bit set to 1). X.29 procedures make use of the user data fields of X.25 and specify a particular way of structuring them. In terms of the open-systems interworking model of ISO, X.29 can be viewed as defining one layer above the virtual circuit layer. It does not, however, fit precisely one of the seven layers of the ISO reference model: combined