Dublin City University

School of Electronic Engineering

Flow and Congestion Control in Frame Relay Networks

A thesis submitted as a requirement for the degree of Master of Engineering in Electronic Engineering

July 1993

I declare that this thesis is entirely of my own work and has not been submitted as an exercise to any other university.

Supervisor: Dr. T. Curran

Anuradha Sahgal
B.E.

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master of Engineering is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Sahgal   Date: 19 July '93
(Anuradha Sahgal)

Date: 19 July '93

Abstract

Flow and Congestion Control
in
Frame Relay Networks

Author: Anuradha Sahgal

This thesis describes the algorithms developed to improve the overload performance of frame relay networks. These algorithms were developed by incorporating a number of congestion control techniques in the existing LAPD protocol (as defined in recommendation CCITT Q.920-Q.921). In addition, the performance analysis of the model is given.

Simulation models were developed to evaluate the performance of these networks. A network simulation tool, OPNET, was used for this purpose. The performance of the enhanced model was compared with the existing protocol.

A significant improvement was observed in the throughput and delay performance of frame relay networks. The results were obtained by executing extensive simulations and collecting data over a number of simulation runs. The effectiveness of the congestion control techniques was tested by analysing the dynamic performance of the model.

These techniques, if implemented in the existing protocol, will ensure the good user-perceived network performance at all loads.

Acknowledgements

I would like to thank my supervisor, Dr. Tommy Curran, for his encouragement and guidance during the course of this project. I am also thankful to him for offering me an opportunity and funding to pursue this course.

I also extend my thanks to other members of the Electronic Engineering Department at Dublin City University. I am especially thankful to Dr. Charles McCorkell, John Murphy and Martin Collier for their valuable help, comments and suggestions. I am also grateful to technicians David Condell and Paul Wogan for helping me in my hours of need.

I am also extremely thankful to my fiancee Manish Gulyani for his help, support and encouragement at all stages of this work. I also value the interesting discussions that I had with John Kavanagh. I thank my friends Jeeva Kumar Kanagaratnam, Ziaul Karim and Monica Gillooly for being there when I needed a morale boost. Last but not the least, I extend my sincere thanks to my parents and friends for their constant encouragement and support.

Table of Contents

Chapter 1

Introduction

1.1 Introduction

In the recent years there have been considerable developments in the area of telecommunications. These developments have led to the evolution of the concept of Integrated Services Digital Networks (ISDN). The main objective of these networks is to provide a range of voice and non-voice services over the same digital path via switched and non-switched connections. At the same time, the PC boom has led to the development of a decentralised system consisting of interacting local area networks, in which users exchange large amounts of data in the form of bursts. These changes in the communications pattern have led to an ever-growing need for higher transmission speeds.

The traditional X.25 packet switching networks are however incapable of supporting this increase in user demand. Since the X.25 protocol was developed when most transmission technologies were analogue, it includes extensive error recovery and flow control schemes. The extensive protocol processing at the network nodes causes delay and limits the available transfer speeds to 64 kbps, which is evidently insufficient for the user needs. It is therefore necessary to re-evaluate packet switching to meet the growing demands of the users. In addition, this re-evaluation is taking place with a digital world in mind. As the digital transmission facilities ensure a better quality of data transmission, there is no longer a need for extensive error correcting and flow controlling functions in the network node. These network protocols can therefore provide higher transmission speeds, and meet the demands of the users.

Frame relaying is one such fast packet switching technique. In frame relaying networks, error recovery and flow control takes place on an end-to-end basis. As a result, frame relay technology can offer up to 10 times the throughput of X.25. However, there are concerns regarding the overload performance of these networks.

Congestion can develop in the frame relay network if the offered load exceeds the critical system capacity. Due to the inadequate provision of buffers, packets can be lost in the network. However, the correct and sequential delivery of data is ensured by the end-to-end error recovery mechanism. With the fixed window flow control, the loss of a packet leads to the retransmission of up to a window of packets on all the links of the virtual connection. The retransmission traffic however leads to a substantial increase in the buffer utilisation. This in turn increases the probability of overflows in the network. As a result, there is an abrupt increase in the retransmission traffic. This leads to a sudden loss in throughput. At the same time, the network transit delays grow unacceptably large.

It is evident that the fixed window flow control can not control congestion effectively in the frame relay networks. This is due to the lack of control capabilities in the network. It is therefore necessary to implement other congestion control schemes. These schemes should ensure that the network throughput is maintained at maximal levels at all loads.

The aim of this research is to implement other congestion control techniques in the protocol. These techniques should not interfere with the protocol in the absence of congestion. In other words, the network should continue to operate at maximal efficiency in the absence of overload. When overload occurs, these schemes should be effective in controlling the traffic. In addition, they should prevent the performance of the network from degrading.

1.2 Format of the Thesis

Chapter 2: The development in the area of ISDN is described in this chapter. ISDN supports both circuit-switched and packet-switched connections. The continual need for packet switching with ISDN is described. The existing packet switching protocol (X.25) is evaluated for the ISDN architecture. The support of X.25 terminals by ISDN is also described. The evolution of the fast packet switching technology, especially with respect to ISDN is discussed. This chapter contains the background to the development of ISDN frame relay networks.

Chapter 3: This chapter concentrates on the study of frame relay technology. The operation of the frame relaying protocol is described in detail. In addition, the advantages of these networks are also mentioned. To improve the performance of these networks under overload conditions, some congestion control strategies are suggested.

Chapter 4: To analyse the performance of frame relay networks, a simulation model was developed. OPNET, a network engineering tool was used for this purpose. The development of the model is described in detail in this chapter. To implement the mentioned congestion control schemes, the protocol in the developed model was modified. The ease with which these schemes can be implemented in the protocol is also described here.

Chapter 5: The results obtained by simulating the frame relay network model are presented in this chapter. These results were obtained by implementing the LAPD protocol at the end nodes of the system. The rotating window algorithm was used to control the flow of data in the network. Both the dynamic and the steady state performance of the developed system was studied. A detailed analysis of the results is also provided.

Chapter 6: The results obtained by implementing alternate congestion control strategies in the frame relay network are presented in this chapter. The results obtained for adaptive windowing, congestion notification and stop duration are analysed. An analysis of the results portrays a significant improvement in the overload performance of the network.

Chapter 7: This chapter contains the conclusions about the project and the research study. It is concluded that reasonable results were obtained by implementing the proposed strategies.

Once these strategies are implemented in the protocol, there need no longer be any concern regarding the overload performance of the frame relay networks.

Chapter 2

Integrated Services Digital Networks

2.1 Introduction

The rapid advances in computer and communications technologies have resulted in the increasing merger of these two fields. Merging and evolving technologies, coupled with increasing demands for efficient and timely collection, processing and dissemination of information are leading to the development of integrated systems that transmit and process all types of data. The ultimate goal of this evolution is the development of the Integrated Services Digital Networks (ISDN)[24]. This chapter will mainly concentrate on the development in the area of ISDN, with special reference to the channel connections the ISDN will support.

2.2 Integrated Services Digital Networks (ISDN)

ISDN may be seen as the logical progression of the digitisation of the telephone network and the development of digital networks. The Integrated Services Digital Network (ISDN) is defined as a network, in general evolving from a telephony Integrated Digital network (IDN), that provides end-to-end digital connectivity to support a wide range of services, to which users have access by a limited set of multipurpose user-network interfaces.

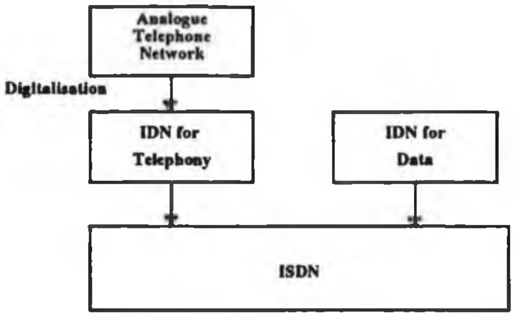


Fig. 2.1 **Development of ISDN using the telephone network as a basis**

The concept of the ISDN is to provide a network in the future which will be able to provide a range of voice and non-voice services over the same digital switches and paths via switched and non-switched connections. In addition, the development of new ISDN services must be done in the context of existing digital facilities and existing services so that they can coexist and inter work during the transition period.

2.3 ISDN Reference configuration

The CCITT has defined reference configurations for the interface point of user premises equipment to an ISDN network. Two concepts are used to describe these configurations.

The first concept is one of *functional groupings* which describe the set of functions which may be required in ISDN arrangements. The specific functions in a functional grouping may be performed by one or more pieces of equipment.

The second concept is one of *reference points* which are given alphabetic letters (R, S, T, U). These are the conceptual points at the conjunction of two functional groupings. The reference points may or may not correspond to a physical interface between pieces of equipment.

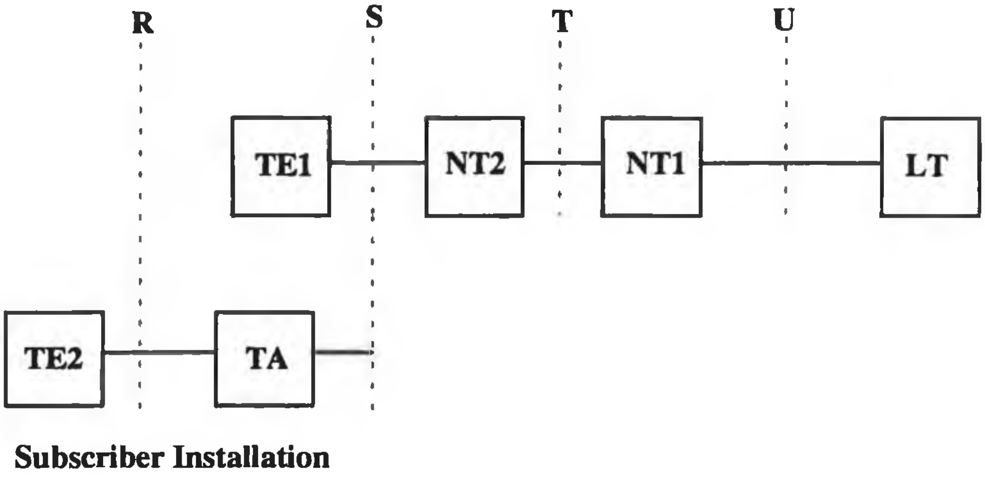


Fig. 2.2 **ISDN Reference Configuration**

In figure 2.2 each box represents a functional grouping.
LT (Line Termination);

NT1 (Network Termination 1) includes functions associated with the physical and electrical termination of the network (i.e. equivalent to Open Systems Interconnection [OSI] layer 1);

NT2 (Network Termination 2) includes intelligent functions (i.e. equivalent to OSI layers 1, 2 and 3). Typical examples of NT2 devices are Private Automatic Branch Exchange (PABX) and Local Area Network (LAN);

TE1 (Terminal Equipment 1) is terminal equipment with an interface that complies with the ISDN user-network interface recommendation (e.g. digital telephone);

TA (Terminal Adaptor) provides adaptor functions for connection of Terminal Equipment 2 (TE2) to the ISDN;

TE2 (Terminal Equipment 2) is terminal equipment with an interface that requires a terminal adaptor (TA) to be compatible with the ISDN user-network interface recommendation (e.g. data terminal with V.24/V.28 (RS-232C) interface).

A series of reference points are also indicated in figure 2.2 to isolate the functional groupings of the ISDN user-network interface. **Reference point T** (terminal) corresponds to a minimal ISDN network termination at the customer's premises. It separates the network provider's equipment from the user's equipment. **Reference point S** (system) corresponds to the interface of individual ISDN terminals. It separates user terminal equipment from network-related communications functions. **Reference point R** (rate) provides a non-ISDN interface between user equipment that is not ISDN-compatible and adaptor equipment. **Reference point U** (user) provides an interface that describes the full-data signal on the subscriber line.

2.4 ISDN Channel Types

The ISDN bit pipe supports multiple channels interleaved by time division multiplexing. Some of the channel types have been standardised are channel types B, H and D.

B-channel is designed for a number of different applications of digitally coded information. The B-channel consists of one 64 kbps channel and may be used for circuit-switched, permanent-switched or semi-permanent connections. The B-channel can be used both for telephony (coded to 64 kbps) and data (circuit-switched and packet-switched for speeds up to 64 kbps) [9].

A user subscribing to a basic access structure will be provided with two 64 kbps B-channels and a single 16 kbps D channel (2B + D). Each B-channel is capable of supporting independently a wide range of standardised communications services, whilst the D-channel is used principally for transporting outband signalling messages. In addition, the D-channel may also be used to convey packet-switched data services.

The primary rate structure provides the customer with a primary rate of 2048 kbps within Europe and 1544 kbps within North America. At the 2048 kbps primary rate the interface structure supports thirty 64 kbps B-channels and a 64 kbps D-channel, and at 1544 kbps it supports twenty-three 64 kbps B-channels and a 64 kbps D-channel. The primary rate interface is intended for PBX connection.

2.6 ISDN Service Aspects

The ISDN will provide a variety of services, supporting existing voice and data applications as well as providing for applications now being developed. The ISDN is expected to handle a variety of data types, covering a range of applications as diverse as very low bit-rate control and alarm channels for the home and business, interactive information services, electronic mail, digital voice, facsimile, file transfers and wide band digital services among many others.

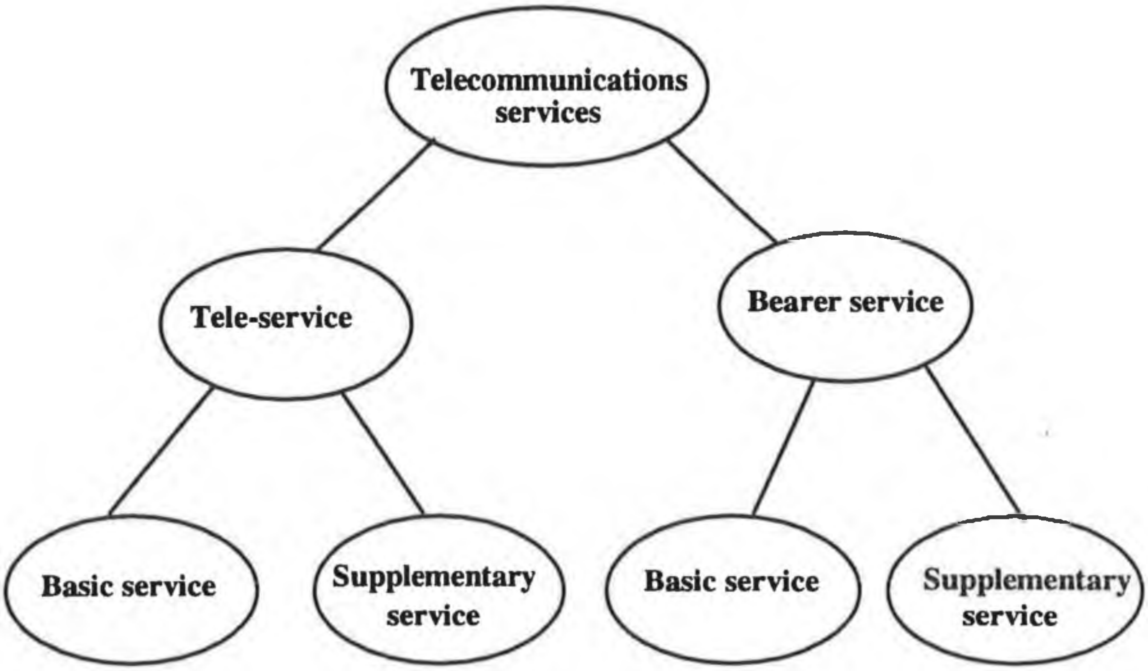


Fig. 2.4 **Classification of Telecommunication Services**

Three types of services are defined by CCITT: bearer services, teleservices and supplementary services.

Bearer services provide the means to convey information between users in real time and without alteration of the content of the message. These services correspond to the lower three layers of the OSI model. Bearer services define requirements for, and are provided by network functions.

Teleservices include terminal as well as network capabilities, and combine the transportation function with the information processing function. They employ bearer services to transport data and in addition, provide a set of higher-layer functions. These higher layer functions correspond to OSI layers 4 through 7. Examples of teleservices are telephony, teletex, facsimile, videotex and message handling.

Supplementary services cannot be used alone and may be used in conjunction with one or more of the bearer or teleservices. These services can be invoked on demand and are aimed at giving support and guidance to the user, and at achieving a high degree of user friendliness. An example of this service is reverse charging.

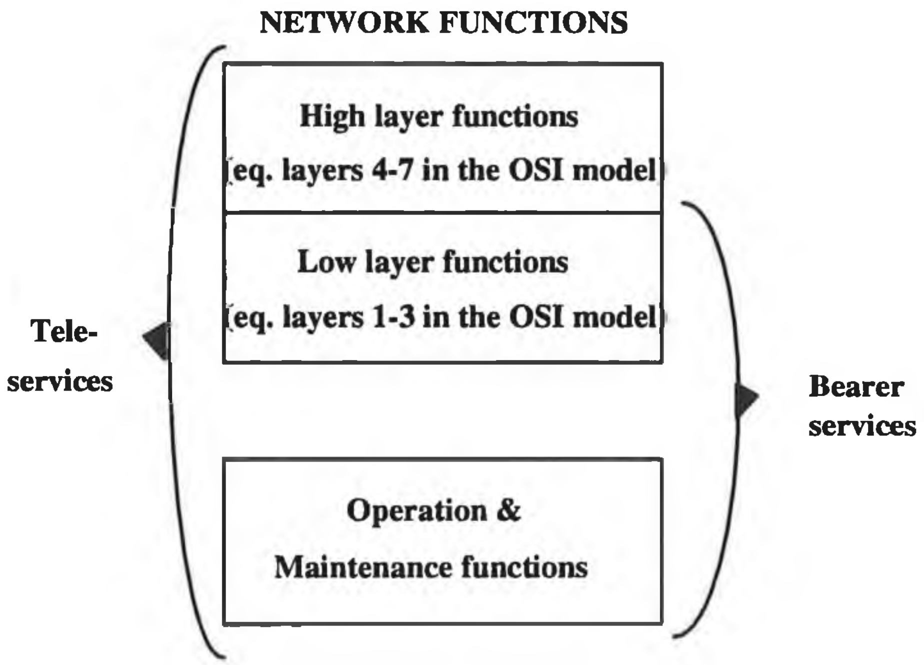


Fig. 2.5 **Connection between concepts, services and functions in ISDN**

2.7 ISDN Protocol Reference Model

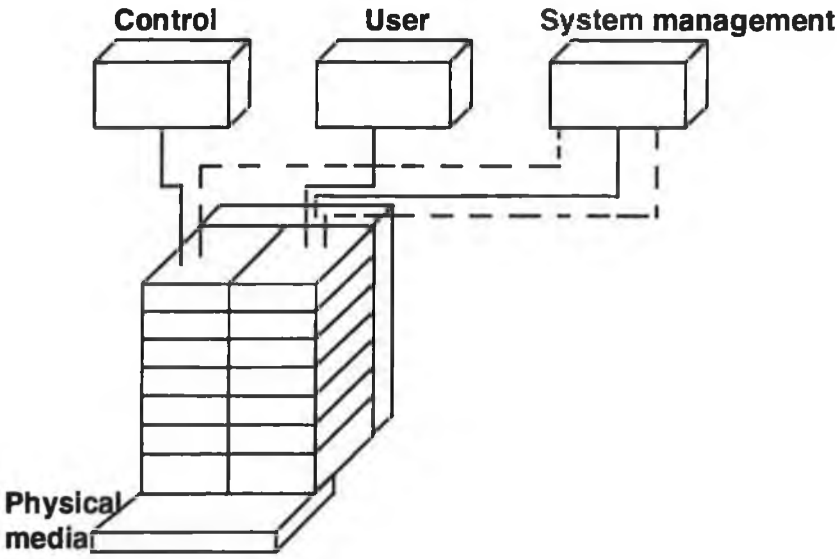


Fig. 2.6 **ISDN Protocol Reference Model**

The ISDN protocol reference model provides a structure for the user/network interface. It is based on the layer principles of the Open Systems Interconnection (OSI) reference model. This model provides for a clear separation of call control (i.e. signalling information), user information transport functions and management functions through all the layers of the OSI model. Signalling information is conveyed on a D-channel and user information is conveyed on bearer channels. The ISDN structure therefore provides greater flexibility than existing user/network interfaces (e.g. X.25) which are based on the use of in band signalling. This is because in the case of ISDN, a single user/network access may be used to provide for a variety of services hence allowing for a fully integrated architecture[23]. The D-channel may be used to provide call control for a number of bearer channels each of which may be used for access to a different service from the network.

The lowest three layers of the ISDN protocol reference model are responsible for providing the network services. The function of these layers are further described below.

2.7.1 Layer 1 - Physical Layer

CCITT recommendations I.430 and I.431 describe the physical layer for the basic rate and the primary rate user-network interfaces respectively. These recommendations specify the requirement for a balanced metallic transmission media having a capability, for each direction of transmission, of supporting 192 kbps and 1544/2048 kbps respectively[16].

The ISDN physical layer supports the requirements for
1) Transmission of encoded bit streams for B-channels (64 kbps) and D-channels (16 or 64 kbps), plus timing and synchronisation
2) Activation and deactivation of subscriber terminals and/or network terminating equipment
3) Physical layer maintenance functions
4) Indication of physical layer status to the higher layers

2.7.2 Layer 2 - Data Link Layer

The Data Link Layer specification for ISDN, called the Link Access Procedure on the D-channel (LAPD) is described in the CCITT recommendation I.440 (general aspects) and I.441 (detailed specification). LAPD uses the same principles and terminology as Recommendation X.25 (LAPB) for packet mode terminals and ISO 3309 and 4335 (High Level Data Link Control) for frame structure and elements of procedure. X.25 is described in the next section.

Flag	**Address**	**Control**	**Data**	**Frame Check Sequence**	**Flag**

Fig. 2.7 **LAPD Frame Format**

LAPD messages are transmitted in frames, delimited by flags. Flags are unique bit patterns not permitted within the information fields of messages. The LAPD is shown as a hierarchy of two groups of protocol functions: peer-to-peer **procedural aspects** and **core functions.**

The peer-to-peer procedural aspects include:

1) One or more data-link-connections on a D-channel: Since ISDN allows the connection of up to eight physical terminals on each basic access (BA), the signalling information for all these terminals is multiplexed on the D-channel. The terminals sharing the D-channel can be telephones, circuit-switching devices, or packet-switching devices. To ensure that the exchange can distinguish between the different types of signalling, The LAPD employs a two-part address consisting of a terminal-endpoint identifier (TEI) and a service access point identifier (SAPI). The TEI is the address indicating the address of the physical terminal and the SAPI is used to indicate the type of traffic used.

2) Frame delimiting, alignment and transparency: The eight-bit flag sequence 01111110 that appears at the beginning and the end of the frame is used to establish and maintain synchronisation. To eliminate the possibility of this sequence appearing in the middle of the frame, bit stuffing is used. In bit stuffing, a zero is inserted at the transmitter any time that five ones appear outside the flag fields; and these zeros are removed at the receiver.

3) Sequencing: To ensure that the information sent by the transmitter arrives at the receiver in the correct order, sequencing is used. A three-bit number N(s) in the I-frame is used to represent the sequence number of the I-frame.

4) Error detection: The sender and the receiver agree on a Generator Polynomial G(x) at the beginning of the call. The transmitter calculates the checksum of the frame by using this G(x). It then appends the checksum to the end of it. This is represented by the Frame Checksum (FCS) field of the frame. When the receiver gets the frame, it divides it by the Generator polynomial G(x). If there is a remainder, it means that there has been a transmission error and the receiver discards the packet [25].

The core functions include:
1) Flow control: Flow control is required in a network to regulate the transmitter-to-receiver traffic flow so as to protect the network from problems related to overload and speed mismatches. The main functions of a flow control algorithm are to prevent throughput degradation and loss of efficiency due to overload, avoid deadlock, allocate resources fairly among competing users, and to match the speed between the network and its users.

As the capacity of the virtual circuit is limited among other things by the capabilities of the receiving DTE (for instance, the speed of its connecting line), the transmission of the data by the network to the receiving DTE should be controlled by the receiver itself. For this purpose, the receiving DTE sends transmission authorisation to the network, which are passed on by the network to the transmitting DTE.

The send sequence number N(s) and receive sequence number N(r) are used for this purpose and the detailed description of this mechanism is given in Chapter 3.

2) Recovery from detected transmission, format, and operational errors: When the receiver detects an erroneous frame, it discards the frame. The protocol recovers from errors by using information and supervisory (Receiver Ready/REJect) frames. The error recovery mechanism is described in detail in Chapter 3.

The LAPD protocol supports point to point and broadcast data-link configurations. Operations of the LAPD service may be with acknowledged or unacknowledged frames. In the former case, error recovery based on retransmission of frames, and flow control mechanisms, are provided [16]. In the latter case, no error recovery or flow control mechanisms are defined.

2.7.3 Layer 3 - Network Layer

This protocol is the key to ISDN call control. The procedures currently defined are for the control of circuit-switched connections, packet-switched connections, and user-to-user signalling connections.

The functions performed by the layer 3 include:
1) processing of primitives for communicating with the data link layer
2) generation and interpretation of layer 3 messages for peer level communication
3) administration of timers and logical entities used in the call control procedures
4) administration of access to resources including B-channels and packet layer logical connections
5) checking to ensure that services provided are consistent with user requirements

Layer 3 also provides the following functions: routing, relaying, network connection multiplexing, conveying user-to-network and user-to-user information, error-detection and recovery, sequencing and flow control.

2.8 ISDN Channel Connections

Integrated Services Digital Networks support a variety of applications using both **switched** and **non-switched** connections. Switched connections in an ISDN include both **circuit-switched** and **packet-switched** connections and their concatenations [CCITT Recommendation I.120 (1984)].

In the case of **circuit-switching,** a private transmission path is established between any pair or group of users attempting to communicate and is held as long as transmission is required. Because these networks statically reserve the bandwidth in advance, any unused bandwidth on the allocated circuit is just wasted. This technology usually provides a blocking mode of operation, with other users denied access to the network. These connections are mainly used for voice and data traffic.

In **packet-switched** technology, blocks of data are individually transmitted from a source to a destination across a network. The packets from multiple users share the same transmission path and distribution facilities. They are stored and forwarded at each node along a path, sharing buffer and link transmission resources with all other packets being transmitted across a given link on the path.

Circuit-switching alone is not enough to transport data in the ISDN network. This is because for many data services, such as point of sale and credit card checking, the 64 kbps bandwidth of a B channel is a waste of resources. Many of the existing terminals need only 1.2 to 2.4 kbps for normal operation. In some European countries more than 80 percent of the traffic in the data network uses speeds lower than 2.4 kbps [22].

In addition, the needed quality for transmission may not always be available in circuit-switched connections. Presently, three information transfer capabilities exist in circuit-mode: (1) voice quality, (2) modem quality, and (3) bit integrity quality. For the first two types, the quality may be insufficient for the data unless sophisticated subscriber equipment is used. The third type is only available in a network that is fully digital from terminal to terminal.

Packet-switching guarantees bit integrity quality. Furthermore, the network can be used in a much more efficient way. This is because the D-channel can also be used for data transfer. In addition, calls can be multiplexed on one link so that the silence periods on the channels is decreased.

2.9 Existing Packet Switching Technology

The existing packet-switching protocol, X.25 and the evolution of fast packet-switching technologies is described in this section.

2.9.1 X.25

The CCITT recommendation X.25 specifies an interface between a user and a packet-switched network. This standard is almost universally used for interfacing to packet-switched networks and will be employed for packet-switching in ISDN [24].

X.25 is organised as a three-layer architecture, corresponding to the lowest three layers of the Open Systems Interconnection (OSI) model.

Layer 1 is the physical transmission channel which handles electrical interface, sequencing, circuit-identification, and so on. It supports one layer 2 connection [22]. This layer ensures that a valid physical connection exists between the Data Terminating Equipment (DTE) and the Data Circuit-terminating Equipment (DCE). CCITT protocol X.21 is used for this purpose [21].

Layer 2 takes care of sequencing, redundancy, error detection, frame or packet retransmission, multiplexing of packets, and quality of service parameters [22]. The link-level protocol at the data link layer is a subset of High Level Data Link Control (HDLC), labelled LAPB (balanced link access procedures) [21].

It is at the third, network layer that X.25, as an interface architecture, is actually distinguished. This layer is called the packet level in X.25 terminology. **Layer 3** uses layer 2 functions and provides support for signalling and data transmission by exercising the necessary network connections, flow control, release, and so on.

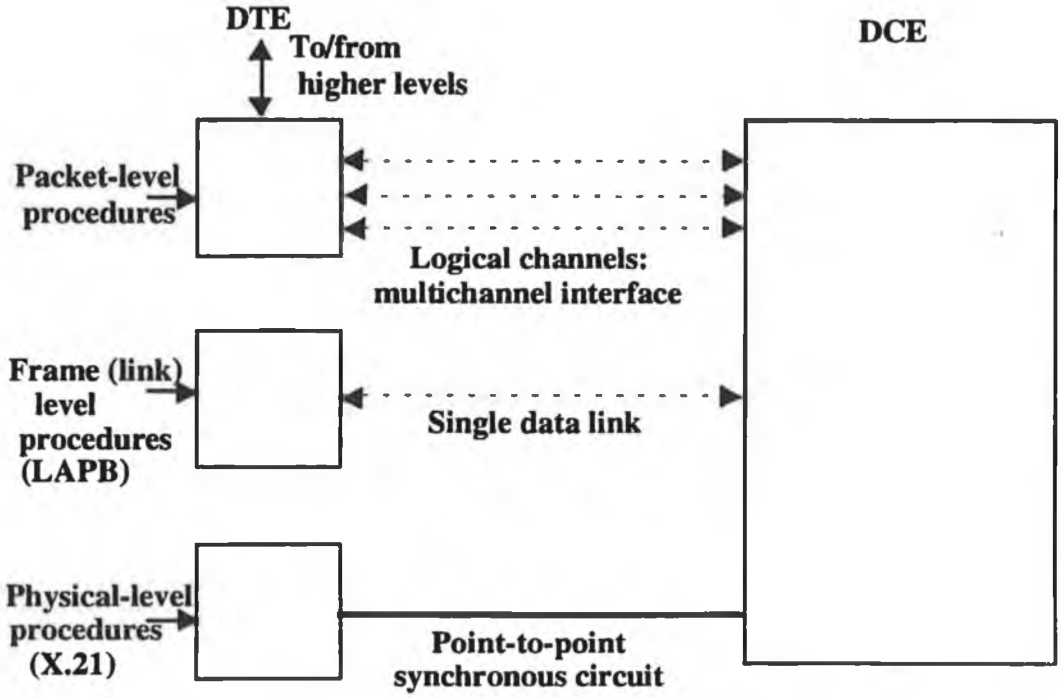
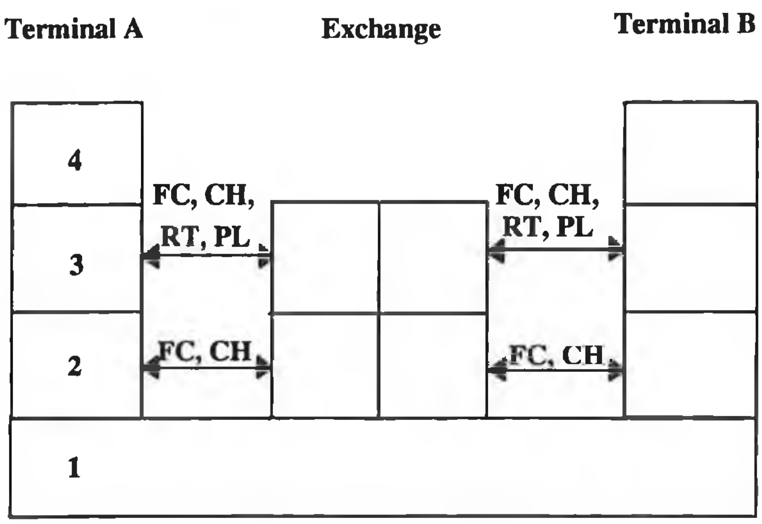

DTE
To/from higher levels
DCE
Packet-level procedures
Logical channels: multichannel interface
Frame (link) level procedures (LAPB)
Single data link
Physical-level procedures (X.21)
Point-to-point synchronous circuit

Fig. 2.8 **X.25 Layers**



FC	**Flow Control**
RT	**Routing**
CH	**Congestion Handling**
PL	**Packet Loss Treatment**

Fig.2.9 **Packet Switching (X.25 Protocol)**

X.25 is an effective communications protocol for meeting intermittent data communication needs. X.25 does not stipulate pre-allocation of a specific portion of the bit flow to a specific connection between two users i.e. the whole bit flow is available to the user who has data to send at the moment [15]. This means that several so-called "virtual connections" are set up in parallel over the same physical link, each of which is allocated a logical channel: a unique identity indicated in the data packet header. The X.25 protocol is based on the store-and-forward principle. Packets entering the network are stored in a buffer and passed on in the order of their arrival. The transfer over each individual link is monitored separately, and an error results in initiation of retransmission on the link. This enables correct data transfer even in networks with heavily disturbed connections.

2.9.2 Support of X.25 terminals by an ISDN

The requirements for ISDN are based on the need to provide access from X.25 terminals to existing X.25 packet data networks [14]. The support of X.25 terminals by an ISDN is defined by Recommendation I.462 (X.31).

This recommendation recognises that the two networks (i.e. packet- and circuit-switched) may not in fact merge to become a true ISDN, but may remain complementary networks, each suited to different services and facilities. X.31 addresses the problems of integrated access procedures between the two networks.

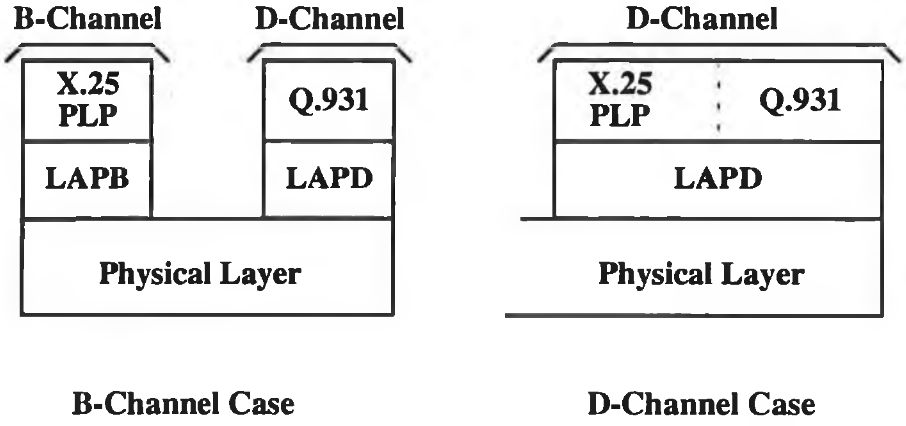


Fig.2.10 **X.31 Access Protocol Suite**

Since ISDN is based on the digital telephony network, which provides 64 kbps connectivity, it cannot be assumed that all digital exchanges are capable of packet-switching. Therefore two scenarios are described in I.462 for the network handling of packet-switched connections. They are the minimum and maximum integration scenarios.

The **minimum integration scenario** refers to the transparent handling of packet calls through the ISDN [14]. A transparent circuit-switched connection is provided from the user terminal to a port of the packet-switched network. Only access via the B-channel is possible. Support is given to packet calls on a physical 64 kbps semi-permanent or switched B-channel.

In this case the switching of the X.25 call is left to the Packet Switched Public Data Network (PSPDN) [22]. The ISDN network provides only access to the PSPDN, and the user is a subscriber of the ISDN as well as of the PSPDN. This packet handler then expects that a message interchange is started between itself and the subscriber as for the set up of an X.25 call in the PSPDN. Frames can be interleaved on the same B-channel, or multiplexing techniques can be used to split one physical channel of 64 kbps into several channels of a lower bandwidth, but the packets are routed unchanged to the packet-switching network, where the content of the packet is analysed to execute the switching function.

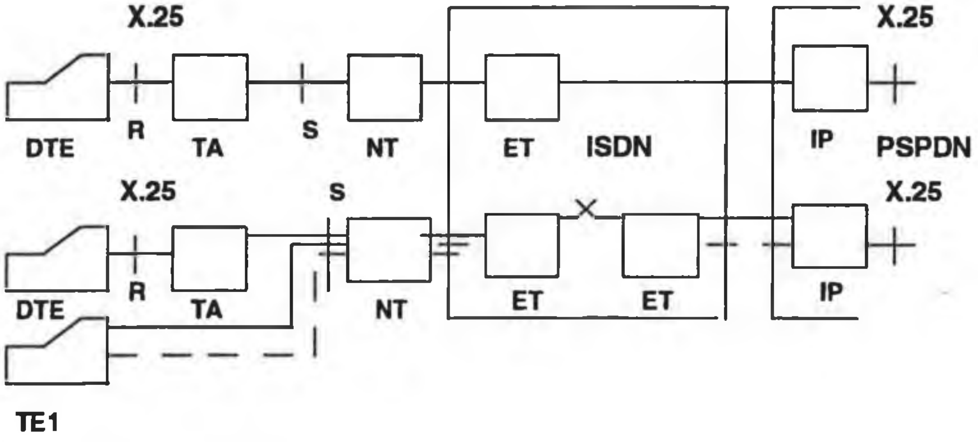


Fig 2.11 **Minimum Integration Scenario**

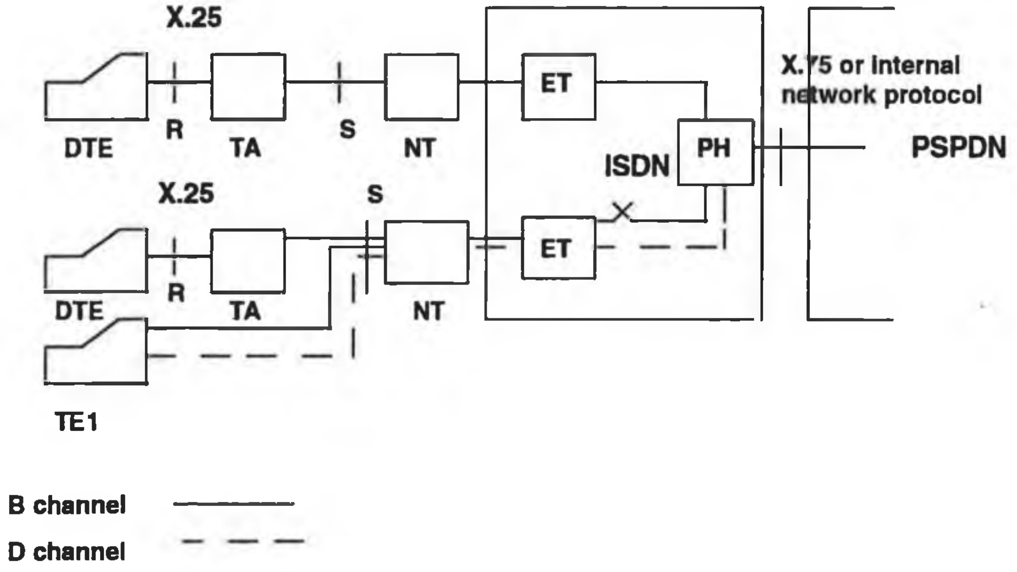


Fig. 2.12 **Maximum Integration Scenario**

Legend for figs. 2.11 and 2.12:

DTE - Data Terminating Equipment;

TE1- Terminal Equipment 1;

TA- Terminal Adaptor;

NT- Network Termination;

ET- Exchange Termination;

IP- Interworking Port;

PH- Packet Handler;

PSPDN-Packet switched Public Data Network

The **maximum integration scenario** refers to the provision of a packet handling function within the ISDN. Both B- and D-channel access is supported, with the packet handler performing the necessary processing for packet calls, standard X.25 functions for X.25, as well as path setting functions and possibly rate adaptation [14]. The call is set up using the D-channel signalling capabilities as for circuit switching. Packets can be sent over the D- or the B-channel [22].

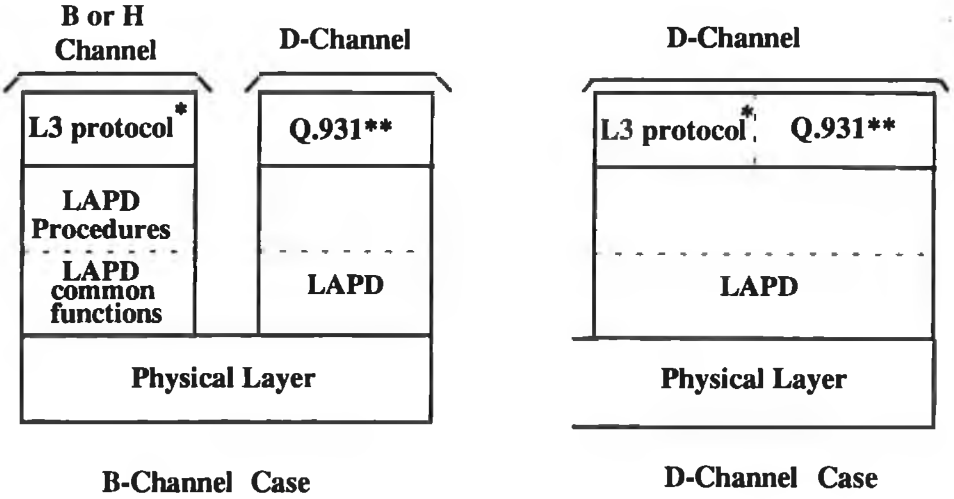
The procedure for setting up calls for each channel type is different. The procedures for B-channel access are separated into a two-stage set-up procedure. The ISDN access circuit is first established using LAPD signalling procedures on the D-channel, followed by the control phase of the virtual circuit(s) using X.25 procedures on the B-channel [14].

D-channel access is on a 'permanent' access basis with no establishment phase being required across the ISDN. This D-channel access requires only X.25 procedures to establish a call into and across the packet network.

It is evident from above that the X.31 approach has shortcomings owing to the complex interaction of call control procedures in the X.25 and the ISDN networks. In X.25, the signalling information is carried in band, whereas in ISDN it is carried out of band in the D-channel. With the X.31 approach, two call control phases are required, one out of band to set up the bearer channel using signalling procedures on the D-channel, followed by in band X.25 virtual circuit signalling procedures on the B-channel.

2.10 Fast Packet Switching

The X.25 packet switching protocol was developed to operate in an environment characterised primarily by relatively low-speed transmission facilities (<= 64 kbps) and relatively high bit error rates. It was therefore necessary to have link-by-link error and flow control. However, in today's era of integrated digital networks, high speed transmission facilities (> 1 Mbps) with very low bit error rates are becoming increasingly available. Emerging trends indicate that high-speed transmission will be used with optical fibre transmission providing efficient statistical multiplexing and minimum delays, together with advanced signal processing and very high-speed technology in end-user equipment and network nodes, respectively. This evolution has led to the development of the concept of **fast packet switching**, which refers to the exploitation of packet switching in a high-speed technology environment [24]. Fast packet switching is different from traditional X.25 packet switching in that it requires only one level of protocol processing at each intermediate node. In addition, this technique dispenses with the need for extensive link-by-link error and flow control. Services that require error-free transmission employ some higher-level error recovery protocol.



*	User information transfer functions only

**	All bearer channel control functions

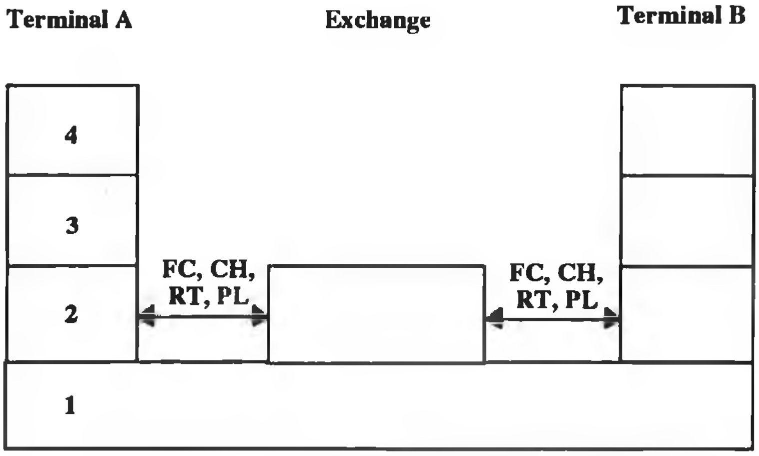
Fig.2.13 **ISDN Packet-Mode Access Protocol Suite**

Fast packet switching is an attractive technology for incorporation into the ISDN. In addition, this technique overcomes the problems encountered in the X.31 approach of supporting X.25 terminals by an ISDN, by providing out of band signalling.

2.10.1 Further Developments in ISDN Packet Switching

The signalling problems arising with Recommendation X.31, coupled with the advancement in transmission technology have led to the evolution of fast packet switching networks. These networks will be incorporated into the ISDN. Depending on the degree of control exercised by the network on the data sent between two users, different packet handling techniques are recognised. The degree of control exercised by the network for processing bearer channel information depends on network capabilities, requested constraints and quality of service for a given instance of communication [23]. All the schemes mentioned below require a complete processing of the Q.931 signalling protocol.

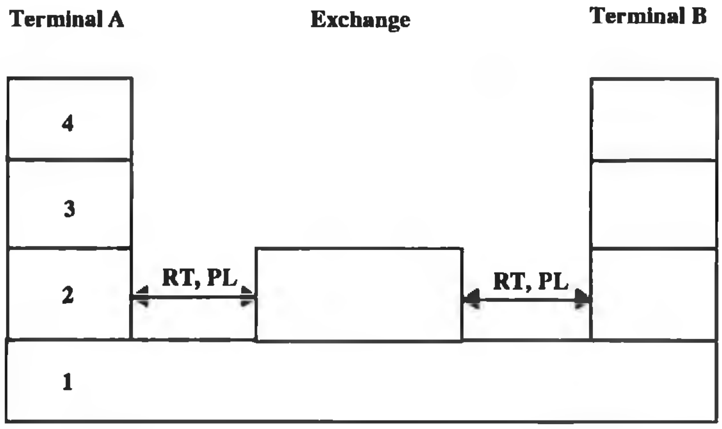
Frame switching differs from traditional packet switching in that it does not analyse a frame down to the level of the packet i.e. there is no layer 3 control. Flow control is done at the same place routing is done, so big users cannot disturb small users [22]. In frame switching, the network terminates the link at a point at which information can be temporarily stored in order to perform speed adaptation between the two extremes (i.e. flow control).



FC	Flow Control
RT	Routing
CH	Congestion Handling
PL	Packet Loss Treatment

Fig. 2.14 **Frame Switching**

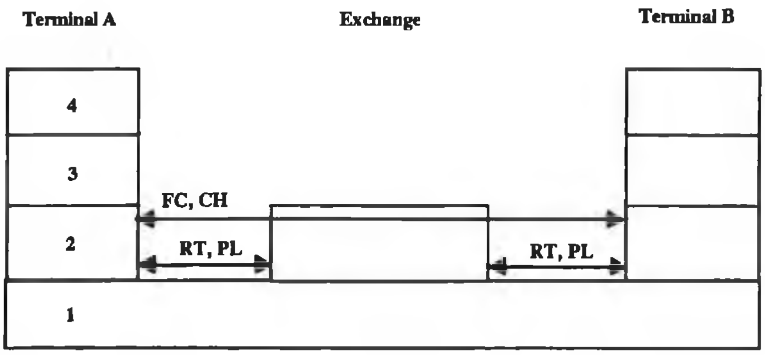
Frame relaying is different from frame switching in that the network intervenes at the core layer functions of the LAPD protocol.



RT	Routing
PL	Packet Loss Treatment

Fig. 2.15 **Frame Relaying/Frame Relaying type 1**

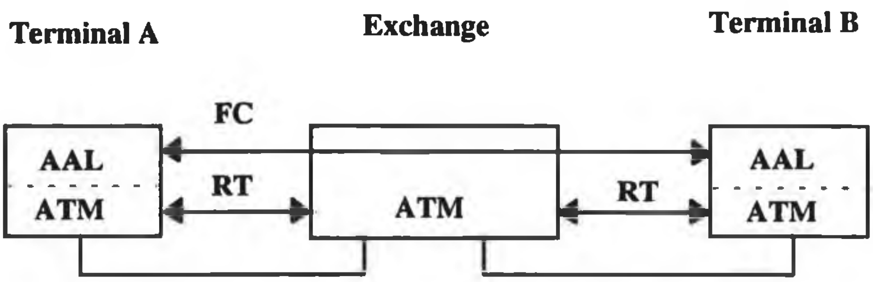
It does not provide a buffer point in the network for speed adaptation. No flow control is provided by the network and this is left to the terminals at both ends. Frame relaying guarantees the delivery of the frames, but frames that are lost in the network cannot be recovered unless retransmission is requested by the receiving terminal. This technique will be applied only in high-quality transmission networks, such as those based on optical fibre links. There are two variants of frame relaying - type 1, with no lost frame recovery, and type 2, which foresees an edge-to-edge lost frame recovery. Combinations of frame relaying and frame switching nodes are also possible in the network.



FC	**Flow Control**
RT	**Routing**
CH	**Congestion Handling**
PL	**Packet loss treatment**

Fig. 2.16 **Frame Relaying type 2**

Cell relaying is a variant of frame relaying type 2 because end-to-end lost frame recovery is specified. However, cell relaying uses fixed length frames called cells in contrast to variable frame sizes in frame relaying. Both Asynchronous Transfer Mode (ATM) and Metropolitan Area Networks (MAN) use cell relay technology. As in frame relaying, cell relaying also minimises the amount of processing by the network and assumes high-quality transmission links to transfer the data.



ATM	**Asynchronous Transfer Mode**
AAL	**ATM Adaptation layer**
FC	**Flow Control**
RT	**Routing**

Fig. 2.17 **Cell Relaying**

2.11 Conclusions

The arrival of narrow band ISDN has added the dimension of services to telecommunications. In addition, there is an increase in user demand to use data communication networks for new ISDN applications and for interconnecting LANs. The existing packet switching protocol, X.25 provides inadequate data rates for these new applications. This has led to the investigation of fast packet switching techniques. These techniques are being developed with the intention of implementing them in the ISDN. In this study, we are going to focus on frame relay networks.

As the cost of providing service within the network is directly related to the level of network protocol intervention, and the performance obtainable for a given connection is also dependent on the protocol intervention required in the network, frame relay switches have a much better cost-performance ratio as compared to conventional packet switches. It is estimated that for the same cost, a frame relay switch can handle between 5 to 10 times the throughput of a packet switch. This additional throughput capability improves network delay performance by allowing faster transmission facilities both as access channels and as trunks between network switches [23]. Frame relaying allows the use of 1.5 to 2 Mbps facilities at their full bandwidth which reduces network transit times to the order of a few milliseconds - making propagation effects the limiting factor in delay.

Chapter 3

Frame Relay Technology

3.1 Introduction

Frame relay is an adaptation of packet-switching technology developed for ISDN [20]. The growing requirements of data users, and mainly those resulting from LAN-to-LAN communications, coupled with two environmental changes have led to the evolution of frame relay technology.

Firstly, devices that communicate via the network (Data Terminal Equipment) are now more intelligent and often connected to LANs with built-in error recovery to take account of errors occurring on the LAN [19].

Secondly, line quality has improved significantly as digital transmission facilities have replaced analogue lines. The growing incidence of fibre reduces the transmission errors still further to a negligible level.

As the high quality and speed of the modern transmission links reduces the need for error control on a per-link basis [24], frames are simply forwarded through the frame relay network without error correction. Errors or congestion problems will result in frames being discarded and recovery is the responsibility of the end user systems. By taking responsibility for some functions away from the network, the bandwidth and processing overheads in frame relay networks is less than that in conventional X.25 networks, and a much faster packet-switching facility is provided. However, the reduced functionality at the network nodes makes these networks very vulnerable to congestion. With the present scheme of fixed window flow control, operating on an end-to-end basis, the loss of packets may result in the retransmission of up to a window worth of packets on each virtual circuit. This would further aggravate congestion, hence causing the network performance to deteriorate. The aim of this chapter is to provide an overview of frame relay networks and to discuss flow and congestion control techniques to improve the performance of these networks under overload conditions.

3.2 Frame Relay Protocol

Frame relay technology can be used for access to the ISDN, or as an independent technical solution. Since frame relay has evolved as a part of the ISDN portfolio, the protocol complies with the ISDN protocol reference model. All call control is performed out of band using the techniques of circuit switching for call setup, operation and maintenance.

The packets are switched through the network using only the layer 2 frame level. Frame relaying operates entirely within the data link layer and uses the multiplexing possibilities of LAPD [22]. Different user data streams can therefore be multiplexed at the data link layer within the same access channel.

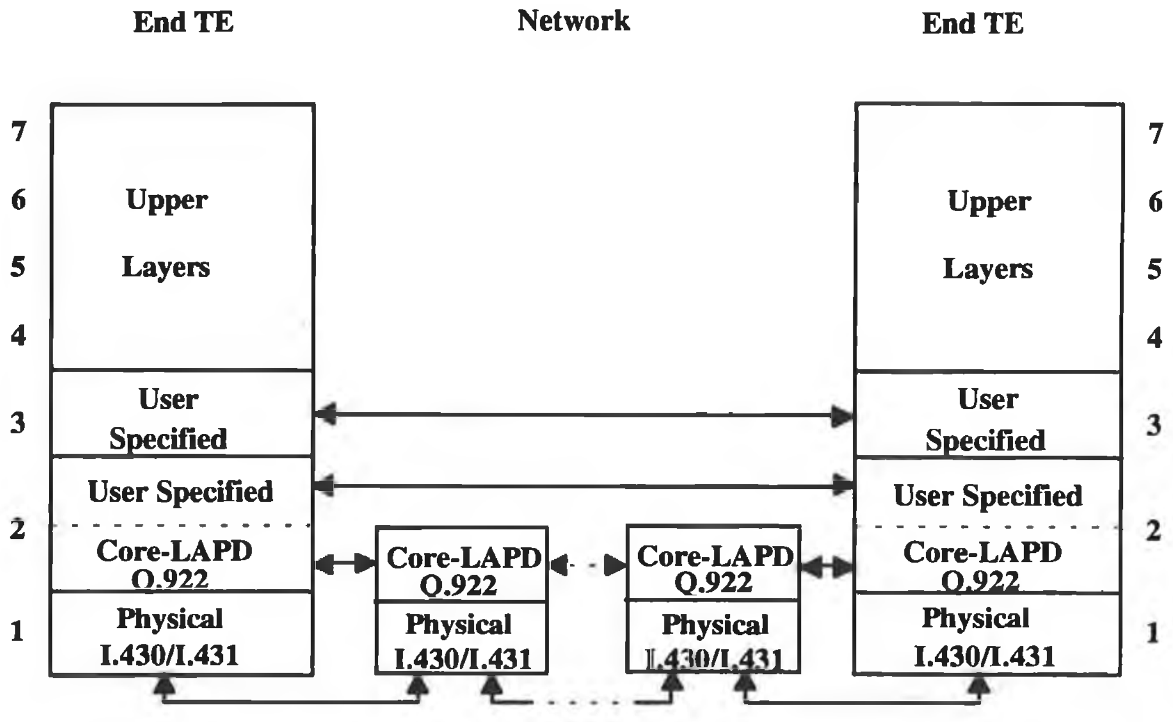


Fig.3.1 **The Frame Relay Protocol Stack**

In a frame relay network, only the core procedures of the LAPD protocol are implemented at the network nodes. These core functions for data transfer are specified by CCITT (Recommendation Q.922) and are also defined in Chapter 2. With these core functions as a basis, a frame-relaying network provides the following attributes in the user-plane:

- full duplex transfer of frames
- preservation of the order of frames from one user-network interface (UNI) to another
- non-duplication of frames
- detection of transmission, format and operation errors
- transport of the user data contents of a frame transparently; only the frame address and FCS fields may be modified
- no acknowledgement (to the user) of frames within the network
- signalling of the start of congestion

However, as the LAPD protocol is not terminated at the network nodes, procedural functions such as error recovery and flow control are left to the user and are performed on an end-to-end basis.

The reduced functionality at the network nodes enables frame relay networks to assemble, route and disassemble packets much faster than X.25 networks can [20]. The highly streamlined protocol processing also makes the network operation more efficient, thereby improving the throughput and the delay performance [9].

3.3 The Frame Relaying Network

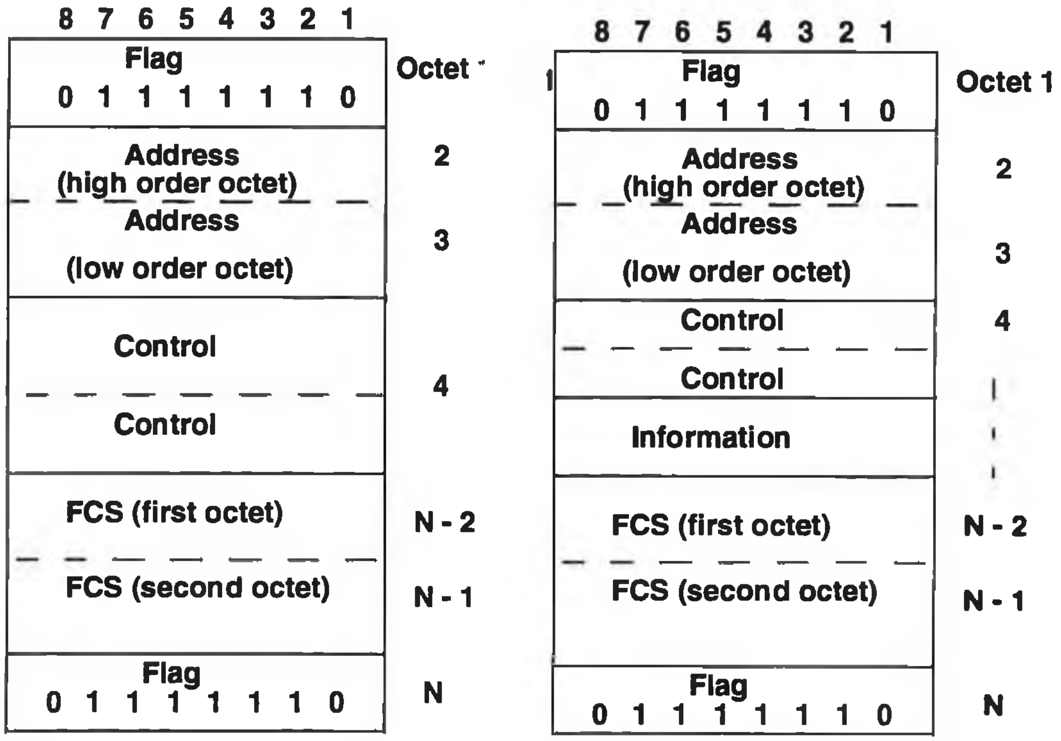
A Frame Relay (FR) network is made up of network nodes and user equipments (DTEs) connected to the network. All the network nodes are FR nodes with only the end systems terminating full LAPD. The DTE, e.g., a personal computer, a gateway, a route or a host computer, is provided with the interface defined for Frame Relay [15]. A Frame Relay interface acts like a local area network in which data may be sent to any other device in the network by simply specifying the destination's address. Because the frame relay interface recognises the frames that are to be transmitted, the rate of transfer, theoretically, is under the control of the sender. The backbone network to which a frame relay interface is connected, must provide bandwidth on demand. Only then can a user realise the full benefits of frame relay.

The sending DTE transmits frames to the network. The structure of these frames is described in the next section. Each of these frames contains an identification code (Data Link Connection Identifier, DLCI), to distinguish the different user data streams within the same access channel. The label DLCI is assigned when a frame relaying call is being established. This allows the network to route each frame on a hop-by-hop basis along a virtual path defined either at call setup time or at subscription time (in case of Permanent Logical Link Connections, PLLC) [22].

The use of several identification codes permits several parallel sessions in different directions to coexist on one physical connection. In this way, a DTE can communicate simultaneously with different destinations over the same physical connection to the network. It is therefore possible for a frame relaying network to provide efficient communications between various stations on different LANs (such as ethernet, token ring, FDDI).

3.4 Frame Structure in the User Plane

All data link layer peer-to-peer exchanges in a frame relay network are frames conforming CCITT recommendation Q.922. The structure of these frames, shown in figure 3.2 is based on the LAPD protocol. Format A is for frames containing no information field and format B is for frames containing information fields [6].



Format A **Format B**

Fig. 3.2 **Frame Relay Frame Format**

The fields are formatted and used according to the following conventions:

Flags: All frames start and end with the flag sequence consisting of one "0" bit followed by six contiguous "1" bits and one "0" bit. Flags are used for delimiting frames. In order to ensure that no bit patterns in the payload portion of the frame inadvertently match the "01111110" pattern of the delimiting flags, frame relay performs "zero bit insertion". When the frame relay access device is transmitting data that is between the opening and closing flags of a frame, it will insert a "0" bit after all contiguous "1" bits. These extra "0" bits are removed at the receiving end of the connection by zero bit extraction.

Address Field: The address field consists of two octets. It identifies the intended receiver of a command frame and the transmitter of a response frame. The address field format contains the address field extension bits (EA), a command/response indication bit (C/R), a Data Link Connection Identifier (DLCI) sub-field, and explicit network congestion notification bits-Forward Error Congestion Notification (FECN), Backward Error Congestion Notification (BECN) and Discard Eligibility (DE) bit.

Extended Address: The EA bit is the low order bit (bit 1) of the octets comprising the frame-relay address field. In current implementations, only two octet address fields are used. In this case, the EA bit in the first octet of the address field is set to 0 and the EA bit in the second octet of the address field is set to 1.

Address field format

8	7	6	5	4	3	2	1
DLCI (high order)						**C / R** **0/1**	**EA** **0**
DLCI (low order)				**FECN**	**BECN**	**DE**	**EA** **1**

Fig. 3.3 **Address Field Format**

Command/Response: The C/R Indication bit identifies a frame as either a command or a response. The user side shall send commands with the C/R bit set to zero, and responses with the C/R bit set to one. The network side shall do the opposite, i.e., commands are sent with C/R set to 1, and responses are sent with C/R set to 0.

Command/Response	Direction	C/R value
Command	Network side -> user side	1
	User side -> network side	0
Response	Network side -> user side	0
	User side -> network side	1

Fig. 3.4 **C/R field bit usage**

DLCI: The Data Link Connection Identifier is the addressing mechanism of frame relay. The DLCI consists of the six most significant bits of the second octet plus the four most significant bits of the third octet of the frame relay frame. Additional bits, dependent upon the value of the extended address (EA) bit, may be used to form a complete DLCI.

FECN and BECN: The Backward Explicit Congestion Notification (BECN) bit, if implemented, may be set by a congested network to notify a frame relay access device that congestion avoidance procedures should be initiated. As a frame-relay frame travels through a network it may encounter congestion. It is the responsibility of the network to recognise this condition and set the Forward Explicit Congestion Notification (FECN) bit.

Traffic	No Congestion		Congestion A - > B and B -> A	
	FECN	BECN	FECN	BECN
A -> B	0	0	1	1
B -> A	0	0	1	1
Traffic	Congestion A -> B No Congestion; B -> A		Congestion B -> A No Congestion; A -> B	
	FECN	BECN	FECN	BECN
A -> B	1	0	0	1
B -> A	0	1	1	0

Fig. 3.5 **State Table for FECN and BECN bits**

When the receiving frame-relay access device finds that the FECN bit is set for a given Permanent Virtual Connection, it must then set the BECN bit for frames being transmitted back to the source of the frame containing a FECN bit.

Discard Eligibility: The DE bit is used to indicate a frames suitability for discard in network congestion conditions. In theory, the frame-relay network would discard frames with a DE bit set to 1 instead of frames with a DE bit set to 0 when the network became congested and could no longer carry all the data being presented to it.

For instance, a frame that contains a LAN server broadcast message could have its DE bit set to one by a router as the frame was being built for transmission. If the LAN server broadcast message only reaches its destination some of the time, this would be acceptable because of the nature of LAN server broadcast messages.

When a Permanent Virtual Connection exceeds its Committed Burst Size for more than the Committed Rate Measurement Interval, the DE bit might be set.

Control Field Formats: The control field identifies the type of frame, which will be either a command or a response. The control field will contain sequence numbers where applicable. Two types of control field formats are specified; numbered information transfer (I-format) and supervisory functions (S-format).

Control field formats

Control field bits (modulo 128)	8	7	6	5	4	3	2	
I format	N(S)							0
	N(R)							P
S format	X	X	X	X	S	S	0	1
	N(R)							P/F

N(S) Transmitter send sequence number

N(S) Transmitter send sequence number

S Supervisory function bit

P / F Poll bit when issued as a command
Final bit when issued as a response

X Reserved and set to 0

Fig. 3.6 **Control Field Formats**

The *I-format* is used to perform information transfer between layer 3 entities. The send sequence number N(s) is set equal to the current value of the send state variable V(s), which denotes the sequence number of the next in-sequence I-frame to be transmitted. The receive sequence number is set equal to the current value of the receive state variable V(r), which denotes the sequence number of the next expected in-sequence I-frame to be received. The Poll/Final (P/F) bit serves a function in both command and response frames. In command frames, the P/F bit is referred to as the P bit, and in response frames it is referred to as the F bit. The P bit set equal to "1" is used by a data link layer entity to poll a response frame from the peer data link layer entity. The F bit set to "1" is used by a data link layer entity to indicate the response frame transmitted as a result of a poll command.

The length of the information field in the LAPD frame is adjustable to a maximum value defined for the service concerned. The length is normally set to a value at which the information from the application - a TCP/IP packet, an SDLC frame, an X.25 packet can be carried without having to split [15].

The *S-format* is used to perform data link supervisory control functions such as acknowledge I frames, request retransmission of I-frames, and request temporary suspension of I-frame transmission. The two S bits are used to identify the type of command/response, i.e. Receiver Ready (RR), Receiver Not Ready (RNR), and Reject (REJ).

Frame Check Sequence: FCS is used to verify that a frame is not corrupted during transmission. The FCS is a two-octet field that follows the user-data field and precedes the closing flag. The FCS is the result of applying the CCITT Cyclic Redundancy Checking (CRC) polynomial to the frame-relay frame form, and including, the first bit of the address field to the last bit of the user data field, excluding bits inserted for transparency. If the FCS calculated by the source frame relay access device and that recalculated by the destination frame relay access device do not match, then the frame is discarded.

As opposed to X.25, where an erroneous FCS triggers a retransmission procedure over the link, the network cannot guarantee that all frames will reach their destination [15]. Frames indicated as erroneous by the FCS are discarded, but this does not necessarily mean that the user loses any data. Checking to ensure that the DTE-to-DTE transmission has been error free is usually a mandatory function in the higher-level protocol used for information transfer in the application concerned. If an error occurs, data must be retransmitted through the entire network, and this obviously takes longer than retransmission only over the link on which the error has occurred. An approximate limit where Frame Relay is effective is a Bit Error Ratio (BER) of no more than 1/1,000,000 on individual links. This means that modern, digital transport networks are well-suited for the simplified protocol, but if the chain contains an analogue link, a short delay time for each link must be weighed against the expected frequency of end-to-end retransmissions.

3.5 Flow and Congestion control in Frame Relay Networks

Since LAPD protocol is not terminated at the switches, flow control is implemented on an end-to-end basis in the frame relay network. The **send sequence number N(s)** and the **receive sequence number N(r)** are used for flow control in frame relay networks. In addition, these networks commonly use the **window rotation algorithm** to regulate the flow of traffic between the transmitter and the receiver.

3.5.1 Fixed Window Flow Control

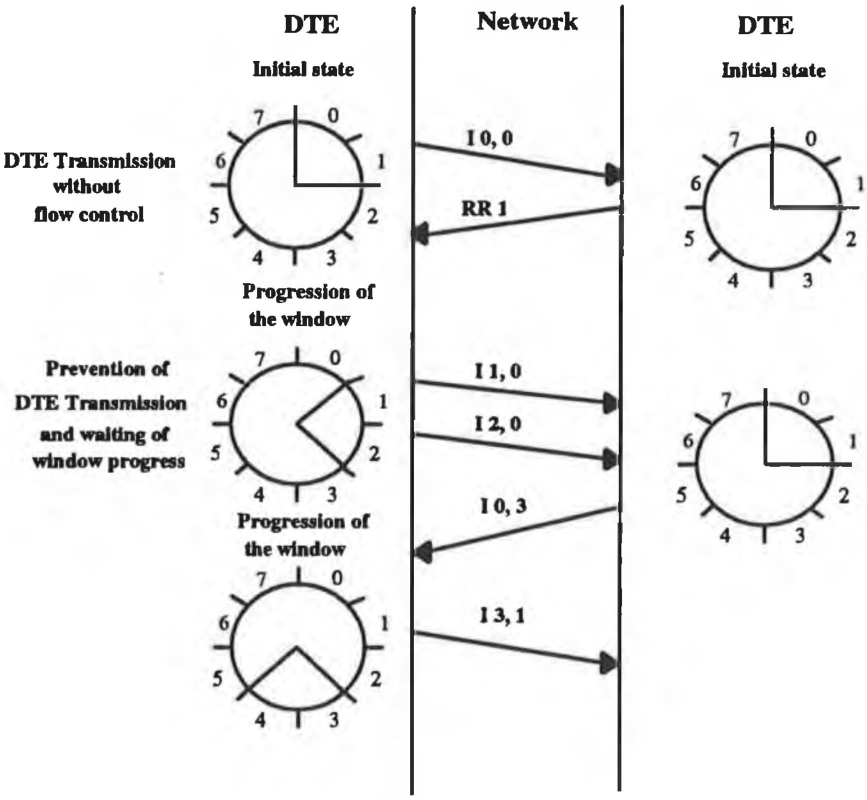
According to the **window rotation algorithm**, a sender is permitted to transmit a fixed number W (the window size) I-frames without having to wait for an acknowledgement. During the call setup phase of each virtual circuit, a window of size W is negotiated. This window W is then maintained on each logical channel during the data transfer phase. In addition to the window, the N(s) and N(r) are also used for flow control.

The **send sequence number N(s)** is used for the sequential numbering of each data packet transmitted on a given logical channel. The numbering is performed modulo 128 and the sequence numbers cycle through the entire range, 0 through 127.

The receiving DTE controls the flow of information by sending transmission authorisations to the network, which are then passed on to the sending DTE. Transmission authorisation is achieved by using a **receive sequence number N(r)**, which is contained in all I frames and supervisory frames. The numbering of the N(r) is also performed modulo 128. The N(r) is an indication of the expected send sequence number of the next received I-frame. After setting up or resetting the virtual circuit N(r) is transmitted as zero. The value of N(r) remains at zero until a valid, in-sequence data packet is received. It is then incremented and is used to authorise the sending DTE to transmit a further packet. The DTE can transmit a N(r) value only if it is atleast equal to the last N(r) transmitted and does not exceed the last N(s) received plus one.

The first packet transmitted for each direction of transmission, after setting up or resetting the virtual circuit, has 0 as its send sequence number. After setting up or resetting the virtual circuit, the transmitter can continue to transmit up to W(window size) packets without waiting for a transmission authorisation. However, after sending the W number of packets, the transmitter has to wait for a valid N(r). The transmitter is authorised to transmit only packets whose N(s) is such that N(s) = (last N(s) transmitted + 1) or N(s) < (last N(r) received + W). Flow control is thus maintained by the receiver on each virtual circuit, and is used to prevent a receiver from being overwhelmed with data that it cannot handle.

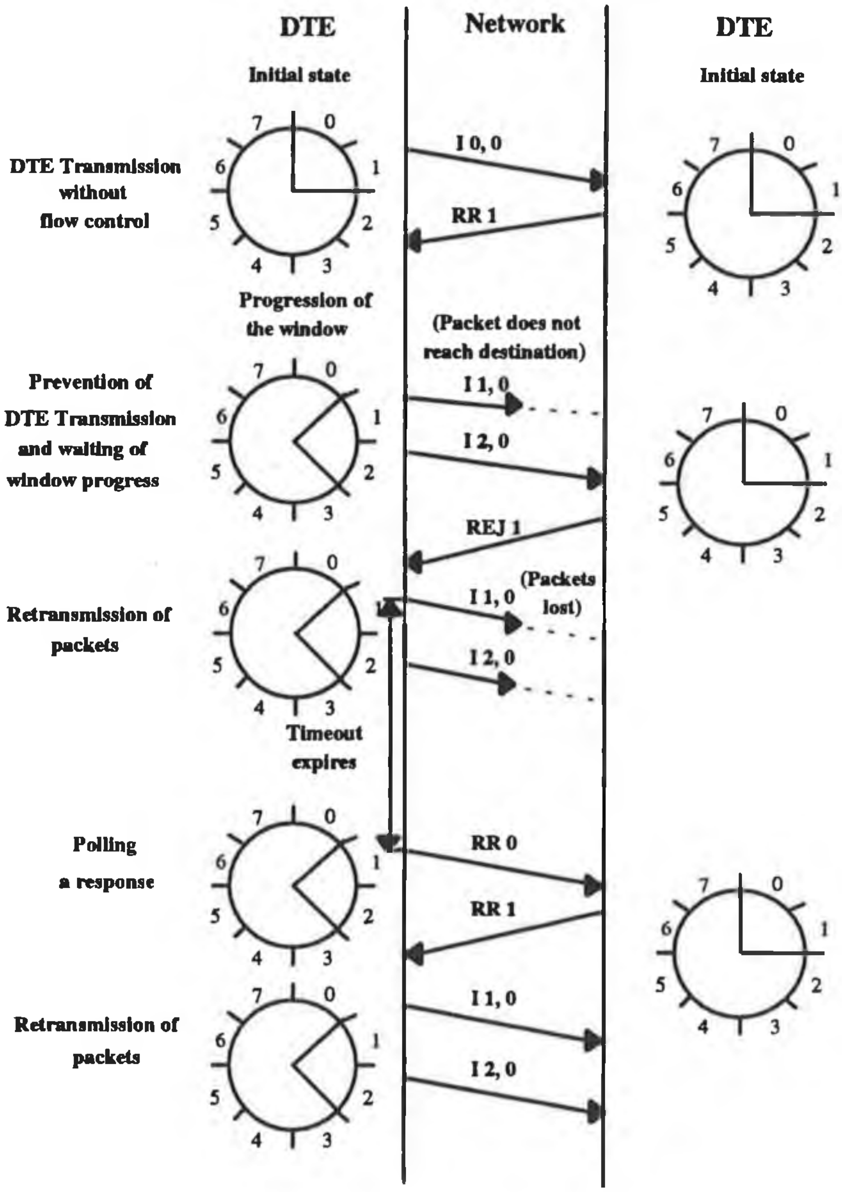
Figure 3.7 gives a description of the window. Each packet is represented by a portion of a cycle, in which the N(s) number of the packet is indicated. It is assumed that the DTE has chosen W = 2 and that the numbering of the sequence numbers is done modulo 8. Flow control is thus ensured independently on each virtual circuit and for each direction of transmission.


DTE
Network
DTE
Initial state
Initial state
DTE Transmission without flow control
Progression of the window
Prevention of DTE Transmission and waiting of window progress
Progression of the window
I 0, 0
RR 1
I 1, 0
I 2, 0
I 0, 3
I 3, 1
7
0
1
2
3
4
5
6

Legend:
I x, y - Information frame, N(s), N(r)
RR y - Receiver Ready, N(r)

Fig. 3.7 **Example of flow control**

This **window flow control** mechanism provides **congestion control** as well [21]. In this case, the objective of this mechanism is not to provide speed-matching, but to prevent resources (buffers and transmission links) along a virtual-circuit path from being congested. In cases of congestion, there may be buffer overflow at the intermediate switches leading to packet loss. To recover from packet-loss or from error condition, the windowing protocol is used in conjunction with **Go-back-N** ARQ(automatic repeat request) procedure.



DTE
Network
DTE
Initial state
Initial state
DTE Transmission without flow control
I 0, 0
RR 1
Progression of the window
Prevention of DTE Transmission and waiting of window progress
(Packet does not reach destination)
I 1, 0
I 2, 0
REJ 1
Retransmission of packets
I 1, 0
(Packets lost)
I 2, 0
Timeout expires
Polling a response
RR 0
RR 1
Retransmission of packets
I 1, 0
I 2, 0
0
1
2
3
4
5
6
7

Fig. 3.8 **Error-recovery with fixed window algorithm**

According to the **Go-back-N** procedure, the transmitter can send up to W (window) number of packets without waiting for an acknowledgement. If the transmitter receives a negative acknowledgement, as specified by the REJ frame, it retransmits the frame whose N(s) equals the N(r) in the REJ frame, and all frames following it. If the transmitter does not receive an acknowledgement within a specified interval timeout after transmission, it polls the receiver for a response. If the N(r) of the response is less than the N(s) of the next I-frame to be transmitted, it retransmits the I-frame whose N(s) equals the N(r) in the REJ frame and all the frames following it.

Although the fixed window flow control can help in controlling congestion to some extent, it is not very effective in frame relay networks. This is because the LAPD protocol is not terminated at the network nodes [12] and the flow and congestion control is carried out on an end-to-end basis. Since the window rotation and error induced retransmissions are performed only at the end systems terminating LAPD, a transmission error, or a loss of packet on any of the links in the frame relay connection causes retransmissions over all the links [7]. If congestion develops at any network component (that is, if a number of virtual circuits are active simultaneously and the trunk buffers are not large enough to hold their entire windows), frames can be dropped, resulting in the retransmission of up to a window's worth of data on each virtual circuit. This will cause the end-to-end delay to increase, and although the end-to-end window flow control will slow input traffic somewhat, it may not be sufficient to alleviate congestion. Instead, the retransmission traffic may further aggravate congestion. In addition, the absence of layer 3 protocol and the layer 2 procedural functions at the network nodes, prevents the network from sending signals to the DTE to throttle the traffic.

3.5.2 Congestion Problems

It is evident that the lack of flow and congestion control in the frame relay network can result in performance degradation in the data transfer of a frame-relaying network. This condition usually occurs when the traffic exceeds the critical capacity of the network or when one of the equipment fails [22].

The performance of the network is degraded in terms of throughput, frame loss, and frame delay. In addition, facility failures can lead to session disconnects unless responsive failure detection and recovery capabilities are provided in the network.

The main cause of throughput degradation in a packet network is the wastage of resources. This may happen because a user acquires more resources than strictly needed, thus starving other users (e.g. a slow sink fed by a fast source may create backlog of packets within the network which prevents other packets from getting through). The two resources that are most commonly wasted in a packet network are communications capacity and storage capacity.

Buffer wastage is an indirect consequence of limited nodal storage: a given end-to-end packet stream may be blocked at an intermediate node along the path because all of the buffers have been hogged by other streams. This may happen even if the channel bandwidth is plentiful along the blocked stream path, thus causing an unnecessary loss of throughput. The source of this throughput degradation is that some users unnecessarily monopolise the buffers at some congested node.

There can be a degradation in throughput if the offered load increases beyond the critical system capacity and if there is an inefficient allocation (and therefore wastage) of buffers. Since it is economically prohibitive to provide adequate number of buffers in the network, losses can occur. The retransmissions of frames discarded due to buffer overflow cause the buffer utilisation to increase, which in turn further increases the overflow probability. This positive feedback causes abrupt increases in the buffer utilisation, overflow probability, and the fraction of the network capacity used for retransmissions. As a result there is a sudden drop in throughput.

In addition, some users, because of their relative position in the network or the particular selection of network and traffic parameters, may succeed in capturing a larger share of resources than others, and thus enjoy a preferential treatment. This uncontrolled competition results in the unfair sharing of network resources.

To minimise the impact of network congestion and failure on end-user performance and to ensure that the network operates at peak efficiencies at all times, effective control strategies are needed. In addition, these strategies should operate without incurring excessive processing overheads.

3.5.3 Congestion Controls

To minimise the impact of network congestion and failure on end-user performance and to ensure that the network operates at peak efficiencies at all times, effective control strategies are needed that can operate without incurring excessive overheads.

Congestion controls deal with the control of traffic flows upon the occurrence of network congestion [9], without which the network's useful throughput would greatly decrease while the network transit delays would grow unacceptably large. All users would suffer service degradation even if the congestion is caused by a few "overactive" users. The objectives of a congestion-control strategy are to ensure that the network throughput is maintained at maximal levels, to ensure fairness by preventing a small number of users from hogging network resources and to minimise the adverse effects of congestion on user-perceived performance.

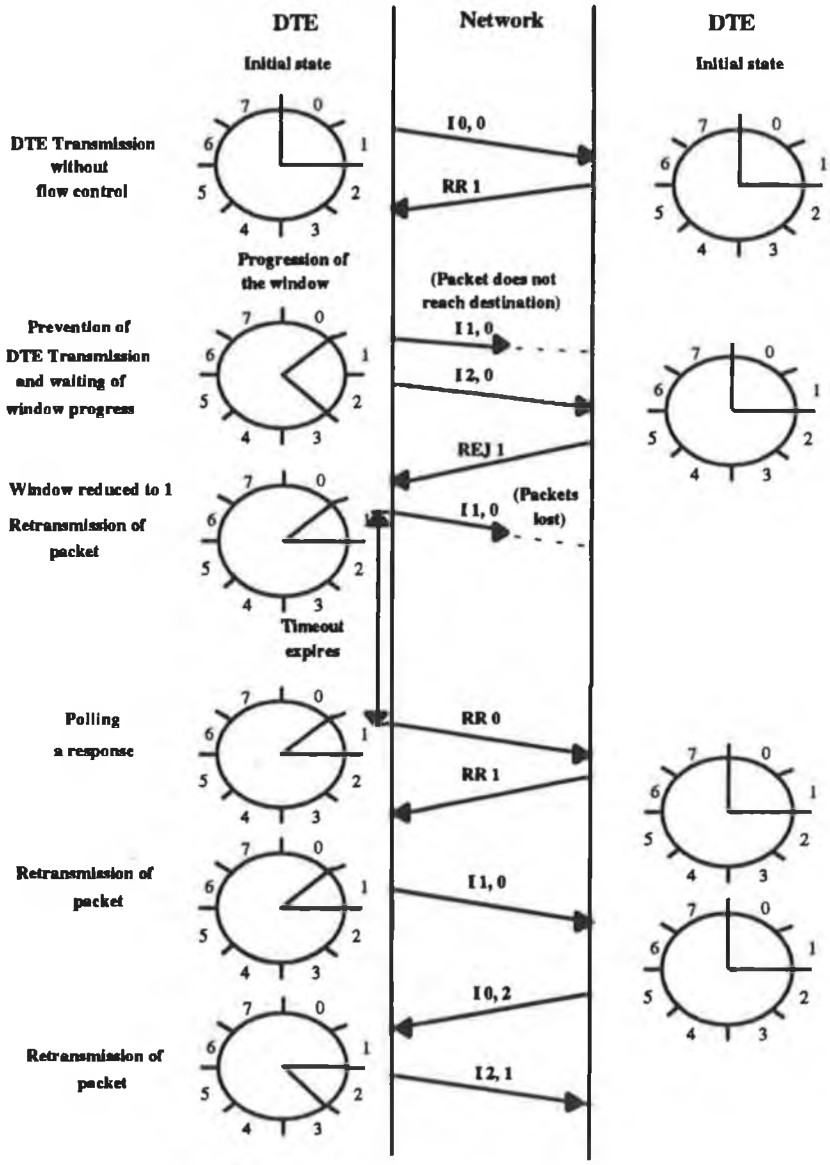
These objectives can be attained by implementing congestion avoidance mechanisms, (CCITT Recommendation I.370) which should react fast, have the adequate granularity, be adaptive and tailored to the circumstances, and allow the users out of forced idleness as soon as the congestion disappears [22]. These congestion control mechanisms belong to the user plane and can be provided partially by the user and partially by the network. The mechanisms have to be developed such that the variance in quality of service is equally spread over the users.

Some of the congestion control mechanisms are described here:
Buffer management strategy: The proper dimensioning of resources is the best protection against congestion. Congestion can be avoided by allocating a window's worth of buffers for each virtual circuit on a trunk. However, as this strategy may be economically prohibitive, the data transmission networks can not always be configured for the worst case conditions.

Selective frame discarding: When a frame is dropped due to congestion, the following frame on the same virtual circuit is received out of sequence at the end point, resulting in a reject and retransmission of a up to a whole window worth of frames. Therefore, a strategy that identifies frames received on a virtual circuit immediately after a frame is dropped, and continues to drop them if the congestion persists, will help in localising the performance degradation to as few virtual circuits as possible.

Adaptive windowing: The end points reduce their window sizes to W_{min} in response to out-of-sequence deliveries, and gradually increase them after successful transmissions. By reducing their window sizes whenever there is an indication of a possible congestion, we achieve a high responsiveness to the flow control mechanism in the sense that an immediate and very effective throttling of the input traffic is performed [4]. Subsequent to reduction, the endpoints again attempt to increase their window sizes. This process is tightly coupled to the reception of acknowledgements. In this way, we achieve control of the speed by which the window size is increased, by the momentarily ability of the network to transport frames successfully.

Congestion Notification: In cases of overload, the network nodes can notify the end-user to take some appropriate action. A network node can register a tendency towards overload when the number of frames queued for access to outgoing lines exceeds a preset limit. When the node registers a tendency towards overload, the network requests the DTE to reduce its data flow by sending a Forward/Backward Explicit Congestion Notification (FECN/BECN) consisting of the check bits of the frame header.



Legend:
I x, y- Information frame, N(s), N(r)
RR y- Receiver Ready frame, N(r)
REJ y- Reject, N(r)

Assumptions:
$W_{max} = 2$
$W_{min} = 1$
Sequence numbering is modulo 8

Fig. 3.9 **Example of Adaptive windowing**

In the case of destination controlled transmitters, the FECN bit is set in the core aspects protocol; and in the case of source controlled transmitters, the BECN bit is set in the core aspects protocol in frames transported in the reverse direction. The network takes no action but leaves it to the DTE to decide on the best way of controlling its data flow. All networks must transport the FECN and the BECN indications, either unmodified or, if in congested condition, with the appropriate indication set.

However if overload occurs, the network will discard frames, and recovery will take place on an end-to-end basis.

When end users receive implicit or explicit indication of network congestion, they should reduce their load. However, even if the terminals are not able to act on the information, they should have the capability to receive the explicit congestion notification generated by the network. By reducing the window size to W_{min} on receiving an Backward Explicit Congestion Notification bit set, users can reduce their information transfer rate and this may result in an increase in the effective throughput available to the end user during congestion.

Stop Duration: The end users can also reduce their load by stopping the transmission of packets for short periods of duration. If the load offered to the network is itself reduced, the network can quickly recover from the state of congestion.

3.6 Advantages over Existing Networks

Because of the simplified protocol in the frame relay network, the frame handling capacity is much higher than that of X.25 networks. It is estimated that FR switches can handle up to 10 times the throughput of existing X.25 networks [18]. This additional throughput capability improves network delay performance by allowing faster transmission facilities both as access channels and as trunks between network switches. Frame relaying allows the use of ISDN primary rate user-network interfaces at their full bandwidth, thereby reducing the latency of the network.

The data link layer protocol in frame relay networks is LAPD. Calls can therefore be established in one step, and there is clear separation between signalling and data transfer. In addition, this allows the frame relay networks to disassociate the physical topology from the logical topology of interconnections between devices which, with frame relay, are provided by virtual circuits [19]. Frame relaying allows for the multiplexing of these virtual circuits over one physical connection [5]. It is therefore possible to interconnect N sites with N physical links only. Earlier, a full-mesh interconnected network, with N(N-1)/2 paths, was required to interconnect LANs.

This was required to guarantee low latency and to assure alternate routes in case of failures. For large interconnections, these models were cost-ineffective. Frame relay networks are cost-effective for interconnecting LANs. In addition, they offer resilience and alternate routing at lower costs, reduce network latency and provide increased reliability.

3.7 Implementation of Frame Relay

Frame Relay can be implemented in different ways [9]:

* As part of a private switching network that also supports other types of communication, such as X.25 and SNA.

* As a virtual private network consisting of an operator network section dedicated to use by a specific customer, company, organisation.

* As a public Frame Relay service provided by the PTT or some other operator.

* As a hybrid solution: a private network covering the primary geographical area of a company, and peripheral units via public Frame Relay services.

3.8 Conclusions

Although frame relay is a standardised interface that provides multiplexed access to bandwidth-on-demand backbone networks and delivers LAN like performance over a wide area, frame relay networks are vulnerable to congestion. It is therefore essential to explore flow and congestion control strategies to prevent the degradation of the network performance. The use of congestion notification bits to initiate congestion avoidance procedures have been specified in the frame relay standards. However, some of the present frame relay compatible terminal equipments do not support these functions. In this study, this technique and many others will be implemented and the performance of the frame relay networks will then be compared.

Chapter 4

Network Modelling and Simulation

4.1 Introduction

To evaluate the performance of frame relay networks, a number of simulation models were formulated and implemented. The simulation models were developed using the network simulation Computer Aided Design (CAD) package, OPNET. OPNET runs on the SUN SPARC and HP APOLLO workstations, in a UNIX environment. The source code was developed using OPNET simulation kernel routines, with ANSI C as the base language.

A number of simulation experiments were designed for this study. OPNET was also used to verify the validity of the developed model. As the simulation model was entirely in software, it was easy to alter the simulation program and explore different design alternatives. In addition, statistically significant data were gathered. The simulation data was obtained by executing the simulation experiments for long times. Data points were also collected for different parameter values. The data collected was then analysed.

4.2 OPNET - a brief overview

OPNET, or Optimised Network Engineering Tools, is a hierarchical object oriented simulation system that can be used for the simulation of various communication networks such as ISDN architecture, local area networks and mobile packet radio networks. As it is designed specifically for development and analysis of communication networks, it provides extensive detail not available in simpler resource-based simulation packages. OPNET simulations are based on four separate modelling domains called network, node, process and link. The dependencies between these modelling domains are shown in fig. As the figure illustrates, network models rely on the definition of the node models which in turn incorporate process models. In addition, link models are used to characterise links within the network domain.

In the **Network Domain**, node models are instantiated and each instance may be assigned independent attributes including identification and position, and user-defined attributes. Nodes which are designed to attach to physical links may be interconnected to form arbitrary network topologies.

The **Link Domain** allows incorporation of custom or user-specific link models within OPNET simulations. These models are specified in C and are linked into the simulation. Point-to-point links are represented by lines between source and destination nodes. The point-to-point links are unidirectional, therefore a duplex link is represented by two links, one for each direction. The point-to-point links have a number of built-in attributes which can be specified by the user. They include transit delay incurred by packets/frames forwarded over the link, bit error rate which is the probability of bit errors in packets/frames transmitted over the link.

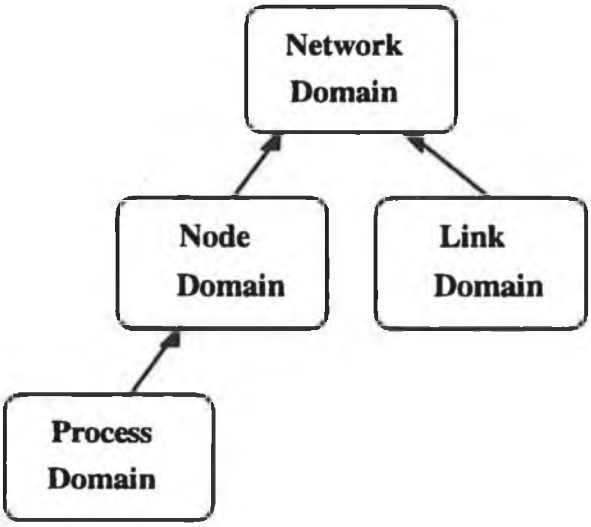


Fig. 4.1 **OPNET Modelling Domains**

In the **Node Domain**, the internal structure of the nodes is defined. The internal structure of the nodes consists of modules which can generate, process, store, receive and transmit packets and manage resources according to a user defined process. These modules can be interconnected to form arbitrary complex node architectures. The processor and queue modules execute process models specified as finite state machines (FSM). The generator module stochastically produces packets conforming to a user-specified format, and according to user-specified Probability Density Functions.

Transmitter and Receiver modules are the interface to the link level models which transfer packets between nodes.

Process Models are specified using a graphical editor which captures the structure of the process in the form of a finite state machine. The finite state machine (FSM) models a communications process by responding to changes in its inputs, modifying its state and producing new outputs. No loss of generality occurs when using the graphical FSM approach to represent the process model as the state-transition diagram can contain fully general C language code. Process models may make use of a library of kernel procedures which support access to packets, network variables, statistic collection, packet communication and other simulation services.

The two fundamental components of an FSM state are states and transitions. States can be used to represent the significant modes of the process and may have certain actions associated with them. An FSM implements these actions either on entering or on leaving the state. All states can be thought of as being decomposed into three phases of traversal by the FSM. The first phase is the enter executives which are always implemented upon arrival in the state. The second is a possible resting phase where the FSM returns from its invocation. And the third phase is the implementation of the exit executives.

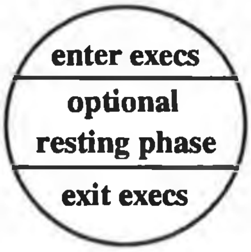


Fig. 4.2 **FSM Representation**

Two types of states are distinguished in implementing OPNET process models: forced and unforced states. These states are illustrated in fig. 4.3. Forced states bypass the second phase rather than return from the process model invocation. Unforced states, on the other hand, always cause the FSM to return from invocation and block immediately after implementation of the enter executives. An FSM will remain in the rest phase until a new interrupt is delivered to the process model, causing a new invocation. In fact, interrupts are always delivered to process models when their FSMs are in a blocked condition, and thus necessarily occupying an unforced state. The FSM will continue to execute until the rest phase is implemented by entering an unforced state. It is evident from fig. 4.4 that the FSM needs at least one unforced state to prevent it from looping continuously from forced state to forced state. If the process model were allowed to loop in this manner, the simulation kernel would be unable to invoke other process models and model the parallel operation of simulation entities. To differentiate between forced and unforced states, forced states are drawn black in hard copy output while unforced states are represented in white.

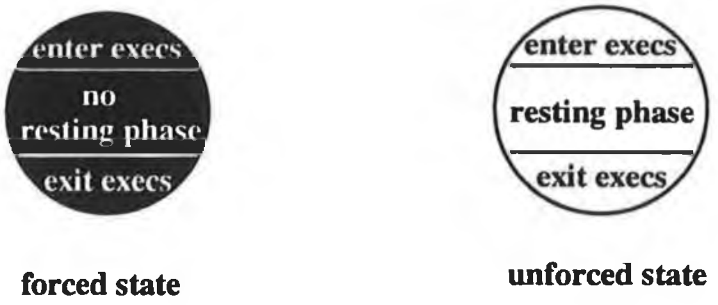


Fig. 4.3 **Forced/Unforced State Representation**

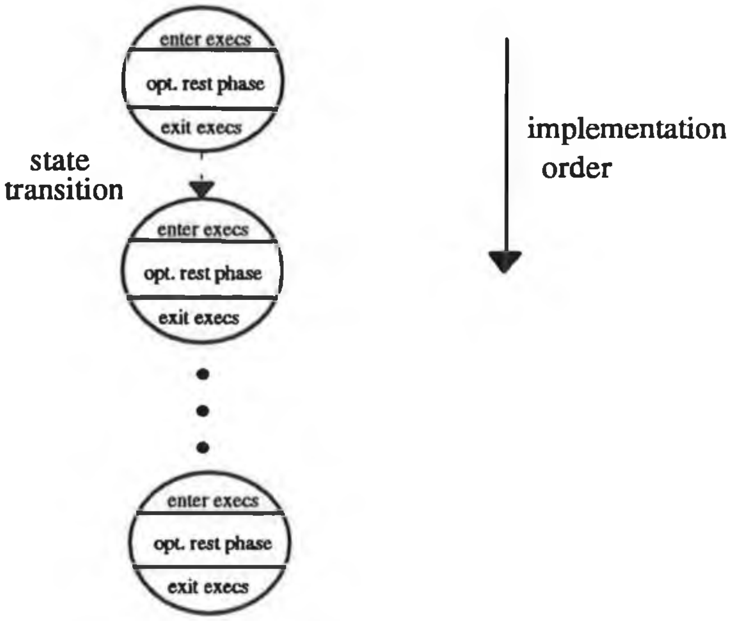


Fig. 4.4 **FSM Phase Traversal**

The transitions represent possible ways in which the process can migrate from one state to another. These expressions, evaluated as Booleans determine whether a particular transition will be followed and a new state entered. Since the finite state machine should occupy only one state at a time, only one transition statement should evaluate true at any one time.

4.3 Frame Relay Network

The study included the design and simulation of frame relay networks. A number of network topologies were studied in which users were connected to each other via frame relay network nodes. Subnetwork objects were used to represent logical and physical groupings of nodes and links within a larger network model. Each subnet consisted of a number of user-nodes and a frame relay node. These subnetworks were internetworked in turn via point-to-point links. Typically, virtual connections were assumed established between user-nodes of different subnets at the beginning of the simulation.

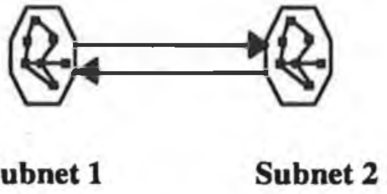


Fig. 4.5 **Frame Relay Network Configuration**

For most of the simulations, the simple network model shown in fig. 4.5 has been used. The network is composed of two internetworked subnets. Each subnet, as shown in fig.4.6, has four user-nodes and a frame relay switch. There is a one-to-one correspondence between the user nodes of each subnet.

The size of the buffer pools at each of the frame relay nodes was taken to be 20 kbytes for most of the simulations. The size of the buffer pools was also varied in different simulation experiments. This was done to highlight the effect of the buffer size on the performance of the system.

The bit error rates (BERs) on all the links were taken to be negligible. This is because the BER in modern digital transmission systems is of the order of 1E-6. However, in a simulation system, a BER of 1E-6 means that there would be one erroneous transmission in every 1,000,000 observed packets. This would mean running the simulation for a long enough time to be able to collect enough data points. However, memory constraints put an upper limit on the time for which the simulation could be run.

In addition, the channel capacity of each transmission link was taken to be 16 kbps (and not 64 kbps ~ which is the channel capacity of an ISDN B-channel) as the time to run the simulation model is also dependent on the size of the system. As OPNET creates large temporary files during simulation, it is not possible to run the simulation for these higher capacity channel values for a long time.

In the frame relay network, the procedural aspects of the LAPD protocol were implemented at the end-user nodes. The frame relay switches simply store and forward frames on the appropriate virtual circuits.

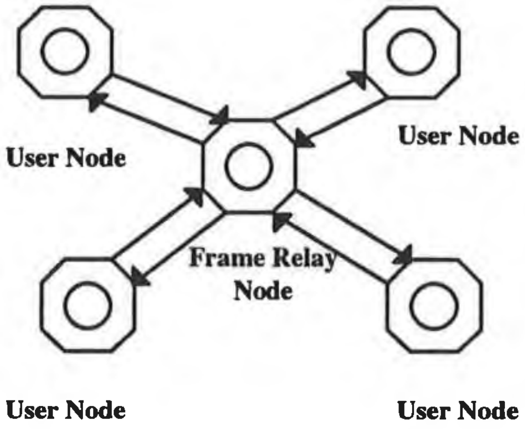


Fig. 4.6 **Subnetwork Representation**

4.4 User Node Model

The node model is used to specify the internal structure and the capability of a communication node. To analyse the steady state performance of the system, the node model shown in fig. 4.7 was used. The user node model consists of the modules *ideal generator, queue, transmitter and receiver,* connected together with packet streams.

The ideal generator module named as source in fig. 4.7 provides a convenient stochastic packet source. The frequency of packet arrivals and the length of packets can be controlled by probability distribution. As packets were assumed to be generated according to a Poisson process, the interarrival time between packets was taken to be exponentially distributed. The length of the packets created by the generator was taken to be constant. The packets created by the generator were made to abide by the LAPD format, specified in the Parameter Editor of the package. Packets produced by the generator arrive immediately (same simulated time) at the destination module connected to the generator's output and cause a stream interrupt there.

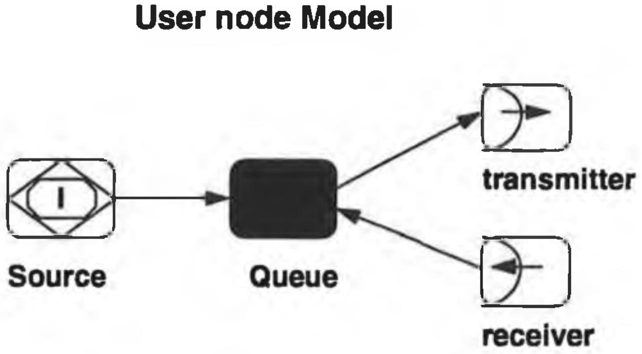


Fig.4.7 **User Node Model**

To analyse the dynamic performance of the system, the *Ideal Generator* module was replaced by a *Processor* module. This module executes a simple model for generation of traffic destined to users on other subnets. In addition, a range of burst durations and a range of data traffic rates was specified. This model was useful in simulating the stochastic data flow in computer communication networks.

The Queue module executes a process model which defines the communications process that the queue module is required to perform. The process model incorporates 'C' code and simulation kernel procedures to model processing functions of the node. In this way, the queue module's behaviour can be completely specified. The queue module contains a number of subqueues, each of which can hold a list of packets. The queuing discipline used and the number of subqueues needed in a particular queue module, and the capacity of each subqueue can also be specified. Subqueues are accessed by the process model using subqueue indices in the kernel procedures. Three subqueues were required for this model: one to queue packets arriving from the generator for transmission, one to hold window (W) number of I-frames for possible retransmission and one to queue control frames for later transmission (i.e. if the transmitter is busy at that moment of time).

The transmitter and receiver modules can be viewed as the interface between a node and a point-to-point transmission link. The maximum data rate for each of these modules can be specified. Since the LAPD protocol is assumed to be operating on an ISDN B-channel, the data rate specified should be 64 Kbps. However, due to memory constraints, the data rate specified in this work is 16 Kbps.

4.5 Frame Relay Node

The frame relay node model consists of a queue module which performs the various functions of the node. These functions are defined by the process model contained in the queue module.

The number of subqueues in the queue module were varied for different simulation runs. Different queuing strategies were deployed by simply modifying the software in the process model. The buffer size at the Frame Relay switch is varied for different simulation runs to see the effect of the buffer size on the performance of the frame relay networks.

There are also receiver modules and transmitter modules for the input and output streams. The maximum data rate specified for these modules was 16 kbps.

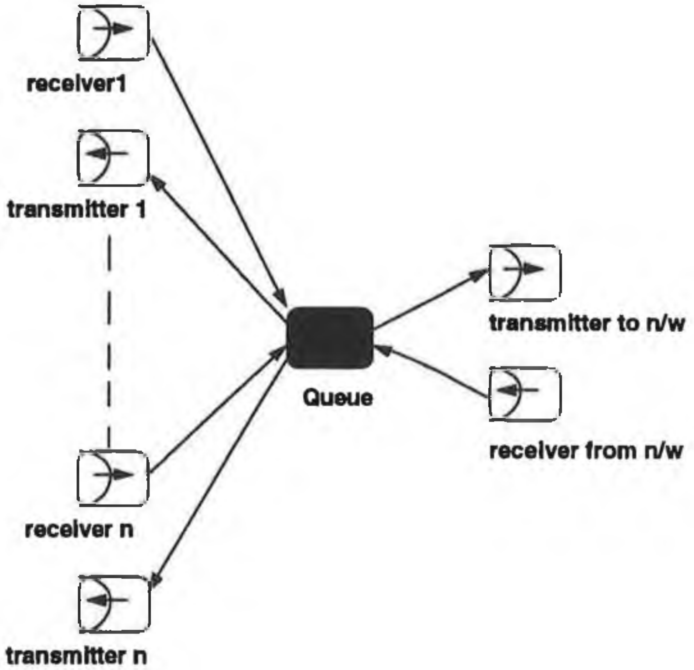


Fig.4.8 **Frame Relay Node Model**

4.6 Generator Process Model

The state transition diagram of the Generator process model is shown in fig. 4.9. In the init state, the process determines the range of packet generation rate. It then prepares a uniform probability distribution for the generation of data rates by future process Kernel procedure calls. In addition, the process also determines the range of burst durations; and prepares a uniform probability distribution for the generation of the same. The process then determines the duration of the first burst and schedules a self interrupt, with code BR for after the burst duration period. It also determines the rate at which packets are to be generated in this period. The time between the generation of corresponding packets is inversely equal to the data rate. The process schedules a self interrupt, with code AR, for after the interarrival period.

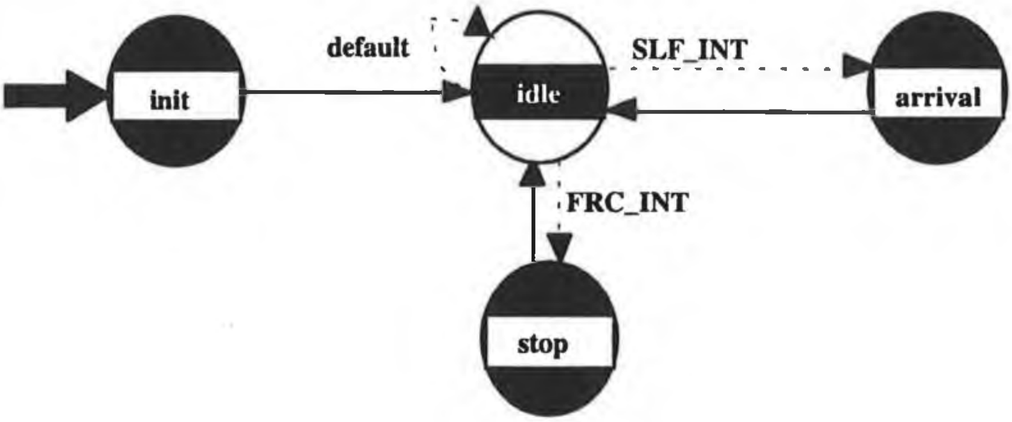


Fig. 4.9 **Generator Process Model**

The process then enters the idle state and stays there till one of the transitions evaluates true. When a self interrupt occurs, the transition *SLF_INT* evaluates true, and the process enters the arrival state. If the code of the self interrupt is AR, a packet of LAPD format is created and sent on the output stream. The process then determines the time of creation of the next packet and schedules a self interrupt for after this time again. If the code of the self interrupt is BR, the model determines the duration of the next burst, and the rate at which packets will be generated during this period. The interarrival times between the generation of packets is correspondingly altered. The process also schedules a self interrupt for after the new burst duration period.

The FRC_INT macro evaluates true when a remote interrupt occurs. This option was used in some of the models exerting backpressure on the generators. When the code of the invoking remote interrupt was 0, the process cancelled the pending self interrupts.

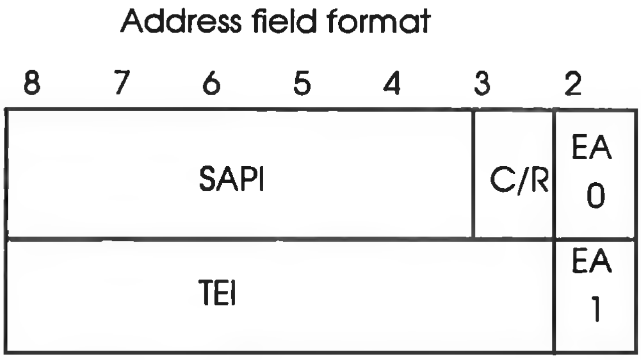
As a result, there was no further generation of packets. However, when there was a remote interrupt, with code 1, the process of generation of packets was initiated again. This was achieved, by scheduling self interrupts, with code BR and AR respectively, for the same simulation time.

4.7 User Process Model

The data phase of the LAPD communications protocol is modelled by the user process model. It is assumed that the virtual connection for data transfer is established at the beginning of the simulation.

The CCITT recommendation Q.922 specifies the inclusion of Forward/Backward Congestion Notification (FECN/BECN) information bits in the frame header. These bits can be used for flow control and are used by the network nodes to notify the DTE of congestion in the network. However, most of today's frame-relay-compatible DTEs cannot handle these signals.

Therefore, this study began with the simulation of frame relay networks which did not support these features. The frame format used for these simulations is also based on LAPD, but does not include the FECN/BECN bits for flow control. This frame format has been specified by the CCITT (Recommendation Q.921). In the LAPD standards, only the address field of the frames is formatted differently to the LAPD address field format. The address field is formatted as shown in fig. 4.10 and used according to the following convention:



EA = Address Field extension bit
C/R = Command / response field bit
SAPI = Service access point identifier
TEI = Terminal end point identifier

Fig.4.10 **LAPD Address Field Format**

Address field format:

The address field consists of two octets. It identifies the intended receiver of a command frame and the transmitter of a response frame. The address field format contains the address field extension bits (EA), a command/response (C/R) indication bit, a data link layer service access point identifier (SAPI) sub-field and a Terminal Endpoint Identifier (TEI) sub-field.

Extended Address: The EA bit is the low order bit (bit 1) of the octets comprising the frame-relay address field. In current implementations, only two octet address fields are used. In this case, the EA bit in the first octet of the address field is set to 0 and the EA bit in the second octet of the address field is set to 1.

Command/Response: The C/R Indication bit identifies a frame as either a command or a response. The user side shall send commands with the C/R bit set to zero, and responses with the C/R bit set to one. The network side shall do the opposite, i.e., commands are sent with C/R set to 1, and responses are sent with C/R set to 0.

SAPI: A Service Access Point (SAP) is a means of identifying a user of the services of a protocol entity. A Service Access Point Identifier (SAPI) identifies a point at which data link services are provided by a data link layer entity to a layer 3 or management entity.

TEI: A Terminal Endpoint Identifier (TEI) is used to identify a specific connection endpoint within a service access point as many terminal equipments may be using one protocol service.

The procedures for multiple frame acknowledged transfer in a frame relay network, using the address field format shown in fig. 4.10 are explained in detail. These procedures were modified for different windowing mechanisms, and for networks using the frame relay format. These modifications are explained in the process model section.

4.7.1 Procedures for Multiple Frame Acknowledged Information Transfer

At the beginning of the simulation, all exception conditions are cleared, the retransmission counter reset, the window size set to assigned value, and the send state variable V(s), the receive state variable V(r), and the acknowledge state variable V(a) are reset to zero. Here, V(s) denotes the sequence number of the next I-frame to be transmitted, V(r) denotes the sequence number of the next in-sequence I-frame expected to be received, and V(a) indicates the last frame that has been acknowledged by its peer [V(a)-1 equals the N(s) of the last acknowledged I frame].

Procedure for the use of the P/F bit:

Command received with P bit = 1	Response transmitted with F bit = 1
I, RR, REJ	RR, REJ

Fig. 4.11 **Immediate response operation of the P/F bit**

A data link layer entity receiving an RR, REJ or I frame, with the P bit set to 1, shall set the F bit to 1 in the next response frame it transmits.

Procedures for Information transfer in multiple frame operation: The procedures which apply to the transmission of I frames are defined here.

(1) Transmitting I frames: DLC transmits messages which are generated by the layers above it in the form of information frames (I-frames). The control field parameters N(s) and N(r) shall be assigned the values of V(s) and V(r), respectively. V(s) shall be incremented by 1 at the end of the transmission of the I-frame.

If timer T200 is not running at the time of the transmission of an I-frame, it shall be started. If timer T200 expires, then the procedures defined in *Waiting Acknowledgement* shall be followed.

If V(s) is equal to V(a) plus W (where W (window) is the maximum number of outstanding I-frames), the data link layer shall not transmit any new I-frames, but may retransmit an I-frame as a result of the recovery procedures.

(2) Receiving I-frames and sending acknowledgements: Independent of a timer recovery condition, when a data link layer entity receives a valid I-frame whose N(s) is equal to the current V(r), the data link layer entity shall pass the information field of this frame to layer 3 and increment its V(r) by 1. If no I-frame is available for transmission, the data link layer entity responds by transmitting an RR response with the F bit set to 0; or if an I-frame is available for transmission, the data link layer entity shall transmit the I-frame with the value of N(r) set to the current value of V(r).

However, when a valid I-frame is received which contains an N(s) value which is not equal to the V(r) at the receiver, the information field of the frame is discarded. The receiver shall not acknowledge (nor increment its V(r)) the I-frame causing the sequence error, nor any I frames which may follow, until an I frame with the correct N(s) is received.

The REJ frame is used by a receiving data link layer entity to initiate an exception condition recovery (retransmission) following the detection of an N(s) sequence error. Only one REJ exception condition for a given direction of information transfer shall be established at one time. An REJ exception condition is cleared when the requested I frame is received.

Receiving acknowledgements: On receipt of a valid I frame or a supervisory frame (RR, REJ), even in the timer recovery conditions, the data link layer entity shall treat the N(r) contained in this frame as an acknowledgement for all the I-frames it has transmitted with an N(s) up to and including the received N(r)-1. V(a) shall be set to N(r). The data link layer entity shall reset the timer T200 on receipt of a valid I-frame or supervisory frame with the N(r) higher than V(a) (actually acknowledging some I-frames), or an REJ frame with an N(r) equal to V(a).

However, if a supervisory frame with the P bit set to 1 has been transmitted and not acknowledged, timer T200 shall not be reset.

If timer T200 has been reset by the receipt of an I- or RR-frame, and if there are outstanding I-frames still unacknowledged, the data link layer entity shall restart timer T200. If timer T200 then expires, the data link layer entity shall follow the recovery procedure as defined in *Waiting Acknowledgements,* with respect to the unacknowledged I-frames.

If timer T200 has been reset by the receipt of an REJ frame, the data link layer entity shall follow the retransmission procedures defined in *Receiving REJ frames.*

Receiving REJ frames: On receipt of a REJ frame, the data link layer entity shall set its V(s) and V(a) to the value of the N(r) contained in the REJ frame control field, stop timer T200, and transmit the corresponding I-frame as soon as possible. If the data link layer entity is transmitting a supervisory frame at the time it receives a REJ frame, it shall complete that transmission before commencing transmission of the requested I-frame. All outstanding unacknowledged I-frames, commencing with the I-frame identified in the received REJ frame, shall be transmitted. Other I-frames not yet transmitted may be transmitted following the retransmitted I-frames.

Waiting acknowledgement: The data link layer entity shall maintain an internal retransmission count variable. If timer T200 expires, the data link layer entity shall enter the timer recovery condition and reset the retransmission count variable; or if it is already in the timer recovery condition, add one to its retransmission count variable. If the value of the retransmission count variable is less than N200, the data link layer entity shall restart timer T200 and transmit RR command with the P bit set to 1. However, if the value of the retransmission count variable is equal to N200, it shall indicate a re-establishment procedure. In this study, the simulation will be stopped as we are interested in studying the performance of the system only in the multiple frame established state.

The timer recovery condition is cleared when the data link layer entity receives a valid RR response with the F bit set to 1. If the N(R) of the received frame is within the range from its current V(A) to its current V(S) inclusive, it shall set its V(S) to the value of the received N(R). Timer T200 shall be reset and the data link layer entity shall resume with I frame transmission.

4.7.2 Implementation of Communications Protocol at User End

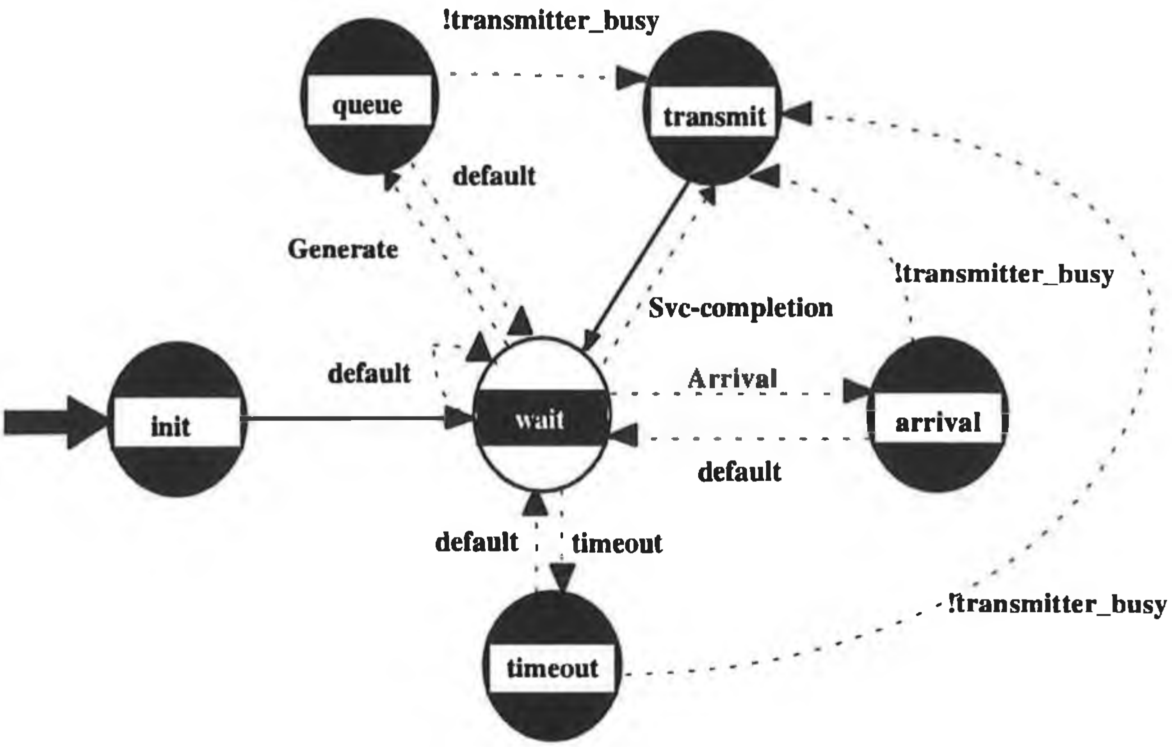


Fig.4.9 **User Process Model**

The LAPD process model state transition diagram is shown in fig. 4.12. The transition statements are defined as 'C' code macros. The *Generate* macro evaluates true if a packet from the ideal generator has arrived at the queue module causing a stream interrupt. The *Arrival* macro evaluates true if a frame has arrived from the network. The process model differentiates between the two transitions by noting which input stream the interrupt occurred on. The *Svc_completion* macro evaluates true when a self interrupt occurs, indicating that the transmitter is now idle. The macro, *timeout* also defines a self interrupt, and is an indicator. The process model differentiates between the two transitions by noting the code of the self interrupt. The transition *timeout* evaluates true when an interrupt scheduled by one of the states occurs. This is used to model the T200 timer in the LAPD protocol. The default transition statement is used to ensure that at least one of the departing transitions evaluates true.

The process model maintains state variables which represent non-volatile storage for variables which must maintain their value from state to state. They include the send state variable V(s), receive state variable V(r), acknowledge state variable V(a), flag to indicate reject exception condition, flag to enable/disable I frame transmission and a variable to maintain the unacknowledged frame count.

The temporary variable values are only valid within a state. Some of the temporary variables used were: type of frame received, value of command/response field in frame received, value of poll/final field in frame received, send sequence number N(s) of frame received and receive sequence number N(r) of frame received.

Description of States: The *init* state is entered when the 'begsim' interrupt occurs at the beginning of the simulation. It initialises the state variables for the process. After initialisation, the process immediately goes into the *wait* state where the process model simply waits for an interrupt to occur.

The queue state is entered when a packet arrives from the network layer which is modelled by an ideal packet generator with exponentially distributed interarrival times. Upon entering this state, the process acquires the packet from the input stream and attempts to insert it in subqueue 0. If the insertion fails, the packet is discarded. If I frame transmission is allowed, the packet from the head of the subqueue 0 is removed and a LAPD frame constructed. The fields N(s), N(r), address, p/f, type, and c/r are all set to appropriate values. A copy of the frame is made and inserted in the retransmission buffer. The frame is then transmitted and the send state variable V(s) and the unacknowledged frame count are both incremented. If the unacknowledged frame count is equal to the maximum window size W, I-frame transmission is disabled. If timer T200 is not already running, a self interrupt is scheduled for after *tout* period.

The *arrival* state is entered when a frame arrives from the network. When this state is entered, the frame is removed from the input stream and the relevant fields are stored as temporary variables. The first function performed by this state is to remove frames from the retransmission buffer which have been acknowledged by the N(r) field of the frame received. When this is finished the acknowledge state variable V(a) is set equal to N(r). The unacknowledged frame count is decremented by the number of frames acknowledged. If frames have been acknowledged, the timer is reset. The process then checks the type, p/f and c/r fields and responds accordingly. If the frame type is RR with p/f = 1 and c/r = 0 the peer entity has sent an enquiry frame so the process responds by transmitting an RR frame with p/f = 1 and c/r = 1 and N(r) = V(r). If the frame type is RR with p/f = 1 and c/r = 1 and the checkpointing enquiry flag is set, the peer entity must be responding to an enquiry sent previously by the process. The process then retransmits all the unacknowledged frames. If an I-frame is received with send sequence N(s) equal to the receive state variable V(r), the I-frame is in sequence and is accepted by the process and V(r) is incremented. If I-frame transmission is disabled due to the transmission being disabled, the process acknowledges the in-sequence I-frame by transmitting an RR frame with p/f = 0 and c/r = 0. If the I-frame is received out-of-sequence, the process responds by transmitting a REJ frame, and the reject-exception flag is set.

The *timeout* state is entered when the timer T200 expires. When in this state the process sends an enquiry frame (RR, p/f = 1, c/r = 0) to the peer entity. The timer is reset, and the timeout recovery condition is entered. When in this condition, the process prohibits I-frame transmission until it receives a response to the enquiry.

4.7.3 Flow Control mechanisms

Various flow and congestion control functions were added to enhance the communications protocol at the end user node. These functions were added to improve the performance of these networks under overload conditions.

4.7.3.1 Rotating window flow algorithm: Flow control was achieved by implementing the window flow algorithm. The flow control in standard LAPD protocol is based on the window mechanism i.e. a transmitter can continue transmitting I frames as long as the number of unacknowledged frames at the transmitting end is less than the "window" number. The successful reception of every I-frame is acknowledged by the receiver. The acknowledgement may either be individually acknowledged by a special acknowledgement frame such as a Receive-Ready (RR) frame, or the acknowledgement may be embedded as N(r) in I-frames flowing in the reverse direction.

Error recovery is performed by the *reject* and *timeout mechanisms* [17]. The *reject mechanism* operates as follows. Upon reception of an out-of-sequence I-frame, the receiver returns a reject (REJ) frame containing the sequence number of the expected frame and goes into the reject-state during which no further REJ frame is sent. It then discards all out-of-sequence frames. Upon reception of the REJ frame, the sender retransmits the I frames starting with the one expected by the receiver. When the expected frame arrives at the receiver, the reject-state is abandoned.

The *timeout mechanism* operates as follows. Immediately after transmission of an I-frame, the sender initiates the timer, if it is not already running. When the receiver an acknowledgement, it restarts the timer if unacknowledged frames remain. If the timer expires, the sender starts a *checkpointing* routine, by transmitting an RR frame with a special bit, called poll (P) bit, set to 1. Upon receiving this frame, the receiver returns an RR frame with the same bit, which is now called Final (F) bit, set to 1. This RR frame includes the sequence number of the I-frame expected by the receiver. Depending on this sequence number, the sender proceeds with the retransmission of a previous I-frame or transmission of a new one. If within the timeout period the sender receives no RR frame with the F bit set, it repeats the checkpointing routine.

If congestion develops at any network component in a frame relay network, the delay through the network will increase causing the end-to-end rotating window flow control to slow the input traffic. If a number of virtual circuits are active at one time, trunk buffers will fill up causing frames to be dropped. For the standard LAPD protocol, dropped frames can result in the retransmission of up to a windows worth of frames on each virtual circuit. This has the effect of increasing the congestion in the network. This suggests that an enhancement to the protocol should be provided. The protocol should contain an effective flow control mechanism to ensure efficient operation under both normal traffic load and overload. As mentioned earlier, one such enhancement is dynamic window flow control.

4.7.3.2 Dynamic window flow control: With dynamic window flow control each transmitter uses a window size W defined during link set-up. The procedures for frame transfer are similar to the procedures defined for fixed window flow. The major difference being that when a user enters error-recovery state (i.e. it receives a REJ-frame, or it does not receive an acknowledgement within a specified interval) it reduces its window size. This is because in frame relay networks, the probability of errors is negligible, and the arrival of a REJ-frame or the expiry of T200 timer, is a clear indication of congestion in the network. Since the traffic load is reduced by decreasing the window size, congestion is alleviated.

A number of dynamic windowing schemes were examined in this study. In all of them, receiving a REJECT frame, or a RR response to a checkpointing routine, is interpreted as a sign of congestion, to which the end-systems responded by reducing their layer-2 window. Under normal conditions, the layer-2 window is at its maximum value (W_{max}) established at call set-up. The actual window size (W) at any time may be different from W_{max} because of the changes caused by the network conditions.

The control schemes examined are:

1) On receiving a REJECT frame, or a response to a poll, the layer-2 window is *reduced by 1.* In this case, the window is increased by 1 if there are n consecutive successful deliveries of layer-2 frames. If the user enters error-recovery condition again, the window is further decremented by 1. In this way, if congestion persists in the network, the window can be decremented to a minimum of 1. By decreasing the window, the traffic load is decremented on the line. The rate at which the window is increased is a function of n. As n becomes larger, the window is increased more slowly. At the same time, the congestion and resulting losses become less frequent. This strategy is referred to as the *high-end linear dynamic windowing mechanism.*

2) In this case, if the user enters error-recovery condition, the window is reduced to 1. In this way, the load on the line is reduced drastically. The window will be slowly increased to its original size; the recovery procedure being similar to that in 1. If on the way to recovery, the user enters error-recovery condition, the window will be decremented to 1 again. This strategy is referred to as the *linear dynamic windowing mechanism.*

3) If the users enters error-recovery state, it will reduce its window *W to W/2;* the recovery procedure being similar to that above. If congestion persists in the network, that is, if the user continues to receive REJ-frames, it will continue to decrease its window to W/2; the minimum window size being 1. This strategy is referred to as the *half-window linear dynamic windowing mechanism.*

4) Reduce *W to 1;* the window is increased from w to w+1 after receiving w positive acknowledgements. Therefore the window increase becomes slower as the window becomes larger. Since the window size is almost a parabolic function of the number of acknowledgements, this strategy is referred to as *parabolic window mechanism.*

In all these dynamic windowing procedures, users enter error-recovery state only when actual packet-loss has taken place in the network. As a result, under congestion conditions, the retransmission traffic prevents the network from providing the maximal throughput. It would therefore be useful to notify the end-users of the congestion in the network before actual packet-loss takes place. This way, the users can reduce their windows and the traffic load before actual loss takes place, thereby avoiding any retransmission traffic in the network. This way, these networks can have much better performance under congestion conditions.

4.7.3.3 Congestion Notification Scheme: Networks deploying the congestion notification scheme use frame formats conforming to the CCITT Recommendation Q.922. The format of the address field is illustrated in fig. 3.3. Frame-relay-compatible DTEs that can handle the use of FECN/BECN signals are capable of providing much higher throughput under congestion conditions.

When the network detects a tendency towards overload, it can explicitly notify the end-user by setting the Forward/Backward Congestion Notification (FECN/BECN) bits. If the end-user receives a frame whose FECN bit is set, it realises that the packets sent by its peer-user are facing congestion in the network. The end-user therefore, sets the BECN bit of the Information or Control frames travelling in the reverse direction. When the end-user receives a frame with the BECN bit set, it is notified of the congestion in the network. The end-user then responds by reducing its window size to 1, thereby reducing the traffic load in the network.

4.7.3.4 Stop duration scheme: This scheme was used in conjunction with the congestion notification schemes. When the DLL cannot transmit I-frames at the rate at which they are generated, it applies backpressure on the generator. Specifically, a packet is not accepted by the link layer, when the total number of unprocessed data units reaches a threshold, equal to $W_{max}/2$. When backpressure is applied on the higher layer (equivalent to the generator here), the generation of data units is stopped for a fixed or randomly selected duration. This will result in the reduction of traffic load in the network. If the generation of packets is restarted only after the previously queued data units are processed and sent, the network may never enter a severely congested condition. In addition, the transit and end-to-end delay will not rise to exceedingly high values.

4.8 Frame Relay Process Model

The state transition diagram for the frame relay network node is shown in fig. 4.13. The *init* state initialises variables to appropriate values. The process then goes into the *idle* state. There are two transitions that can occur when the process is in the *idle* state. The transition labelled *Arrival* is followed when a stream interrupt occurs on any of the input streams. The process is then in the *queue* state. Upon entering the *queue* state, the frame that caused the stream interrupt, is removed from the input stream. Depending on the destination-address in the frame header, the frame is stored in the corresponding subqueue if it is not full. If the subqueue is full, the frame is discarded, hence leading to packet-loss. The number of the subqueues at the switch are equal to the number of output streams.

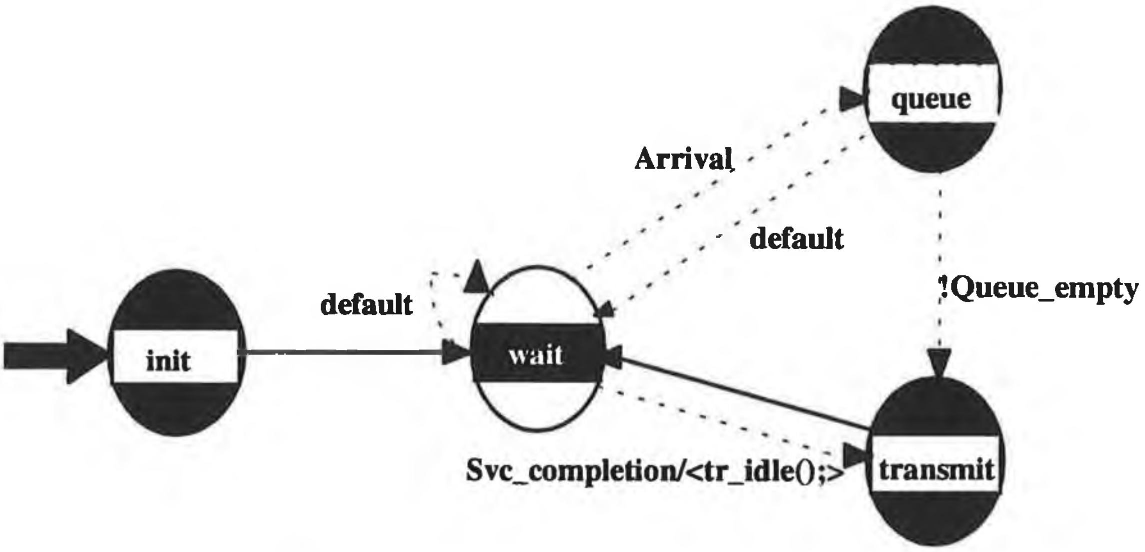


Fig. 4.13 **Frame Relay Process Model**

The *init* state initialises variables to appropriate values. The process then goes into the

idle state. There are two transitions that can occur when the process is in the *idle* state. The transition labelled *Arrival* is followed when a stream interrupt occurs on any of the input streams. The process is then in the *queue* state. Upon entering the *queue* state, the frame that caused the stream interrupt, is removed from the input stream. Depending on the destination-address in the frame header, the frame is stored in the corresponding subqueue if it is not full. If the subqueue is full, the frame is discarded, hence leading to packet-loss. The number of the subqueues at the switch are equal to the number of output streams.

The process then enters the *transmit* state. If one of the subqueues is not empty and the corresponding server is idle, the packet is removed from the subqueue and forwarded on the appropriate output stream. The server is then busy for (packet length/channel capacity) time. This is indicated in the process model by the scheduling of a self interrupt for this time.

The transition labelled *Svc_completion* becomes true when this self interrupt occurs. At this time, the process again re-enters the *transmit* state. In the transmit state, the process checks to see if there are any packets waiting to be transmitted. If a packet is waiting to be transmitted, and if the corresponding transmitter is idle, the process model will serve the packet.

Buffer management strategy: Some of the buffer management strategies implemented were: unrestricted buffer sharing, specific buffer allocation to each virtual circuit, and a combination of the two. In the latter strategy, users were allowed to share the buffers as long as the load was low. However, once the network detected a tendency towards overload, it would examine which user was using more than its allocated share. It would then prevent this user from hogging all the resources.

Service Discipline: The First-Come-First-Served (FCFS) service disciplines was implemented. Packets are served by the process model in the order of their arrival. However, if one of the users is overactive, it might prevent other users from getting a fair allocation of resources.

Handling of Overload Situations: The frame relay process model was modified by adding additional features for congestion control. The process will continuously monitor the input buffer and when it detects a tendency towards overload (i.e. when the queue exceeds a certain threshold value), it will request the end-user to reduce its data flow by sending a Forward/Backward Explicit Congestion Notification consisting of the checkbits of the frame header. Since Frame Relay is bi-directional, the process can set the BECN bit of the frames travelling in the opposite direction; or if there is no traffic in the reverse direction, it can set the FECN bit of the frame that encountered congestion. The network takes no action but leaves it to the end-user to decide on the best way of controlling its data flow. However, if overload occurs, the network will discard frames.

4.9 System Specification

Besides specifying the network, node and process models, the packet formats were specified using the Parameter Editor of the OPNET package. A number of data collection requests, called probes, were specified in the Probe Editor. These probes hierarchically reference a statistic, a module, a node, and a subnetwork; and are applied to a simulation at run time in order to cause the executing model to produce an output file. The probes were used to debug the new protocols and to evaluate the performance of the network. The Simulation Tool provided an environment for setting up one or more simulation runs, for specifying the input parameters and directing the collected data into named output files. The Analysis Tool was used to analyse simulation result data that was requested using probes in the Probe Editor. Data vectors could be plotted with a variety of graph types. Scalar values obtained from multiple simulation runs were collated and plotted to perform sensitivity analysis for user defined independent model parameters.

4.10 System Simulation

In the beginning stages of the development of the models, the OPNET simulation tool and debugger were used to test specific mechanisms in the protocol. This was achieved by running one simulation at a time. Once the validity of the implementation was confirmed, larger runs of simulations were attempted to obtain a broader data set.

In order to assess the performance of the modelled systems, a number of pre-defined and user-defined statistics were specified in the **probe file**. By specifying the probe file in a given simulation run, these statistics were collected. The simulation probes can measure a number of pre-defined and user-defined statistics. Some of these statistics were recorded as time series vectors in the **output file** specified. These output files were useful for measuring dynamically varying data from the execution of a single simulation run, and were therefore useful for studying the dynamic behaviour of the system. The output vectors can also be used to verify the correct operation of the protocols. However, these output files cannot be contributed to by multiple simulation runs.

Since this simulation study involved the assessment of the effect of varying one or more parameters of the model on one or more output parameters, data across multiple simulation runs was accumulated and stored in a **scalar file**. This was achieved by executing a separate simulation for each new permutation of inputs and recording the resulting output values. The scalar files were used to record some aspect of an output vector over a simulation run, such as its mean, variance, maximum, minimum etc.

In addition, the parameters of the model can be specified in the simulation run table. Since OPNET supports the execution of a sequence of simulation runs, these parameters can be assigned different values in one simulation run specification.

4.11 Conclusions

The importance of modelling and analysis techniques are evident from this study. A network engineering tool, OPNET was used to study the existing flow and congestion problems in frame relay networks. It provided an excellent environment to debug and fine-tune the solutions proposed and to evaluate their effectiveness. The protocol was modified with relative ease, and minimal changes were made in the existing protocol. In addition, it was also used to decide on the gross design parameters of the network models.

Chapter 5

Performance of Frame Relay Networks with Fixed Window Flow Control

5.1 Introduction

In this chapter, the performance of ISDN frame relay networks is evaluated. For this purpose, LAPD data link protocol was implemented at the end nodes of the developed network. The LAPD protocol includes procedures to ensure the correct and timely delivery of data to the end users. The protocol at the intermediate nodes of the network is very simple and these nodes only perform a very simple routing and store-and-forward function. Error recovery and flow control procedures are only implemented on end-to-end basis. This highly streamlined protocol processing makes the network operation more efficient, thereby improving network delay and throughput performance. However, the absence of control capabilities at the network nodes makes these networks vulnerable to congestion, and results in the degradation of network performance. To prevent the network performance from degrading too severely, it is important to select network parameters carefully. The effect of these parameters is demonstrated in this chapter. The Analysis Tool of the OPNET package was used to evaluate the performance of the developed frame relay network models.

5.2 Offered Load

The nature of the traffic offered by the user devices is a major factor in determining the performance of the network. This offered traffic load in data communication networks is usually non-uniform or stochastic in nature. In this study, the higher layers sending the data onto the frame relay network are modelled by the generator module in the user node. The offered load per user is defined as the mean number of bits generated per unit time. However, the load on the network is dependent on the number of active virtual circuits. If a number of users are transmitting data in the same direction on a link, then the load on the link is the sum of the individual loads.

In this study, the highly variable offered traffic load was obtained by writing a process model for a bursty generator. This was useful for studying the dynamic performance of the developed model.

However, because of the extreme variability of the traffic offered to the network, it is difficult to define typical steady state distributions of network traffic. Studies reveal that a Poisson packet arrival process provides the best steady state description of network traffic. Since a Poisson arrival process is equivalent to an exponential distribution of packet inter arrival times, an exponential inter arrival pdf (probability distribution function) was specified at the Ideal Generator module of each user-node. The rate at which the generator produces packets is controlled by the inter arrival args attribute. This attribute is used to specify the mean time between the generation of packets. The simulation has to be run for a long enough time for the inter arrival time between packets to stabilise at this value. In this study, the effect of the offered load on the performance of the developed system was assessed by running a series of simulations for different values of the inter arrival args.

5.3 Network Performance Measures

To evaluate the performance of the modelled systems, the two performance measures used were **Throughput** and **delay**.

Throughput in bits per seconds, can be defined as the average number of correct bits passing a given point in a network per unit time. [12]. In this study, the throughput per logical link is defined as the number of correct, in-sequence bits arriving at the destination per unit time. In the modelled network shown in fig. 5.1, if the total offered load is taken as the sum of the loads generated by users in subnet 1, then the total throughput would be the sum of the correct, in-sequence bits arriving at the users in subnet 2, per unit time. The total throughput is defined as the sum of throughputs on all logical links.

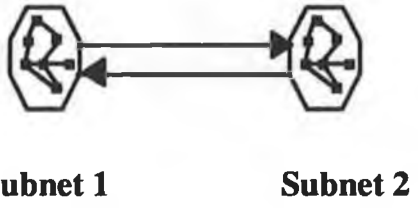
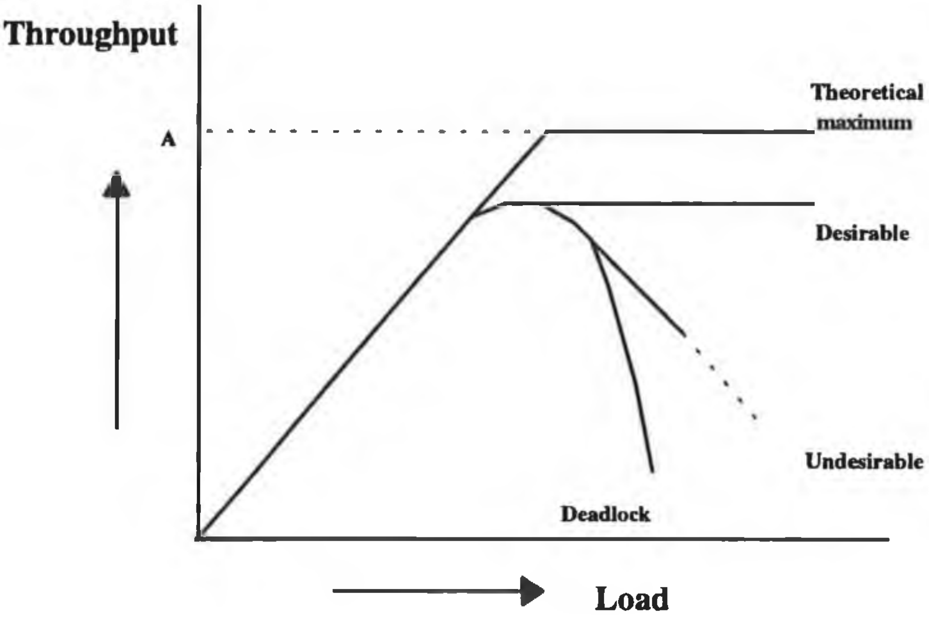


Fig. 5.1 **Network Model**

The end-to-end delay is defined as the time elapsed between supplying a data unit to the link layer at the higher layer interface in the source node until receiving it across the interface at the sink node. It is measured by stamping packets with the current simulation time when they are generated, then finding the elapsed time when the packet arrives at the destination node. The end-to-end delay depends on the time that an I-frame has to wait at the source before it can be processed and the transit delay. The transit delay is defined as the time that it takes for a processed data unit to reach the high layer interface at the sink node in its correct form.

For most of the results presented in this chapter, the network model shown in fig. 5.1 was used. Each user in subnet1 is assumed to have a logical link set up to a corresponding user in subnet2. I-frame transmission on each logical link is two-way. The propagation delays and all the processing times were neglected. The buffer management policy and the transmission scheduling discipline at each network node were assumed to be unrestricted sharing and first-come-first-served respectively. The steady state performance of the frame relay network was evaluated by plotting throughput and transit delay as functions of the offered load.

Fig. 5.2 shows the ideal throughput as a function of the offered load. When the offered load is less than the network capacity, the ideal throughput is equal to the load, otherwise it equals the network capacity. However, due to the inadequate provisioning of resources, the behaviour of the networks can become undesirable (that is, with an increase in the load, the throughput drops) as shown in fig. 5.2.



A = Maximum Throughput capacity of the network

Fig. 5.2 **Network Throughput versus Load**

5.4 Window Flow Control

Flow control is required to prevent a receiver from being overwhelmed with data it cannot handle. In ISDN frame relay networks, a window flow control mechanism is used to control the flow of packets in either direction. With this scheme, each user can send up to W (window) number of packets before receiving a transmission authorisation from the receiver. This mechanism provides congestion control as well. If congestion develops at any network component, the end-to-end window flow control is effective in slowing the input traffic to some extent, but may not be sufficient to alleviate the condition of congestion.

A window of size W is negotiated on setting up of a logical channel. Since the window represents the maximum number of data packets that may be outstanding in any one direction, the performance of the system is dependent on the selection of this parameter. If the sender's window is small and the round trip time is large, the performance of the system is seriously affected. This is because the sender is forced to wait for an acknowledgement before it can commence transmission again. Much better bandwidth efficiency can be obtained if a sender is allowed to continuously transmit frames for a time equal to the round-trip time without filling up the window.

Figs. 5.3 and 5.4 illustrate the effect of the window size on the performance of the system. As can be seen, the load offered to the system is less than the maximum capacity of the system. Simulations were run for window sizes of 1 and 7 respectively. With a window of 1, a user can only transmit a packet at a time. It can transmit the next packet only after receiving an acknowledgement from its peer. This results in the poor bandwidth utilisation of the link, and the poor throughput performance of the system. For a window size of 7, the throughput follows the offered load for loads less than the channel capacity.

With larger window sizes, there will be greater queuing delay at the network nodes. As a result, the transit delay is lower for a window of 1 in comparison to a window of 7. However, the end-to-end delay is greater for a window of 1. This is because it takes longer for a sender to process packets and transmit them. The results are presented in figs. 5.5 and 5.6.

From these results it is evident that a very small window can restrict the flow of data unnecessarily and can result in the inefficient utilisation of the bandwidth. It is therefore essential to select larger windows to ensure the continuous transmission of data.

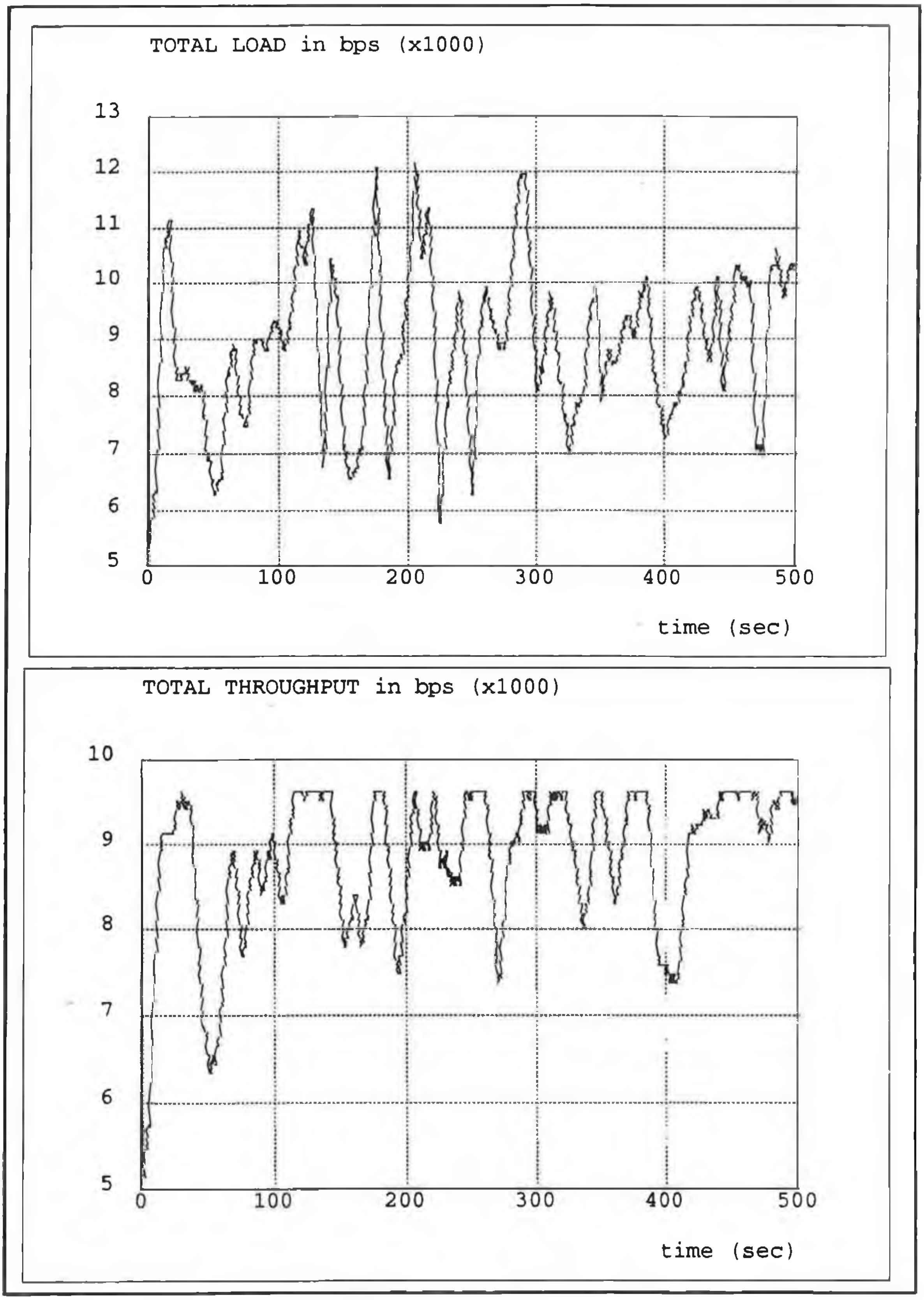


Fig. 5.3 **Performance of model with W=1**

(a) Load versus time; (b) Throughput versus time for the offered load in (a)

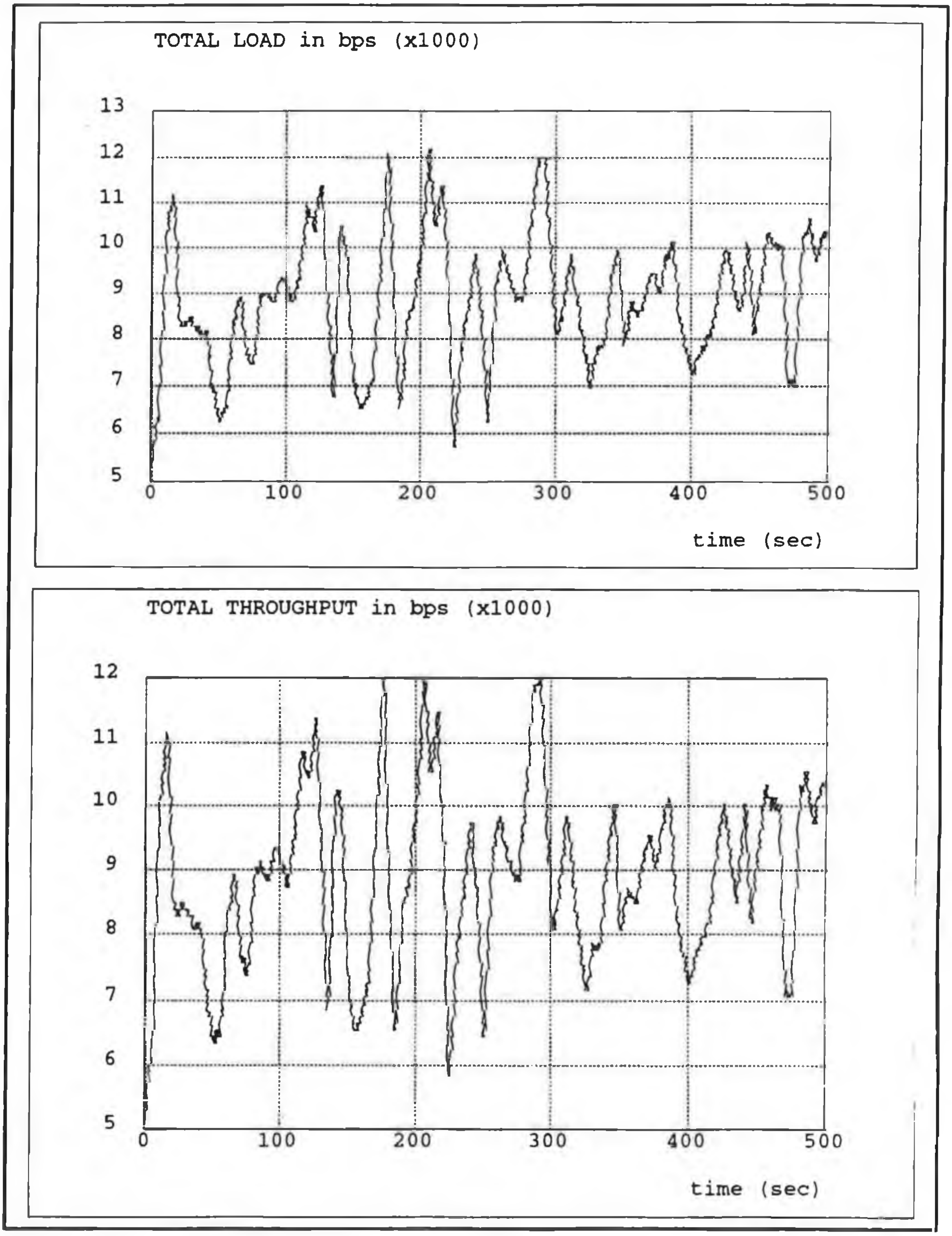


Fig. 5.4 **Performance of model with W=7**

(a) Load versus time; (b) Throughput versus time for the offered load in (a)

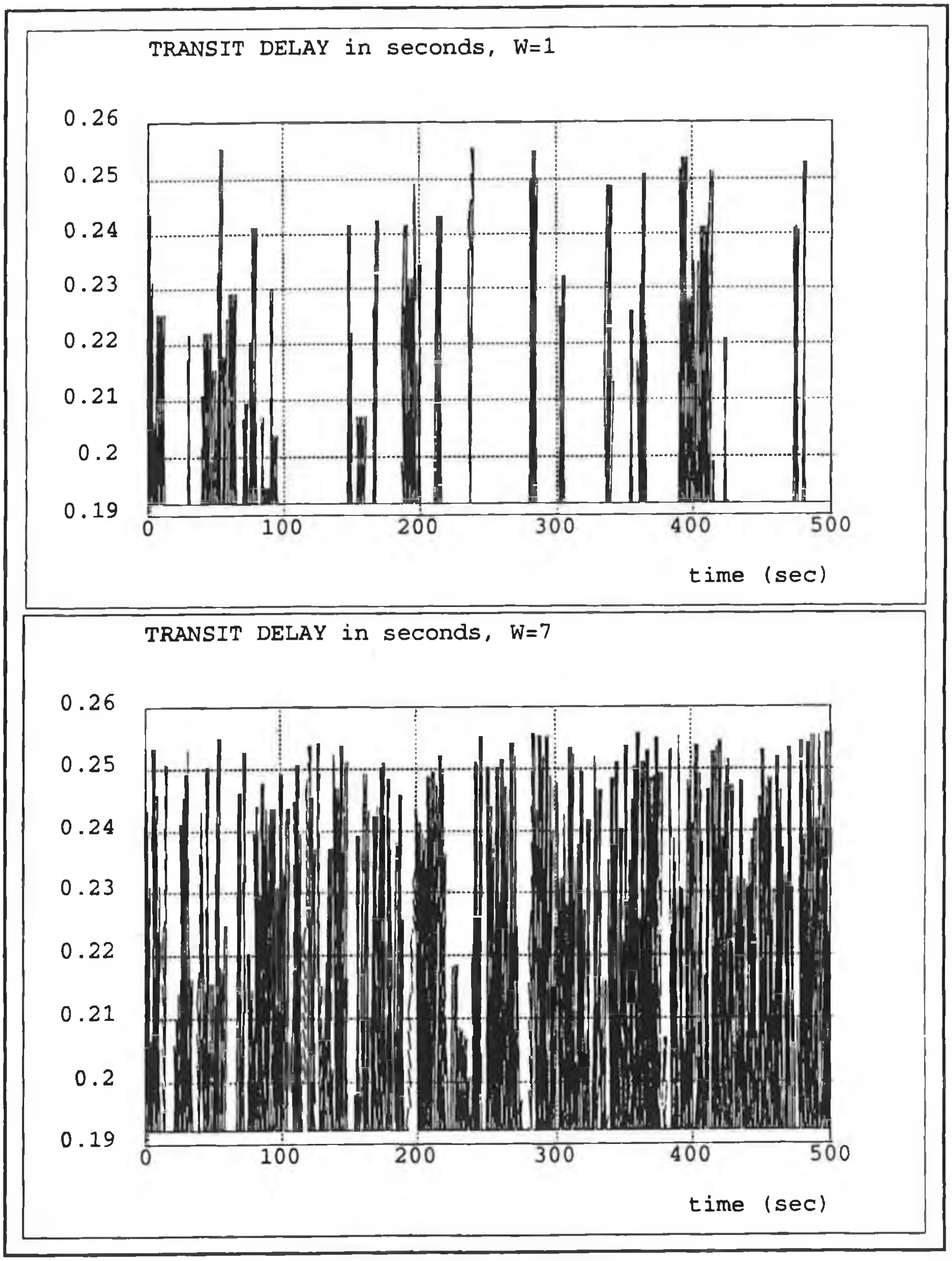


Fig. 5.5 **Transit Delay Performance for the Offered Load in Fig. 5.3(a)**

(a) W=1; (b) W=7

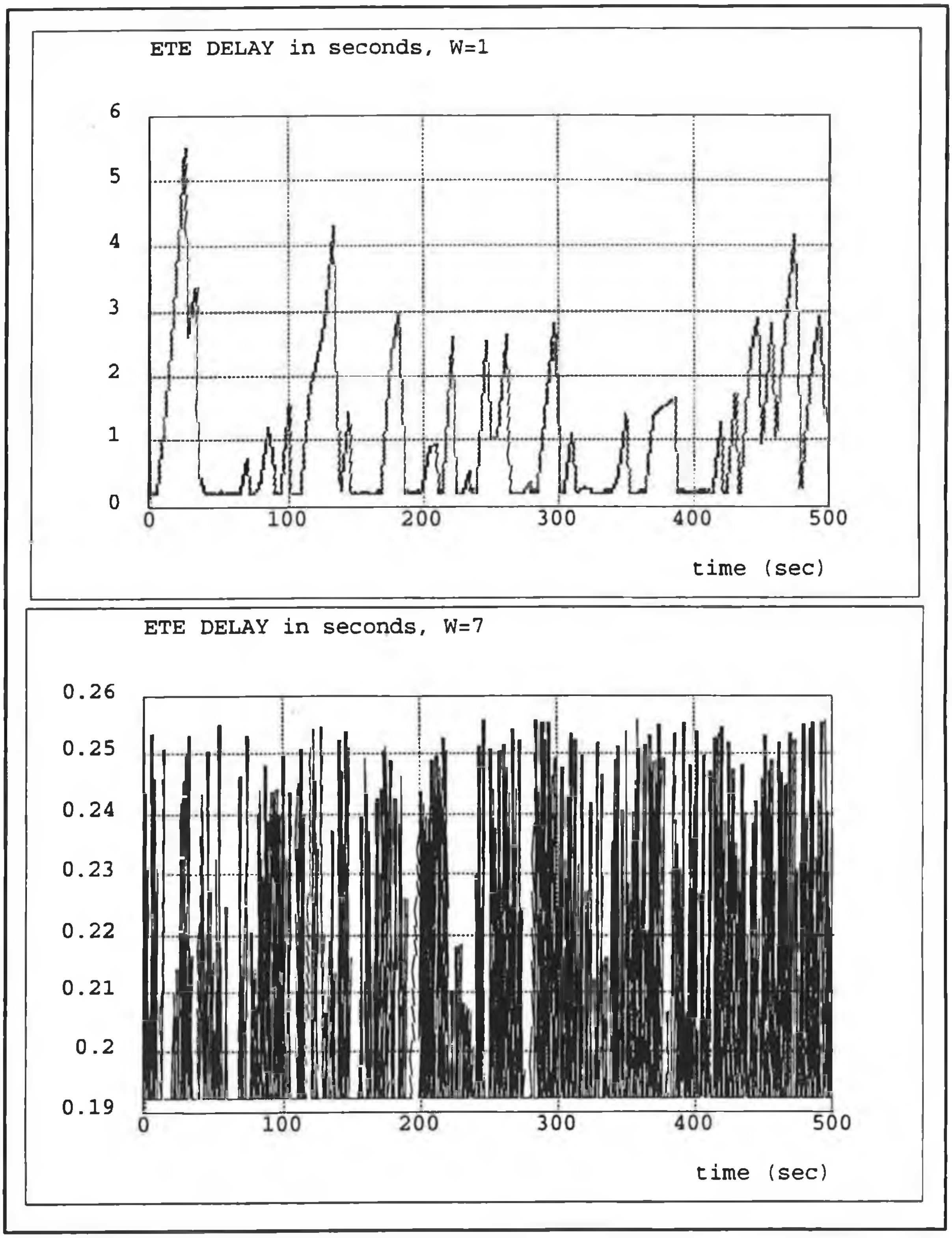


Fig. 5.6 **End to End Delay Performance for the Offered Load in Fig. 5.3(a)**

(a) W=1; (b) W=7

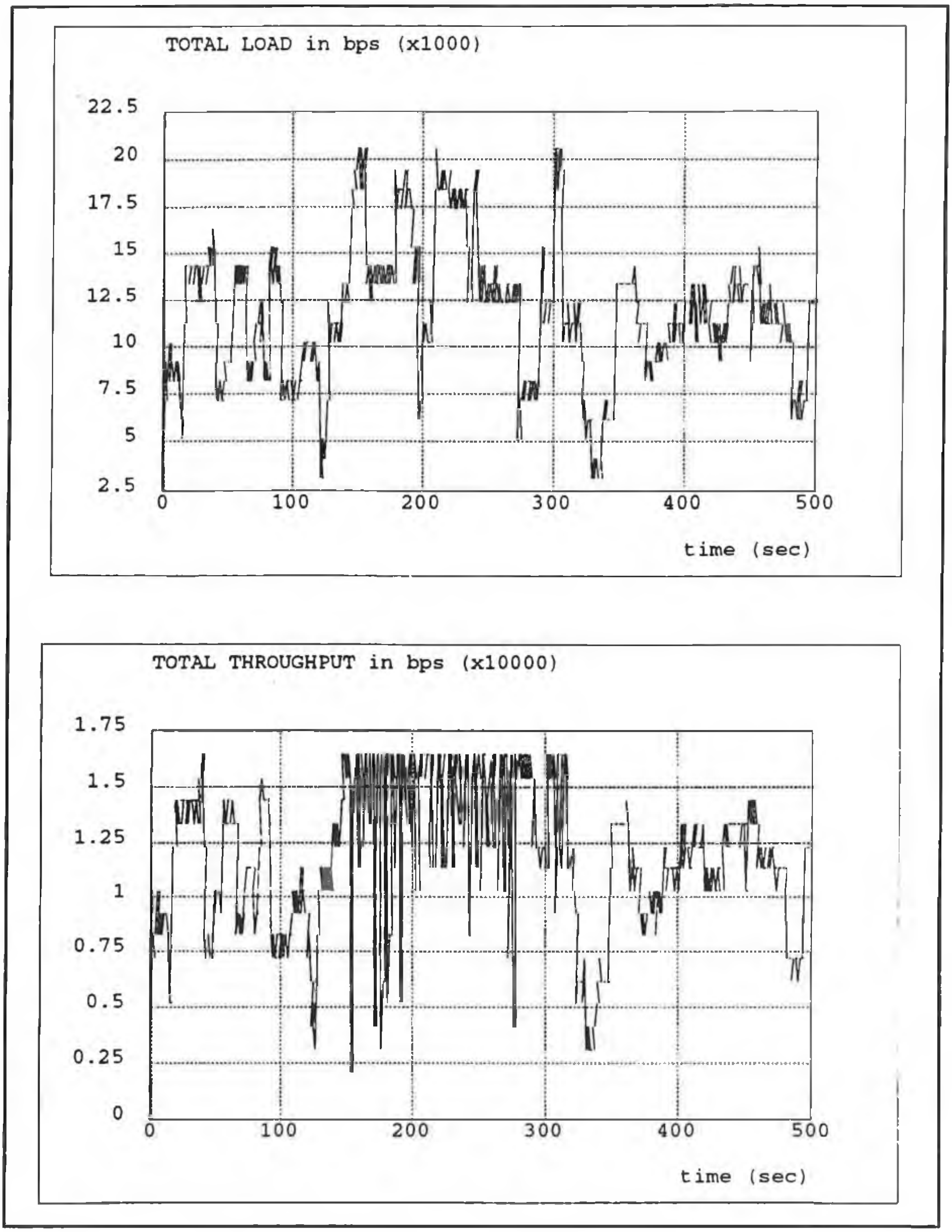


Fig. 5.7 **Performance of model with W=20**

(a) Load versus time; (b) Throughput versus time for the offered load in (a)

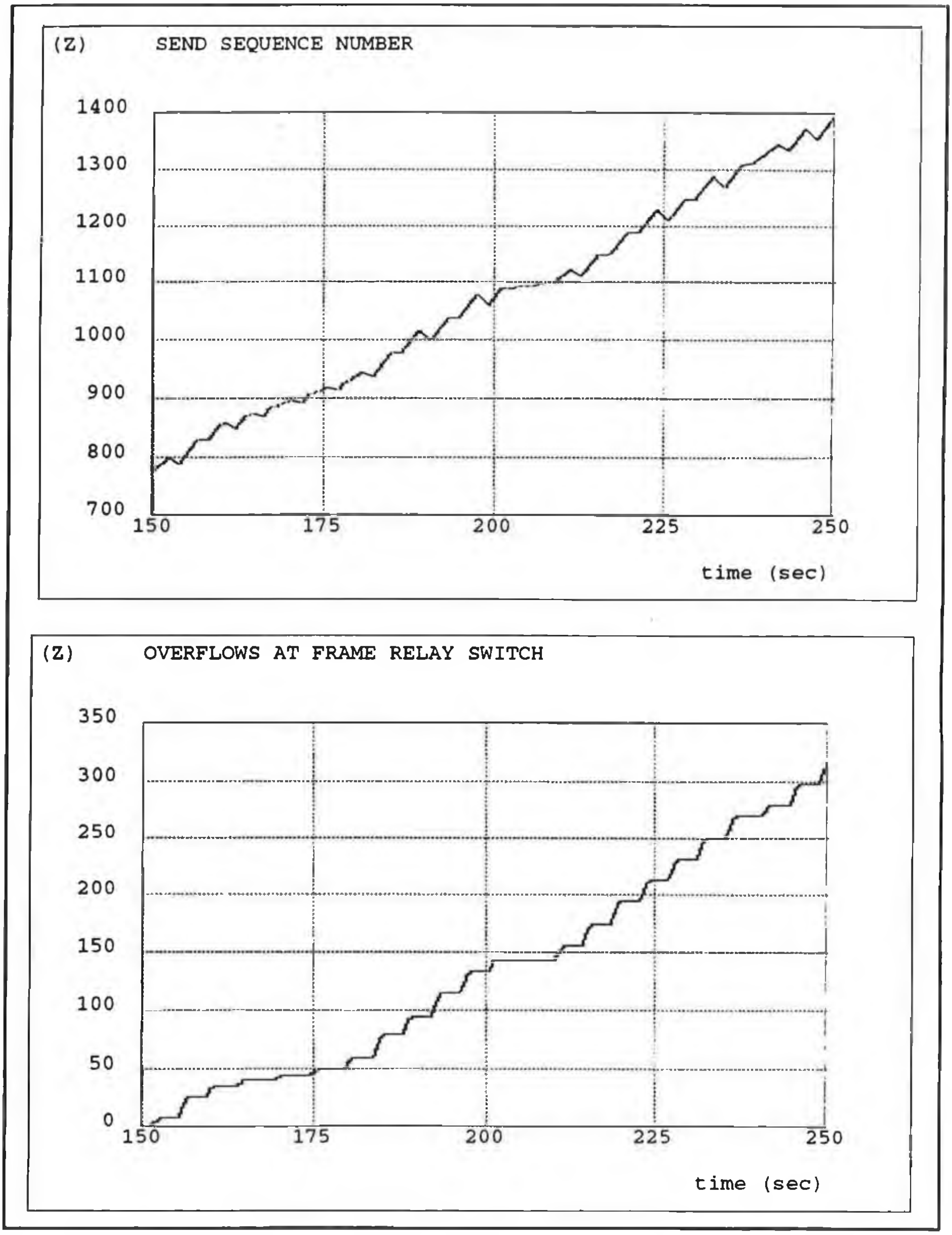


Fig. 5.8 **Performance with W=20 for the Offered Load in Fig. 5.7(a)**
(a) Send Sequence Number; (b) Overflows at Frame Relay Switch

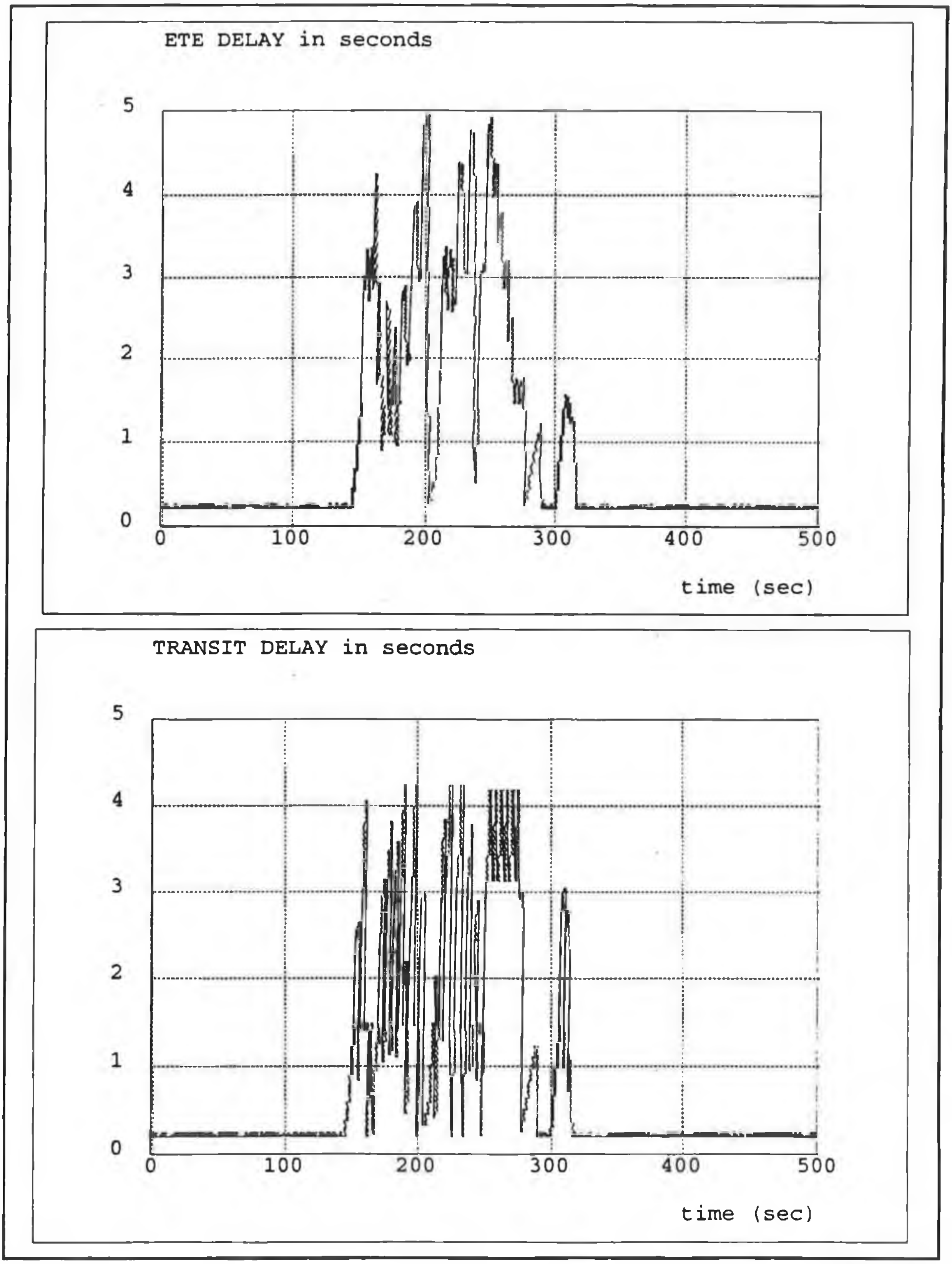


Fig. 5.9 **Performance with W=20 for the Offered Load in Fig. 5.7(a)**
(a) End to End Delay; (b) Transit Delay

The dynamic performance of the system was studied for a window of 20. Fig. 5.7 illustrates the throughput performance of the system. The throughput follows the offered load linearly for loads less than the channel capacity. As the input traffic approaches overload, queues of frames waiting for service begin to build up in the frame relay switches eventually leading to the loss of frames. This is depicted in fig. 5.8(b). This triggers retransmissions at the end users. In fig. 5.8(a), the dip in the send sequence number illustrates the retransmitted frames. The retransmissions of frames discarded due to overflow cause the buffer utilisation to increase which in turn further increases the overflow probability. This positive feedback causes abrupt increases in the buffer utilisation, overflow probability and the fraction of the network capacity used for retransmissions. As a result there is a sudden loss in throughput, as can be seen in fig. 5.7. The delay performance of the system is illustrated in fig. 5.9. At small values of offered load, the delay is small. However, the delay increases as the load is increases owing to the buffer queuing delay in the frame relay switches.

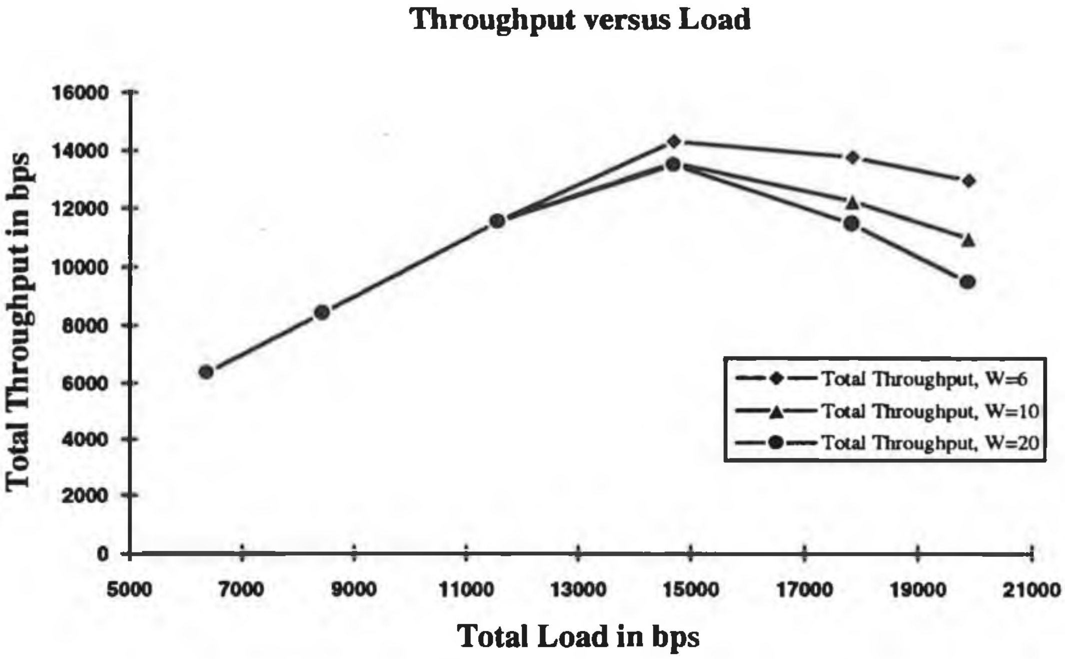


Fig. 5.10 **Effect of Window size on the Throughput versus Load Performance**

To study the effect of the window size on the overload performance of the system, a series of simulations were run. Fig. 5.10 illustrates the results obtained by simulating the network model shown in fig. 5.1 for different window sizes. The throughput is plotted as a function of the load for different window sizes.

Plots of the mean transit delay as a function of the offered load are also shown in fig. 5.11. At small values of offered load, the mean delay is small, but increases as the load is increased owing to the buffer queuing delay in the frame relay switches.

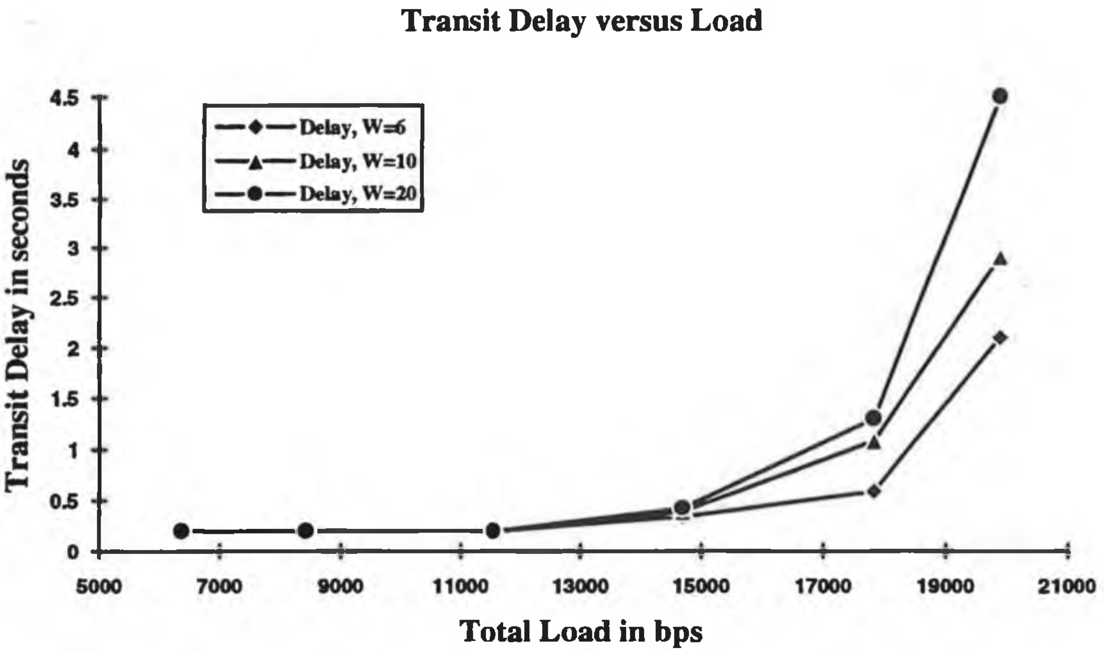
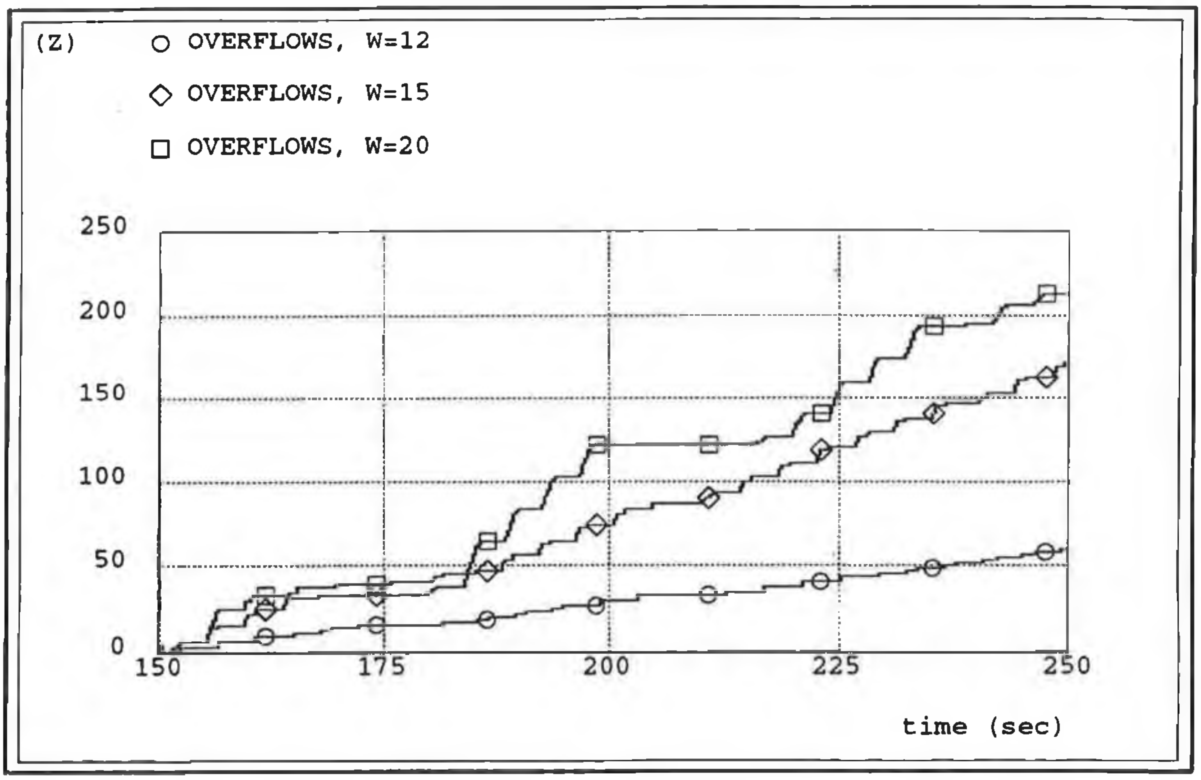


Fig. 5.11 **Effect of Window size on the Transit Delay versus Load Performance**

It is evident that the overload performance of the system is better for smaller window sizes. Since the window size sets a maximum to the number of I-frames that need to be retransmitted for each lost frame, the extra traffic generated by retransmissions decreases with the window size, and the overload performance of the modelled system improves. The probability of overflows taking place is greater, when the window is larger. This was also observed dynamically. For the same load as shown in fig. 5.7, simulations were run for different window sizes. The number of overflows is less for smaller window sizes, as can be seen in fig. 5.12(a). In addition, the end-to-end delay increases to greater values for larger window sizes. This is because of the greater retransmission traffic generated and the greater queuing delay of the packets in the buffers of the frame relay switches.

From this study, it is evident that the overload performance of a system is better when the window is smaller.



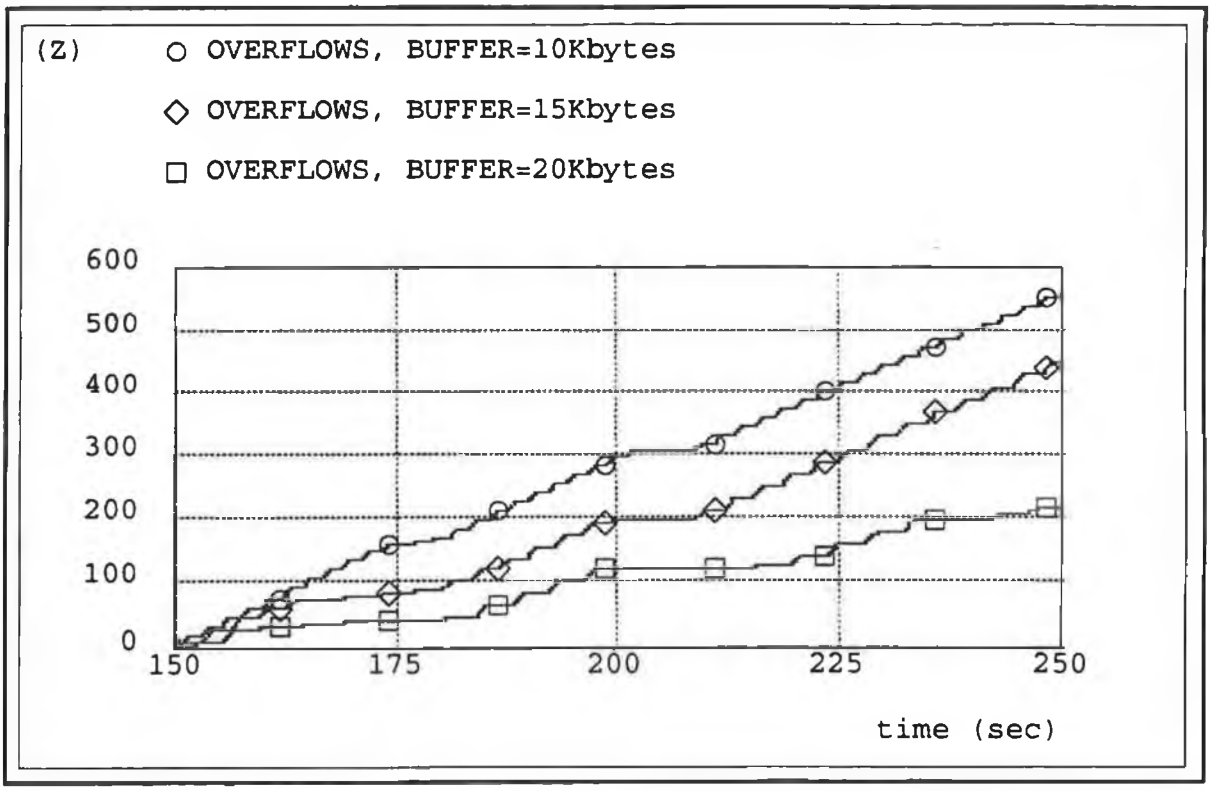


Fig. 5.12 **Overflows at Frame Relay Node; for the Offered Load in Fig. 5.7(a)**
(a) Effect of W size; Buffer=20Kbytes; (b) Effect of Buffer size; W=20

5.5 Buffer size

In addition to the careful selection of window size, the adequate provisioning of network resources will prevent the network performance from deteriorating under overload conditions. If the trunk buffers are large enough to hold window worth of data for the maximum possible virtual connections, there will never be any packet loss in the network and there will be no contention at the resources. However, due to the bursty nature of the data traffic, it is not economically feasible to allot a window worth of buffers for each virtual circuit. Instead, the buffers should be designed to minimise the probability of overload.

By varying the buffer size in a series of simulation runs, the effect of this parameter on the performance of the modelled system was realised. The larger the buffer size, the fewer the losses in the network. This is illustrated in fig. 5.12(b). As a result, there would be less retransmission traffic, and the performance of the system would be better. Fig. 5.13 illustrates the effect of the buffer size on the performance of the system.

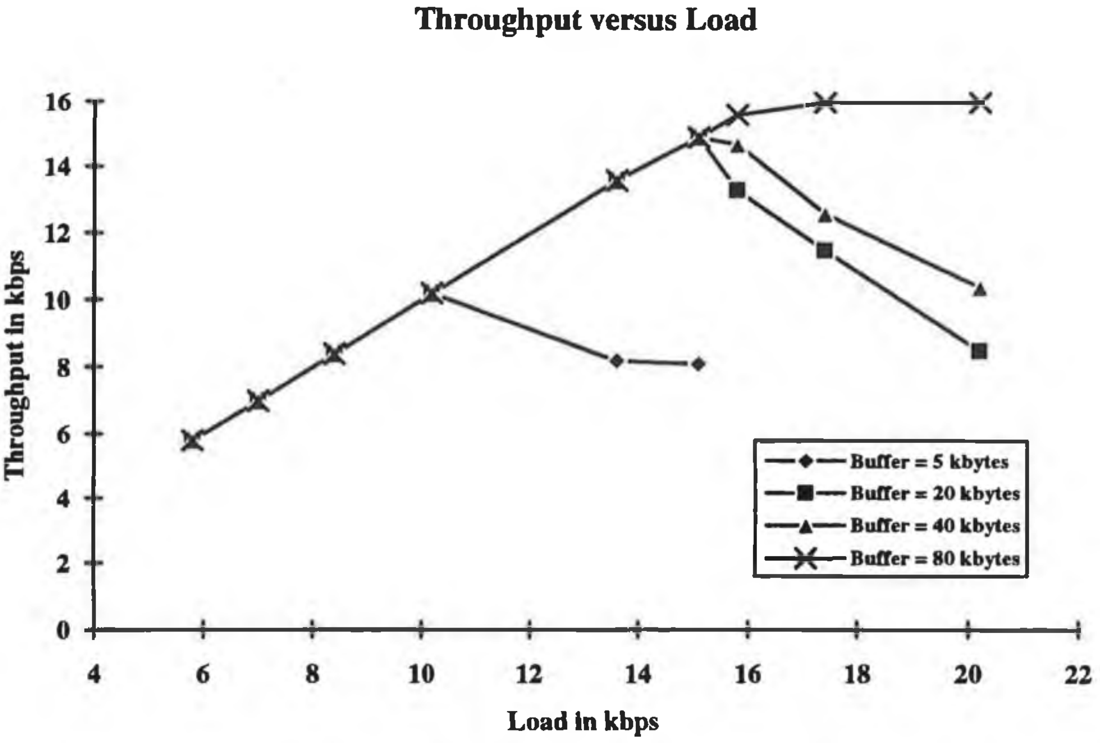


Fig. 5.10 **Effect of Window size on the Throughput versus Load Performance**

5.6 T200 Timeout

With the window flow control, each user can send up to W (window) number of packets before receiving an acknowledgement from its peer-user. After the user has transmitted W packets, it has to wait for an acknowledgement from the receiver before it can commence I-frame transmission again. However, to prevent the transmitter from waiting forever for a positive or a negative acknowledgement that had itself been lost, or delivered in error and discarded, it is necessary to build a timeout procedure in the protocol.

With the timeout procedure implemented in the protocol, the user enters timer recovery state if it does not receive a positive or a negative acknowledgement within a predefined timeout period. It then polls its peer and takes the necessary recovery actions on receiving the response.

This timeout value should be selected carefully. The time for an acknowledgement to arrive depends on the number of hops, and the queuing delay at each of the intermediate nodes in the network. In order to ensure that the timer does not expire for normal queuing delays, the timeout period should be greater than the maximum acknowledgement time.

If this is not the case, the transmitter may enter timer recovery state even though the acknowledgement is not lost. Once the transmitter enters timer recovery state, it waits for a response to its checkpointing poll before transmitting any more I-frames. Since the timeout period is less than the maximum round trip time, it may time out again before receiving a response to its checkpointing poll. In this situation, the transmitter is forced to stay idle even though there is no congestion in the network. This causes the end-to-end delay and the transit delay to increase. With a properly selected timeout value, the performance of the network is noticeably better. The results obtained by simulating the same network for different timeout values is presented in fig. 5.13.

5.7 Conclusions

Since error-recovery and flow control is end-to-end in frame relay networks, these networks are vulnerable to congestion. Although congestion can result in the performance degradation of these networks, the proper selection of network parameters can prevent the performance from degrading too severely. The effect of the window size, timeout value and proper dimensioning of network resources is illustrated in this chapter.

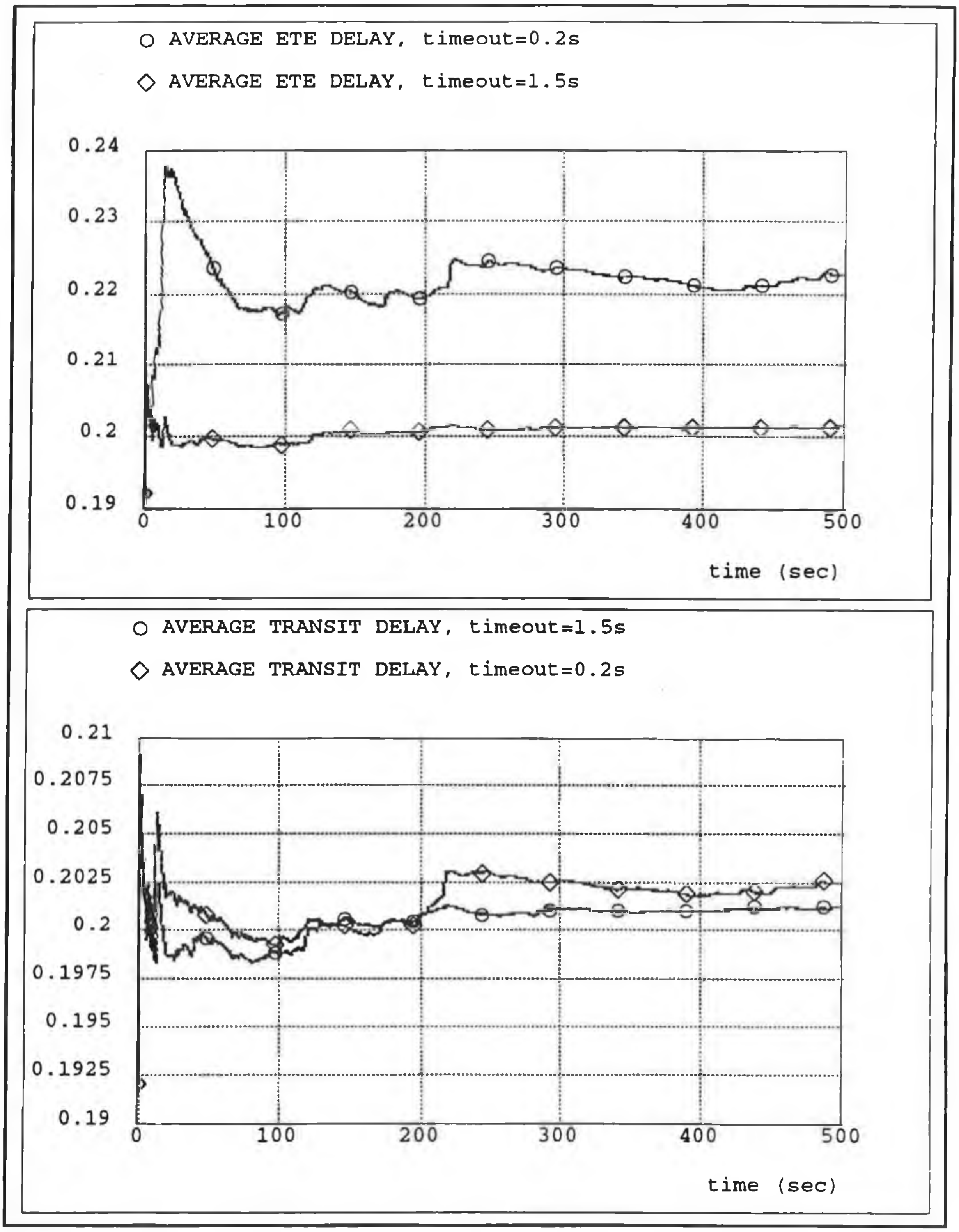


Fig. 5.13 **Effect of Timeout values on the Delay Performance;**
(a) Average End to End Delay; (b) Average Transit Delay

Chapter 6

Performance of Frame Relay Networks with Other Congestion Control Techniques

6.1 Introduction

The vulnerability of ISDN frame relay networks to traffic overloads was demonstrated in Chapter 5. To cope with such vulnerabilities and to meet the high expectations of end-user performance, effective congestion control strategies are needed. These strategies should ensure that the network operates at peak efficiencies at all times, without incurring excessive overheads. The results obtained by implementing dynamic windowing, congestion notification and stop-duration techniques are illustrated in this chapter.

6.2 Adaptive Window Schemes

From the results presented in Chapter 5, it is evident that a very small window is unnecessarily restrictive and prevents the bandwidth from being fully utilised. This can lead to network performance degradation inspite of their being no congestion in the network. At the same time, the overload performance of the network is better for smaller window sizes. To ensure that the network operates at peak efficiency at all times, a dynamic windowing scheme is proposed. According to this scheme, each logical link connection negotiates a maximum window size W. When there is no congestion in the network, the window size of each end point is set at this maximum value. However, upon detection of a frame loss, the end points reduce their window sizes. Dropping the window size reduces the load on the network, thus allowing the network to recover from congestion. The window size is gradually increased after successful transmissions. The various dynamic flow control schemes are described in Chapter 4 and the effect of these schemes on the performance of the network is analysed in this chapter.

The schemes used to reduce the window of a user are defined here.

(1) When a user detects congestion in the network, it responds by reducing its window by 1.

(2) The user responds to congestion by reducing its window to half its present size.

(3) In this case, a user responds to congestion by reducing its window to 1.

In all the three cases, as the network recovers from congestion, the user responds by opening its window.

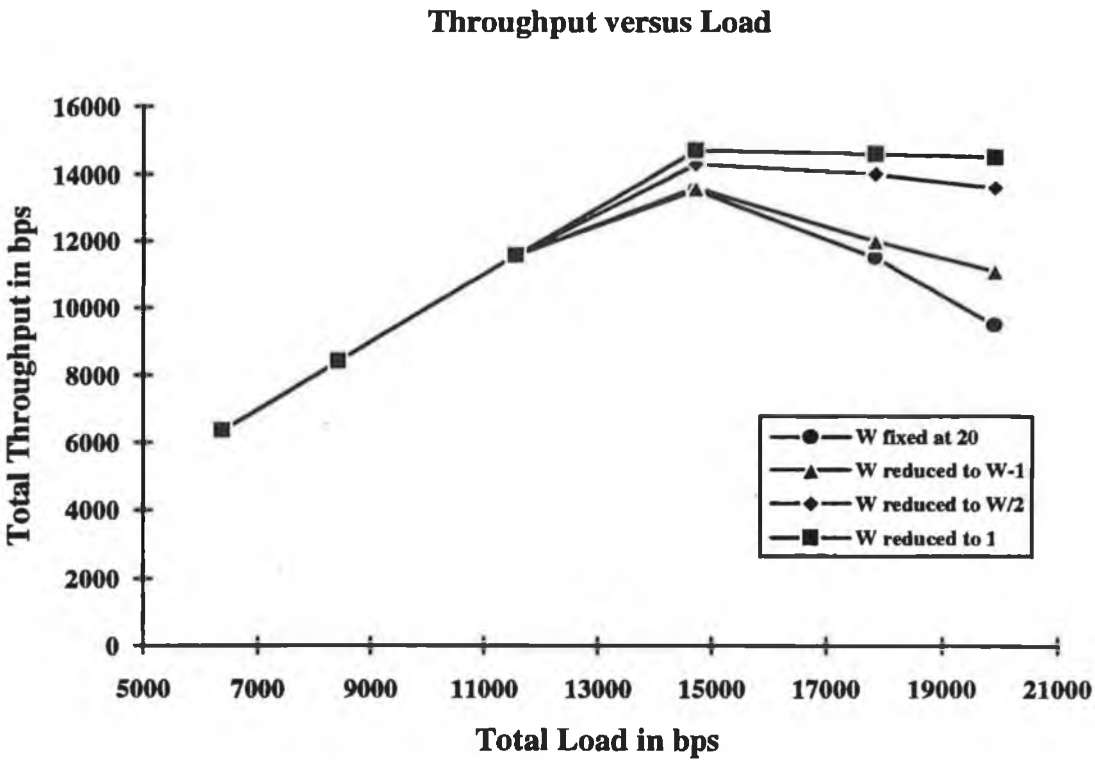


Fig. 6.1 **Effect of Window Reduction Technique on the Throughput versus Load Performance; W_{max}=20; $W_{min=1}$**

The results obtained from simulation are presented here. In all the cases, the maximum size of the window was taken to be 20 and the minimum size of the window was taken to be 1. Figs. 6.1 and 6.2 illustrate the results obtained by using the three reduction techniques. In all the cases, the linear window increase mechanism was used to open the window. More specifically, after every n successful transmissions, the window size is incremented until the window reaches its original size. However, if on the way to recovery, a user receives another negative acknowledgement, it will respond by reducing its window again. The method of window reduction depends on the technique used. Since the reduction in window size results in a reduction in the load on the network, the performance of the network improves. It is evident from the results presented that the performance of the network is best when the window is reduced to 1. In addition, the overload performance of the network is better when the window is reduced to half its size, as compared to the network performance when the window is reduced by one. This is obvious because the networks will recover more quickly from congestion if the users offer less or no traffic.

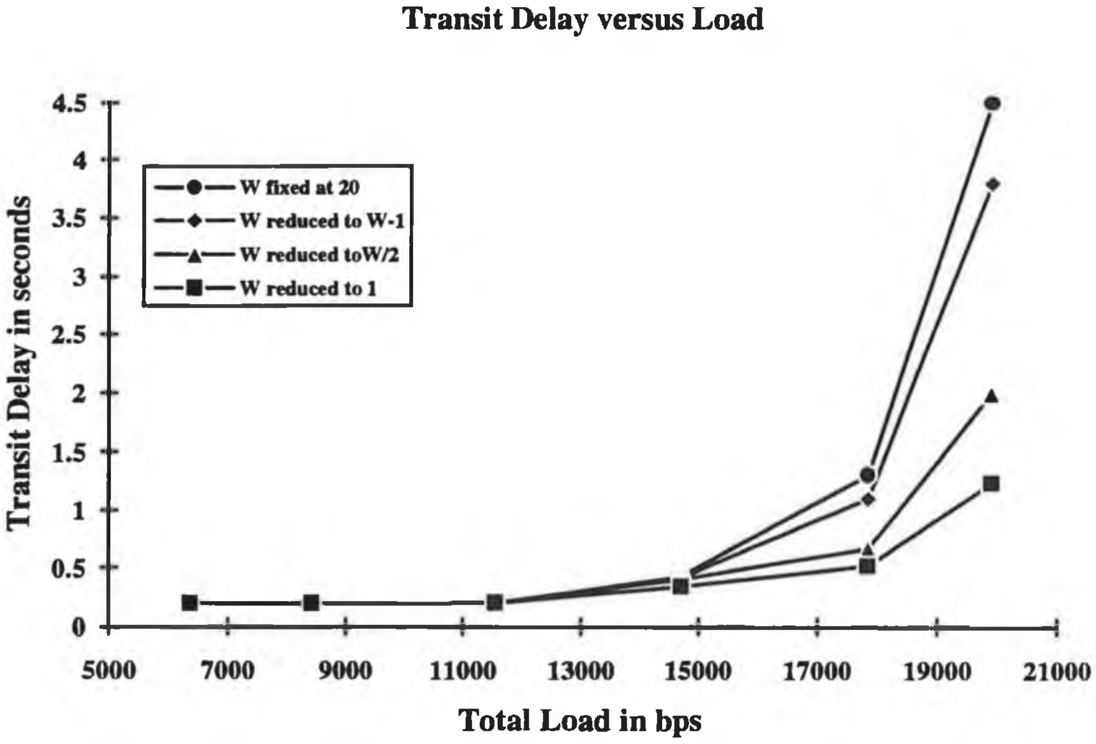


Fig. 6.2 **Effect of Window Reduction Technique on the Transit Delay versus Load Performance; W_{max}=20; $W_{min=1}$**

Figs. 6.4 indicates the improvement in the throughput performance of the network, when the window is reduced to 1 in case of overload and is increased by 1 after 10 successful transmissions. The variation in the window size is depicted in fig. 6.3. When the user detects the congestion in the network, it reduces its window. As the input traffic is reduced, there are less overflows taking place as indicated in fig. 6.4(b). In addition, there are fewer retransmissions as indicated in fig. 6.4(a). The window is increased by one foe every 10 successful transmissions. If on the way to recovery, the user receives a negative acknowledgement or enters timer recovery state, it responds by reducing its window to 1 again.

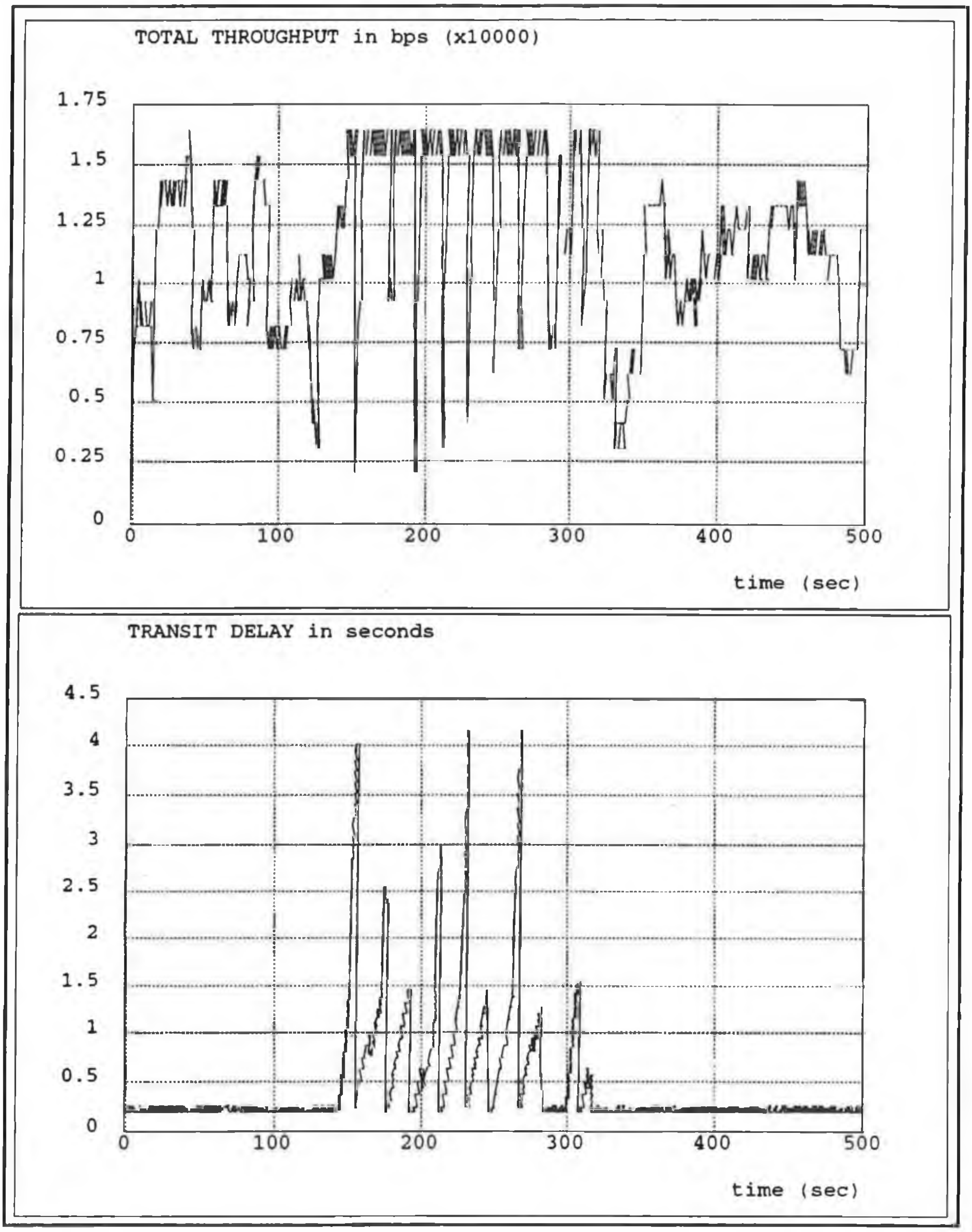


Fig.6.3 **Performance with Dynamic Windowing, for the Offered Load in Fig. 5.7(a)**
W_{max}=20, W_{min}=1, n=10 (a) Throughput versus time; (b) Transit Delay versus time

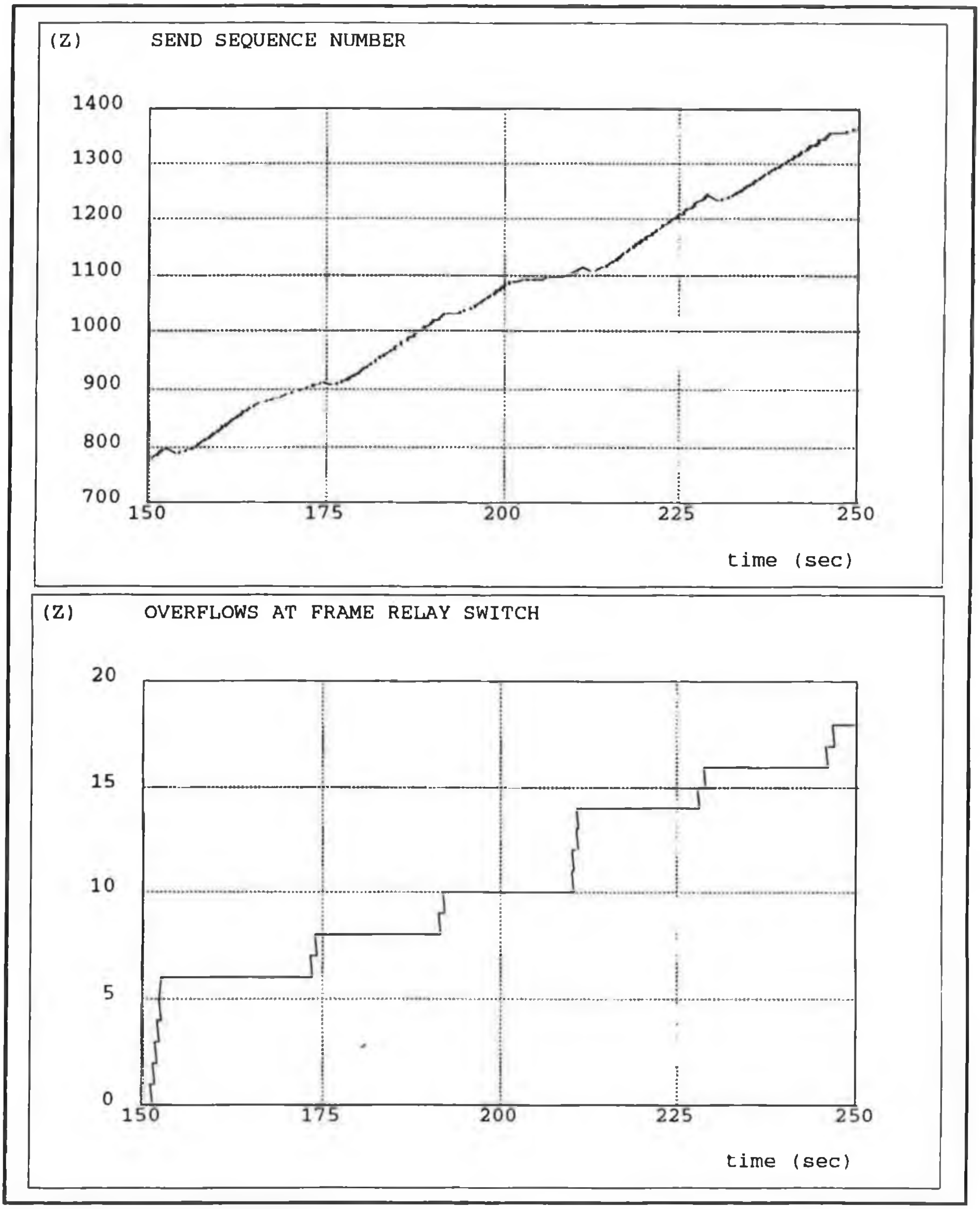


Fig.6.4 **Performance with Dynamic Windowing, for the Offered Load in Fig. 5.7(a)**
W_{max}=20, W_{min}=1, n=10; (a)Send Sequence Number; (b)Overflows

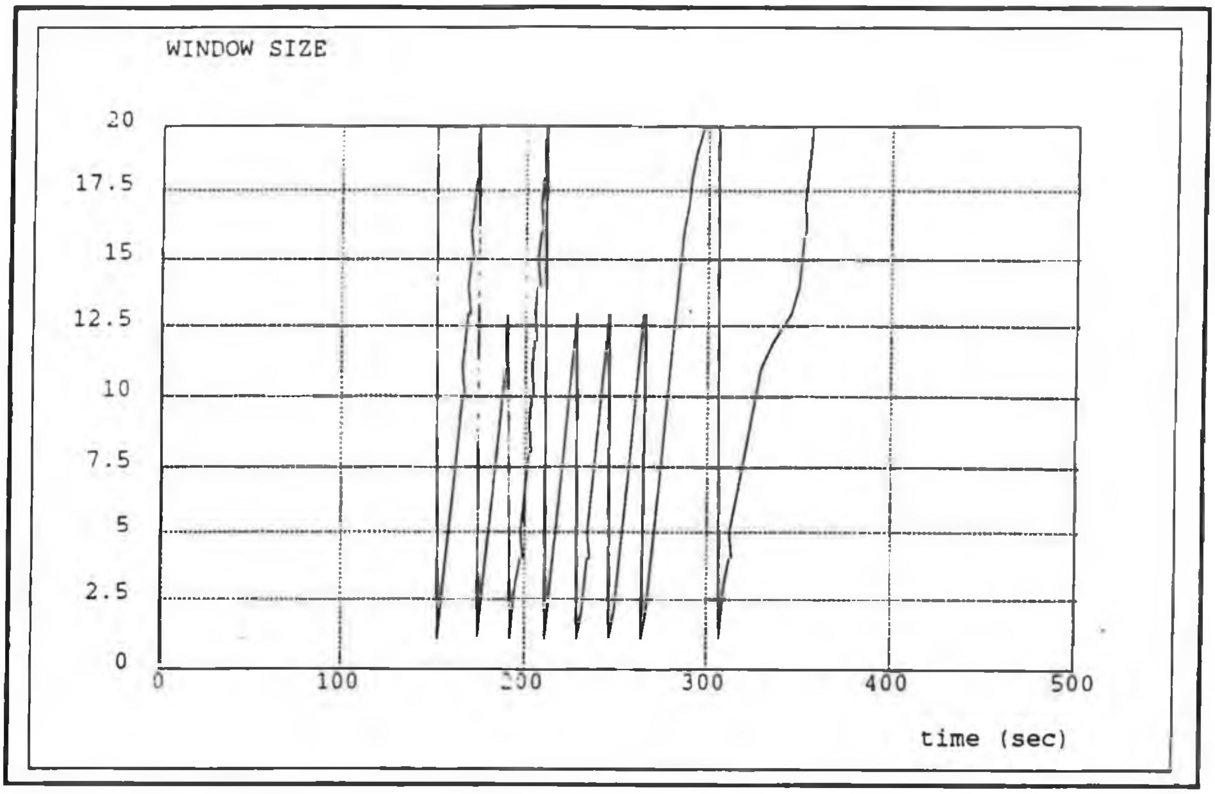


Fig. 6.5 **Dynamic Variation in Window Size**

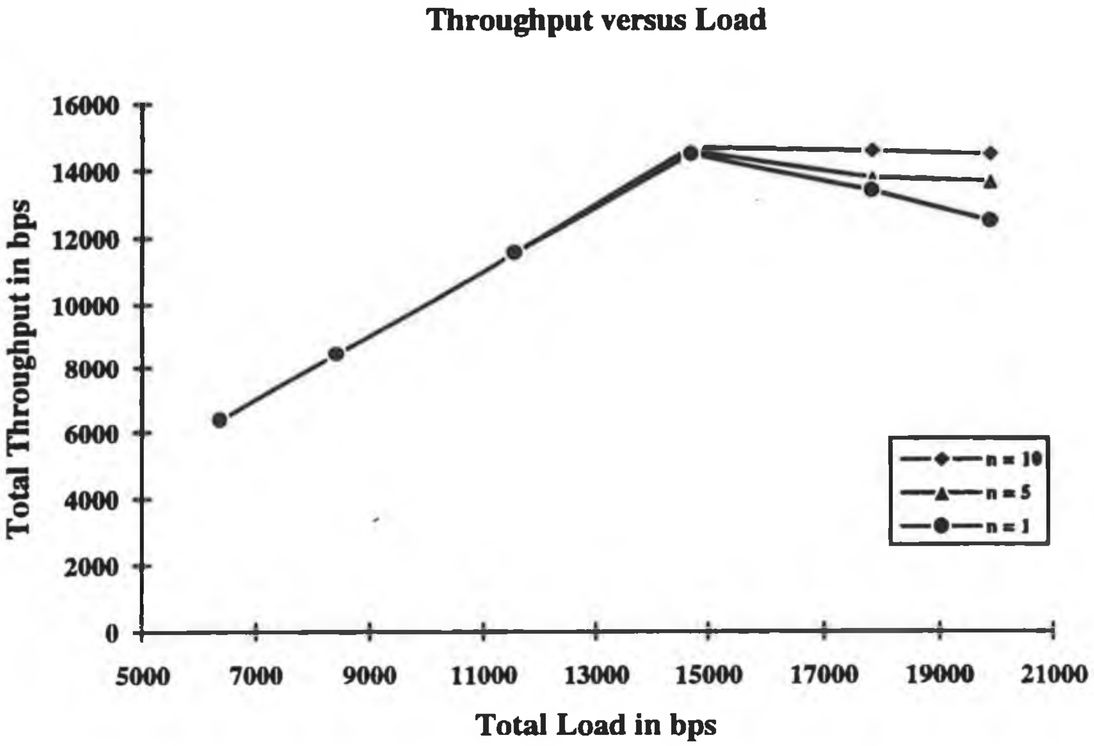


Fig. 6.6 **Effect of n on the Throughput versus Load Performance**

It is evident from this study that the dynamic windowing mechanism is an effective congestion control scheme, as it ensures that the network throughput is maintained at maximal levels. However, in all the cases, the window was increased by 1 for 10 successful transmissions. In other words, the parameter n was set to 10. The effectiveness of the congestion control scheme depends on the selection of the parameter n. Since the window is increased by 1 for every n successful transmissions, the throttling effect on the input traffic will last longer for higher values of n. The larger the value of n, the better the overload performance of the network. The results obtained by simulating the network for different values of n are presented in figs. 6.6 and 6.7

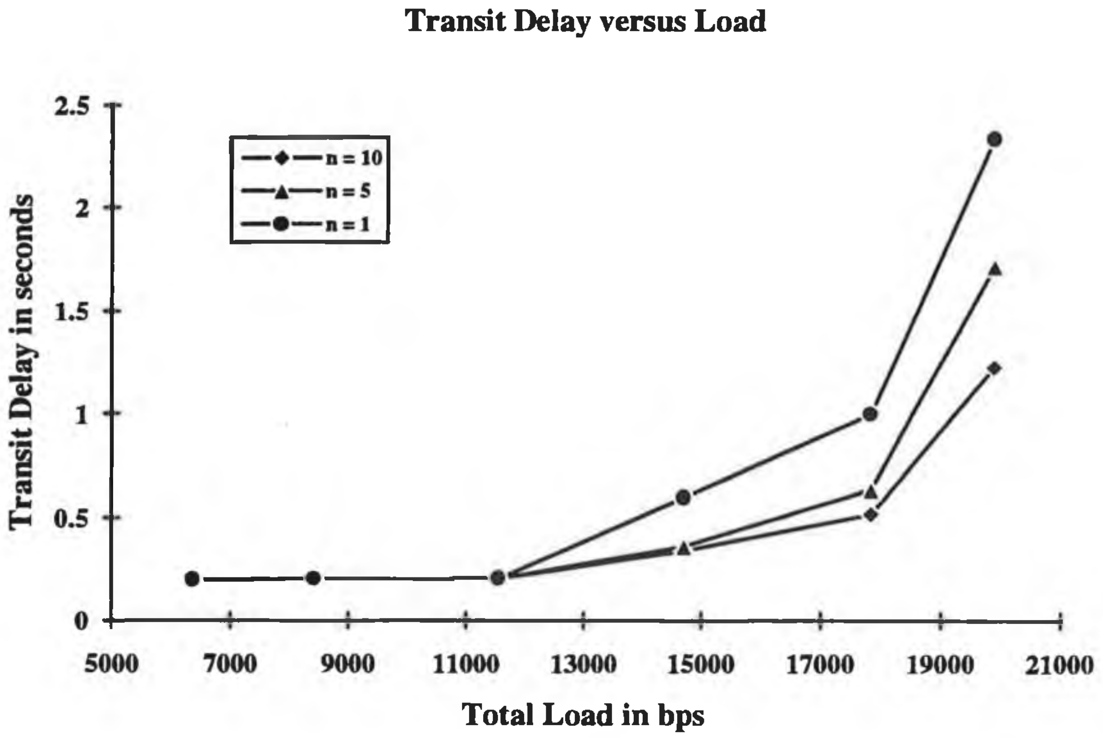


Fig.6.7 **Effect of n on the Transit Delay versus Load Performance**

The parabolic window increase technique was also used in some of the simulations. According to this scheme, if the window of a user is smaller than the maximum window size, it will open its window W by one for every W successful transmissions. A comparison of the two window increase techniques is illustrated in figs. 6.8 and 6.9.

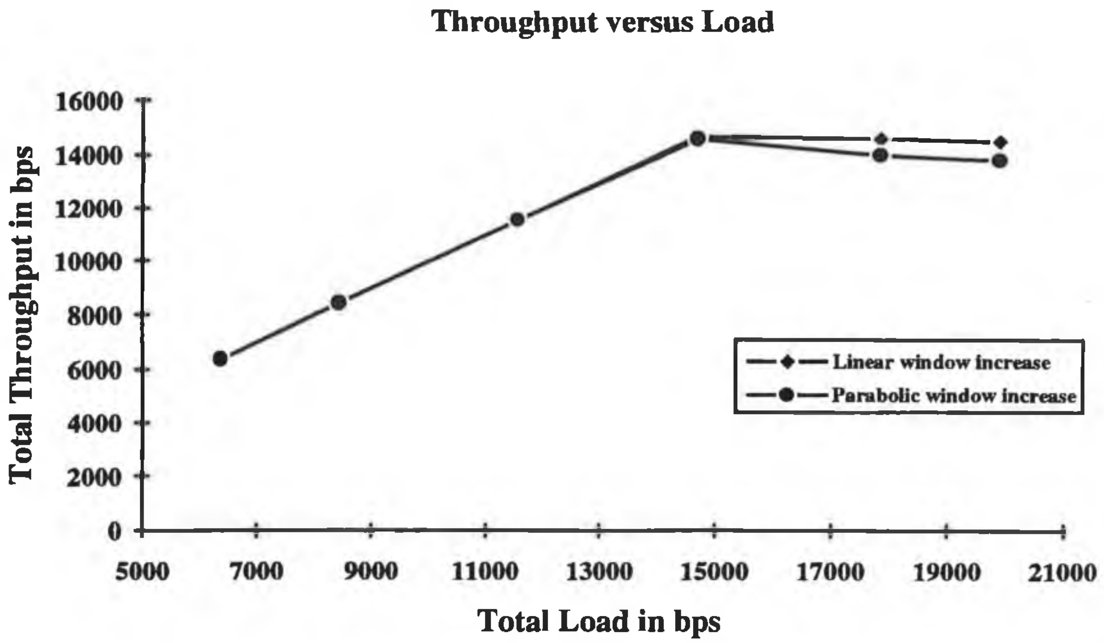


Fig. 6.8 **Effect of Window Increase Technique on the Throughput versus Load Performance**

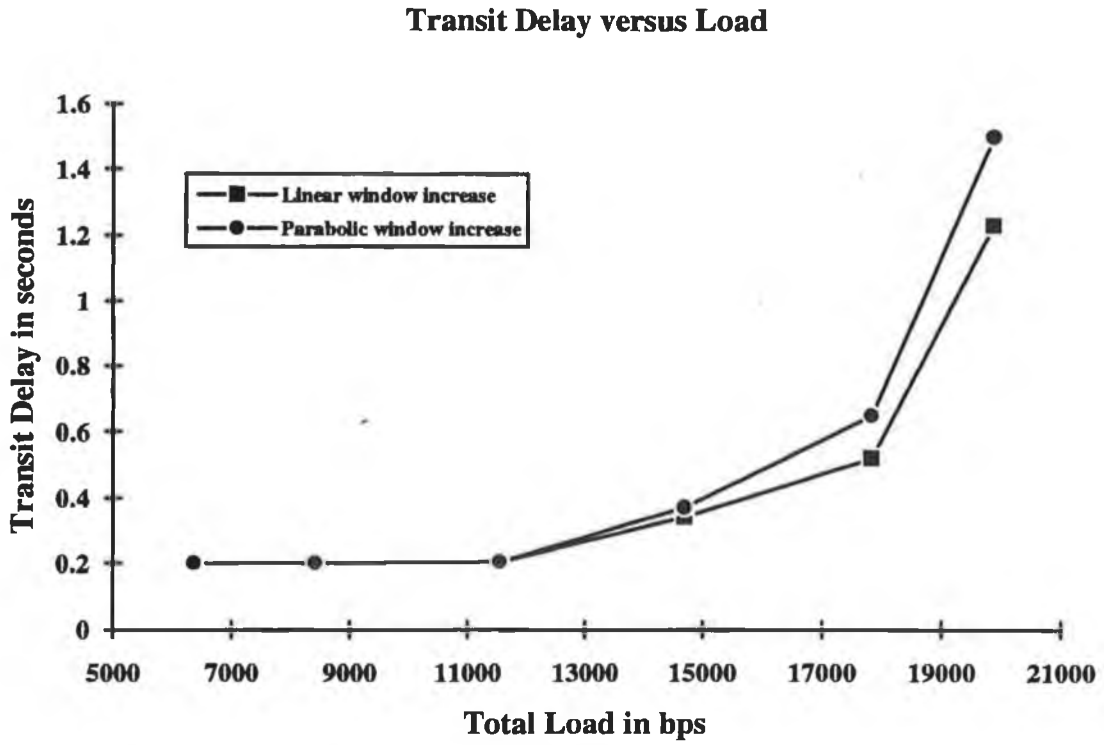


Fig. 6.9 **Effect of Window Increase Technique on the Transit Delay versus Load Performance**

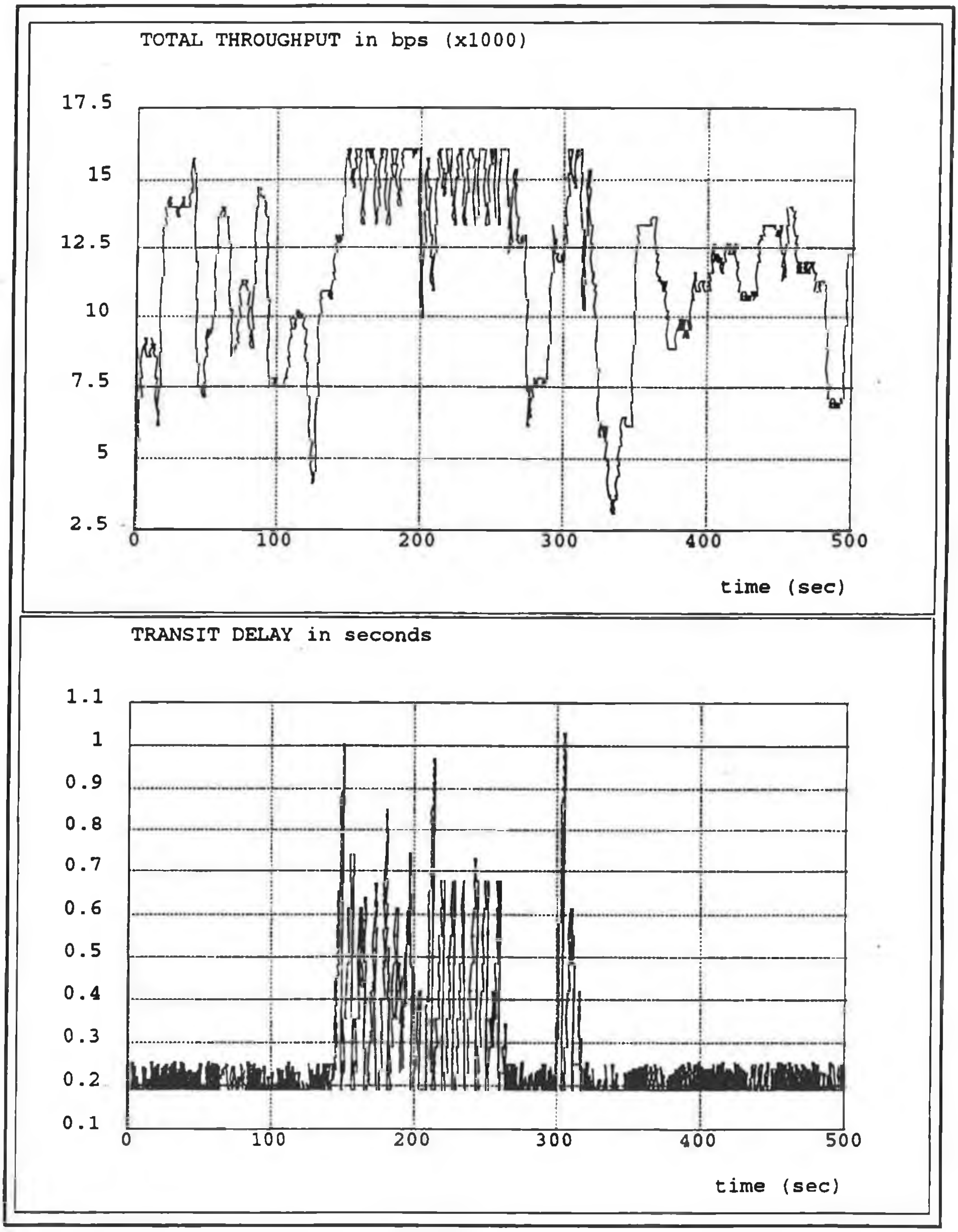


Fig. 6.10 **Performance with Congestion Notification & no sharing of buffers**
(a) Throughput versus time (b) Transit Delay versus time

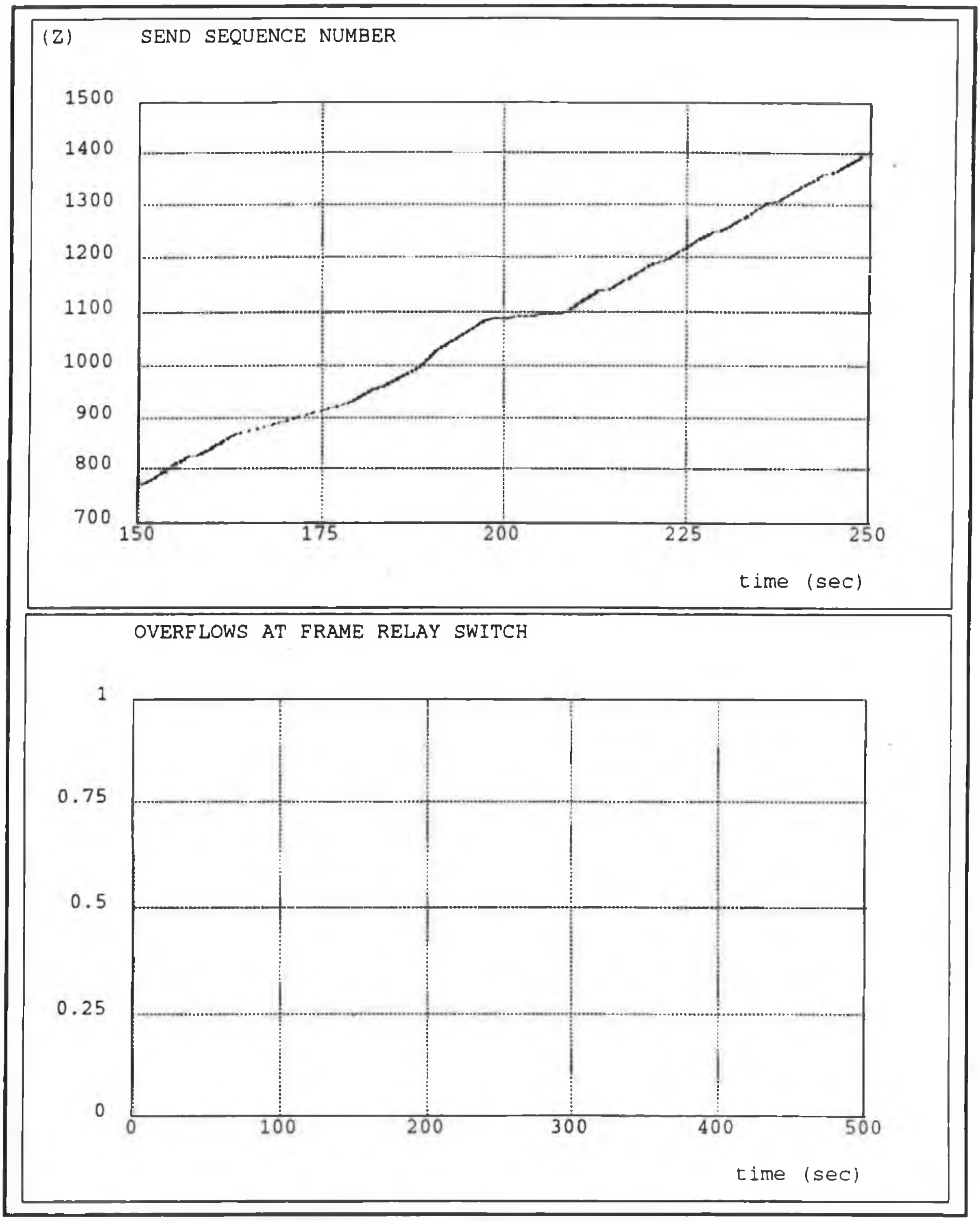


Fig. 6.11 **Performance with Congestion Notification & no sharing of buffers**
(a) Send Sequence Number (b) Overflows at Frame Relay Switch

In both the cases, the maximum size of the window was taken to be 20 and the minimum window size was 1. In the case of the linear dynamic windowing, the parameter N was taken to be 10. Based on these parameters, under both linear and parabolic-window mechanisms, the window is increased from 1 to the maximum after 190 positive acknowledgements. From the results obtained it is evident that the linear window mechanisms operate better than the parabolic one. With the parabolic window increase mechanism, the window increase is fast initially. As a result, the network does not get enough time to recover from the overload condition. Whereas in the case of the linear window mechanism, the network gets enough time to recover from congestion.

6.3 Explicit Detection Schemes

With this scheme, the network nodes monitor the usage of network resources, such as transmit buffers continuously. When a frame on arrival finds the buffer to be above a threshold, congestion status information can be conveyed by setting the FECN bit in the frame. When the frame arrives at its destination, the peer-user responds by setting the BECN bit of the acknowledgement or Information frames travelling in the reverse direction. Alternatively, the network node can notify the sending user of congestion by setting the BECN bit in the frames going in the reverse direction. However, if the buffers are full, the network nodes discard the frames.

In the end system, the LAPD protocol is enhanced to relay congestion advisory messages to the sending end system. The messages can then result in load shedding by the end system. The "adaptive windowing" scheme was used for this purpose. When the end systems receive frames with the BECN bit set to 1, they respond by reducing their windows to 1 and opening it by 1 for every n successful transmissions.

This scheme was implemented with two different buffer management strategies:

(1) With the "no sharing" system, there is a complete partitioning of the buffer pool among the virtual circuit connections. When the node detects the tendency of a certain virtual connection towards overload, it reacts by taking the control actions mentioned earlier in this section. The results presented in fig. 6.10(a) and fig. 6.10(b) indicate the good throughput and delay performance of the system at all loads. Since the end users reduce their traffic before congestion actually occurs, no losses occur (fig. 6.11(b)), there is no retransmission traffic (fig. 6.11(a)) and the throughput of the system does not fall. Correspondingly, the transit delay does not rise to very high values.

(2) With the "optimal sharing" scheme, the buffer pool is shared as long as three is no congestion in the network. However, when the node registers a tendency towards

overload, it checks to see which virtual circuit is using more than its fair share of buffers. It then sets the FECN bit of a frame travelling on the same virtual circuit, or sets the BECN bit of the frame travelling in the reverse direction on the same virtual circuit. Figs. depict the throughput and delay performance of the system for the same offered traffic. This scheme is more efficient throughput wise as it allows for the more efficient utilisation of buffers at the intermediate nodes. The improvement in the throughput and transit delay performance of the system can be seen from figs. 6.13 and 6.14 respectively.

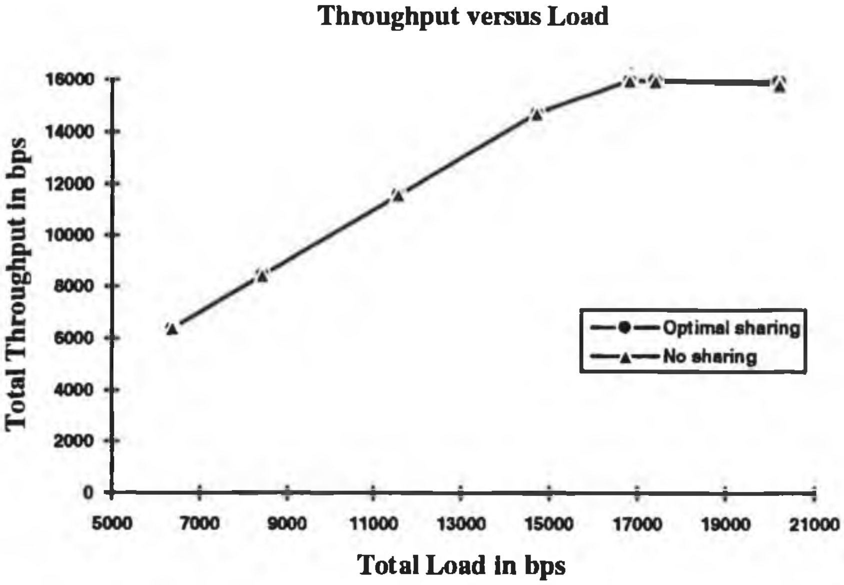


Fig. 6.13 **Throughput versus Load Performance with Congestion Notification; Effect of Buffer Management Strategy**

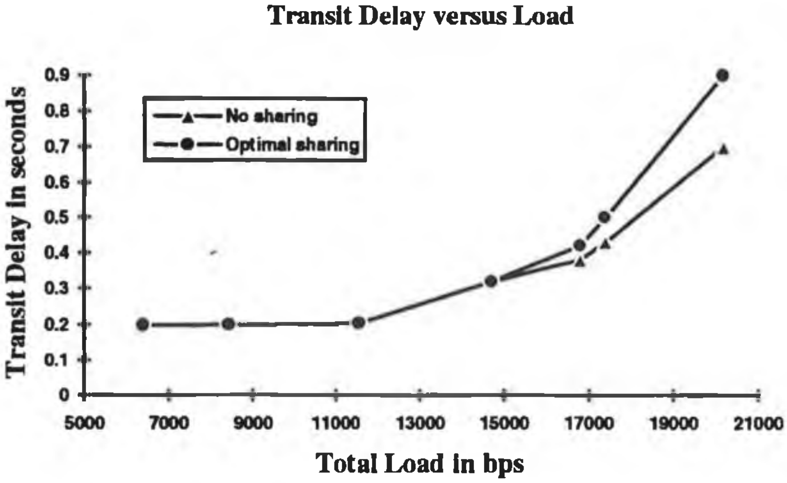


Fig. 6.14 **Transit Dealy versus Load Performance with Congestion Notification; Effect of Buffer Management Strategy**

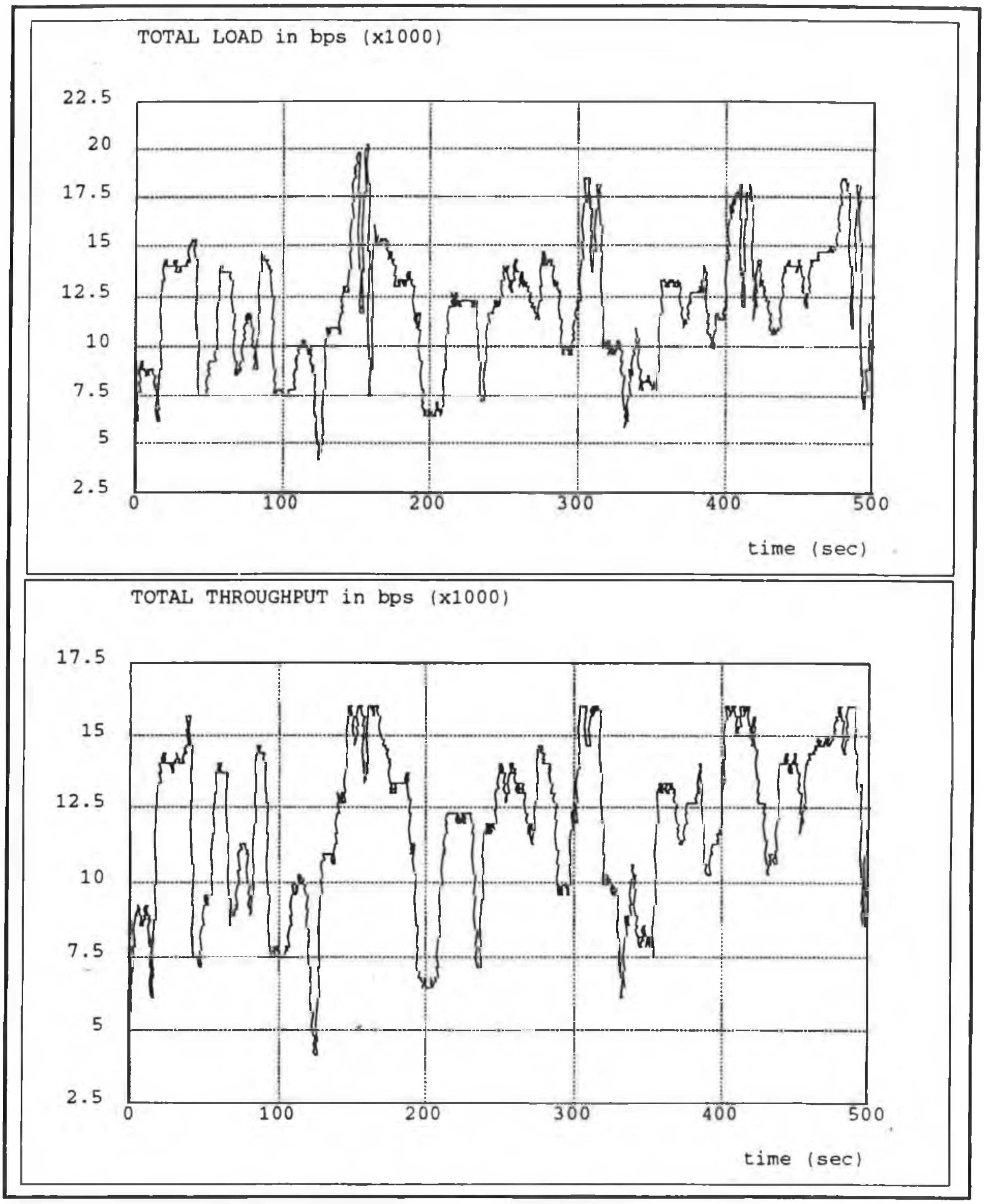


Fig. 6.15 **Performance of Model with Stop Duration**

(a) Load versus time; (b) Throughput versus time for offered load in (a)

6.4 Stop Duration

This scheme was used in conjunction with the congestion notification and adaptive windowing schemes. When the number of unprocessed I-frames are more than a certain threshold, the user realises that it is sending traffic at a rate faster than what the network can handle. It therefore responds by stopping the generation of traffic till this input queue clears up. In this way, the chances of a network entering congestion condition are reduced further. From fig. 6.15(a) we can see that the offered load does not rise above the channel capacity for long durations. Correspondingly, the throughput (fig. 6.15(b)) and the delay (fig. 6.16) performance is better for the system.

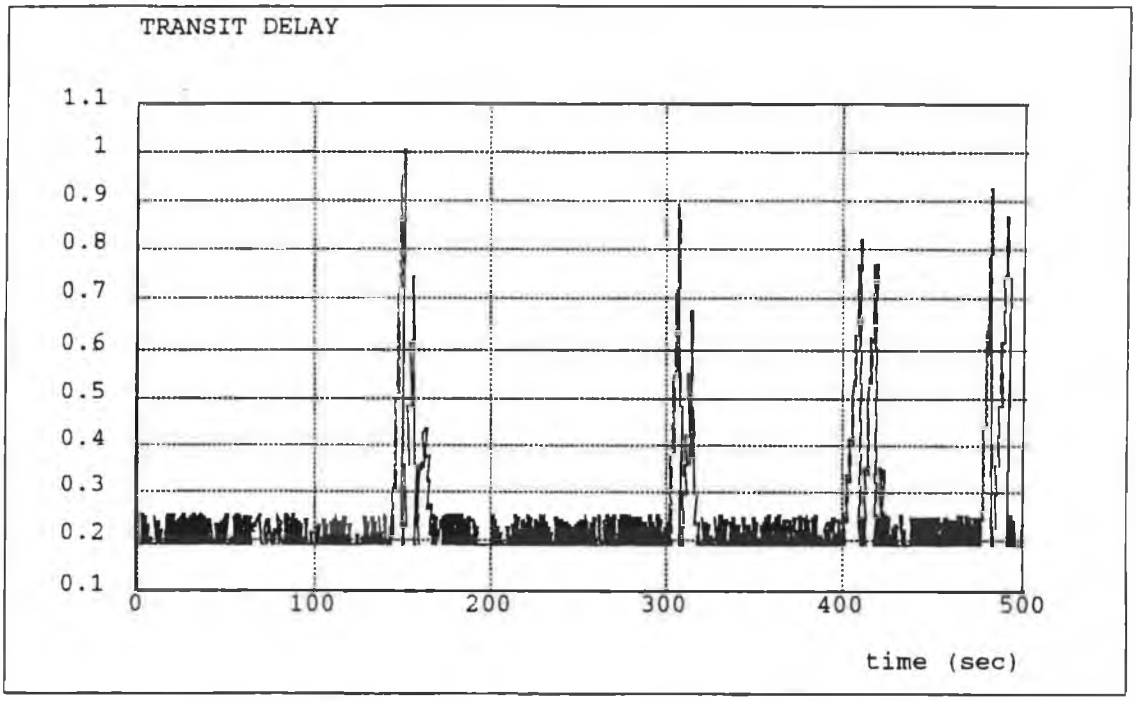


Fig. 6.16 **Transit Delay Performance with Stop Duration**

6.5 Conclusions

In frame relay networks where error recovery and flow control is end-to-end, congestion can severely degrade performance. In this chapter, it was demonstrated that the throughput of the system can be maintained at maximal levels by implementing a combination of "adaptive windowing", "congestion notification" and "stop duration" schemes. In addition, the delay can also be maintained at reasonable levels. These schemes were implemented by making minimal changes to the existing LAPD protocol implemented at the end systems

Chapter 7

Conclusions

7.1 Summary

The aim of this project has been to study the performance of frame relay networks and to implement various congestion control schemes to ensure that the network operates at peak efficiency at all times.

In frame relay networks, LAPD protocol is fully terminated at the end systems. However, the network nodes only perform the core LAPD functions. They are only responsible for detecting errors, switching and multiplexing. Error correction and flow control procedures are only implemented on an end-to-end basis. Consequently, the frame relay networks offer higher capacity and lower latency as compared to conventional X.25 networks. However, the absence of control capabilities at the network nodes makes these networks vulnerable to congestion. This is because the loss of a packet at any of the intermediate nodes in a network can cause the retransmission of window packets over all the links. This can further aggravate the condition of congestion in the network.

The first part of the study involved the simulation of frame relay networks with LAPD protocol implemented at the end systems. The window rotation technique was used to control the flow of data on the network. In addition, the protocol ensured the correct and sequential delivery of data to the end users. Upon detection of a frame loss, the end systems react by retransmitting up to a whole window of packets. A large number of simulations were carried out to study the performance of the networks at all loads. It was observed that a very small window can be overly restrictive in the absence of congestion and can prevent the network from operating at maximal efficiency. If the window is large enough, the sender can continuously transmit frames because the acknowledgements get back before the window is exhausted. A large window therefore ensures that the channel bandwidth is utilised efficiently and that the throughput is maintained at maximal levels. During congestion, the delays caused by higher buffer occupancy slow down window rotation to some extent, and this may result in a reduction in the network traffic. However, if the buffers at the frame relay nodes are not large enough to hold window worth of data for all the established virtual connections, frames can be dropped. This can result in the retransmission of up to a window's worth of data on each virtual circuit, thereby further aggravating congestion. Evidently, the performance of the system degrades more severely for larger window sizes in cases of congestion. This is because of the greater retransmission traffic generated with larger window sizes, leading to greater buffer utilisation, and greater overflow probability.

Although the performance of the network degrades in cases of congestion, it was observed that the adequate provisioning of network resources, and correct estimation of the timeout value can ensure a certain quality of network performance. If the buffer is large enough to hold window worth of data for all the virtual circuits, there would never be any packet loss. In addition, the increase in delays during congestion condition would reduce the window rotation to some extent, resulting in a reduced offered load. However, due to the bursty nature of network traffic, it is not a cost effective option to have full buffer dedication as the buffer size would depend on the size of the window and the maximum number of virtual circuits. In addition, the delays induced by a large buffer during congestion may violate certain delay service objectives.

By simulating the LAPD frame relay networks, it is evident that the lack of control capabilities in the network nodes makes these networks vulnerable to congestion. The second part of the study involved the study and implementation of alternative congestion controls. Adaptive windowing, congestion notification and stop duration were the three congestion control mechanisms used.

With the adaptive windowing scheme, the end system assumes the existence of congestion when a reject is received or when the Layer 2 timer times out, and responds by reducing its window to shed load. When enough consecutive frames are successfully acknowledged, it is believed that the congestion is over. The user responds by opening its window for every so many successful acknowledgements, thereby offering greater input traffic. A number of adaptive windowing schemes were studied. It was noted that the overload performance of the network is the best when a user reduces its window to 1 in case of congestion and increases it by 1 for every n successful transmissions.

The congestion notification scheme was implemented by using a two bit additional field in the header of the data packet. When the user initially transmits a packet, it clears the FECN bit. This bit is ignored by any network node that is not congested. However, if a network node detects the onset of congestion, it sets the FECN bit of a data packet that is flowing in the forward direction. When the data packet reaches the destination, the

BECN bit is set in the header of the acknowledgement packet, which is then transmitted from the destination to the source. When a user receives a packet with the BECN bit set, it responds by reducing its window. With this scheme, overflows were avoided and excellent throughput characteristics were obtained.

The network nodes continuously monitor the usage of the buffers, and the onset of congestion is detected when the buffers exceed a certain threshold. It was observed that the channel bandwidth is utilised most efficiently with the optimal sharing buffer management strategy.

With the stop duration scheme, the Layer3-Layer2 interface at the end system stops the generation of data when it detects a queue of window unprocessed I frames. Since the end system realises that it is generating traffic at a rate faster than what the network can service, it responds by stopping the generation of data till the queue is cleared up. With this scheme, the offered load does not exceed channel capacities for periods long enough to cause congestion. Subsequently, the network throughput performance is maintained at maximal levels and the delay is maintained at reasonable values. This mechanism was used in conjunction with the congestion notification and the adaptive windowing schemes.

The performance of the network was maintained at maximal levels by implementing these schemes. In addition, these schemes do not interfere with the natural LAPD protocol in the absence of congestion. These techniques were implemented by making minimal changes to the protocol. Furthermore, the protocol at the network nodes ensures that inspite of "optimal sharing", the performance of the well behaved users is not affected by the misbehaving ones.

7.2 Conclusions

A substantial number of simulations were carried out and the performance of the system was studied in detail. The bursty generator model was useful in, at least to a certain extent, reproducing the observed phenomena of network traffic. A number of network models were developed and reasonable results were obtained for the developed systems.

A few supplementing studies can be carried out to further assess the performance of the system. Realistic data can probably be used to improve and interpret the modelling of this system. In addition, the performance of the model can be studied if new connections are established in the midst of the simulations.

The studied techniques can be implemented with relative ease in the existing protocol. These techniques are quite effective in maintaining the throughput of the network at maximal levels. In addition, the throughput is increased without experiencing very high delays. The frame relay technology is quite effective in meeting the increased demands of the users. If these techniques are incorporated in the existing protocol, the users can be guaranteed of high throughput and low latency at all loads.

References

1) M. Ifran Ali, "Frame Relay in Public Networks"; IEEE Communications Magazine, March 1992

2) Amit Bhargava, "*Integrated Broadband Networks*"; Artech House, Boston

3) Paul T. Brady, "Performance of an Edge-to-Edge Protocol in a Simulated X.25/X.75 Packet Network", IEEE Journal on Selected Areas in Communications, Vol. 6, No. 1, Jan 1988, pp 190-196

4) Werner Bux, Davide Grillo, "Flow Control in Local-Area Networks of Interconnected Token Rings"; IEEE Transactions on Communications, Vol.Com-33, No.10, October 1985

5) James P. Cavanagh, "Applying the Frame Relay Interface to Private Networks"; IEEE Communications Magazine, March 1992, pp 48-64

6) CCITT Recommendations Q.920-Q.921; Blue Book; Volume VI - Fascicle VI.10; Digital Subscriber Signalling System No. 1 (DSS 1), Data Link Layer; IXth Plenary Assembly, Melbourne, 14-25 November 1988

7) K.-J. Chen, B.T. Doshi, H.Q. Nguyen and K.M.Rege, "Performance of LAPD Frame-Relay Networks: Transmission Error Effects and Congestion Control"; in Proc. 12th Int. Teletraffic Congr., Torino, Italy, May 1988, pp 1100-1108

8) Kim-Joan Chen, Kiran M. Rege, "A Comparative Performance Study of Various Congestion Controls for ISDN Frame-Relay Networks"; in Proc. IEEE INFOCOM'89, Ottawa, Canada, April 1989

9) Kim-Joan Chen, Kelvin K. Y. Ho, and Vikram R. Saxena, "Analysis and Design of a Highly Reliable Transport Architecture for ISDN Frame Relay Networks"; IEEE journal on Selected Areas in Communications, Vol. 7, No. 8, October 1989, pp1231-1242

10) Ericsson, "*Telecommunications-Telephone Networks 2*", Chartwell Bratt Ltd., Sweden, 1987

11) Mario Gerla and Leonard Kleinrock, "Flow Control: A Comparative Survey"; IEEE Transactions on Communications, Vol. COM-28, No. 4, April 1980, pp 553-574

12) Joseph L. Hammond, Peter J.P.O'Reilly, "*Performance Analysis of Local Computer Networks*"; Addison-Wesley Publishing Company, USA

13) Fred Jennings, "*Practical Data Communications - Modems, Networks and Protocols*"; Blackwell Scientific Publications

14) John Lane, "*Exploiting Integrated Voice and Data Networks*", NCC Publications

15) Kajsa Lundfall, "Frame Relay- for faster and more efficient data communications"; Ericsson review no. 1-2, 1992.

16) Dennis MacKinnon, William McCrum, Donald Sheppard, "*An Introduction to Open Systems Interconnection*"; Computer Science Press

17) Mehdi Nassehi, "Window Flow Control in Frame Relay Networks"; IEEE 1988, pp1784-1790

18) Amelia Platt and Michael J. Morse, "Traffic Management in Frame Relay Networks"; Computer Networks and ISDN Systems, 23, 1992, pp 305-316

19) Neil Rickard, "Frame Relay: The best of both worlds?"; Communications International, February 1992, pp 46-48

20) Bob Ryan, "On the Fast Track "; Byte, November 1991, pp 361-364

21) Mischa Schwartz; "*Telecommunication Networks - Protocols, Modeling and Analysis*"; Addison-Wesley Publishing Company, USA, 1987

22) Michel Smouts, "*Packet Switching Evolution - from Narrowband to Broadband ISDN*"; Artech House, Boston

23) D.A. Spencer, J.O. Dimmick, F.M. Burg, J.C. Kaufeld, "ISDN Packet-mode Protocol - Architecture and Services", Presented at Telecom'87, Geneva, October 1987

24) William Stallings, "*ISDN - An Introduction*"; Macmillan Publishing Company, New York, 1989

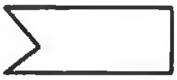
25) Andrew S. Tanenbaum, "*Computer Networks* "; Prentice Hall of India Private Ltd., New Delhi, October 1989

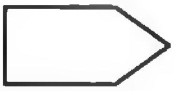
Appendix A

SDL Representation of the Developed Model

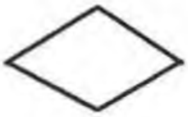
This appendix provides the SDL representation of the data link procedures of the developed model. This model is different from the standard LAPD protocol (as defined in Recommendation Q.920-Q.921) in that it incorporates three congestion control schemes - Adaptive windowing, Congestion Notification, and Stop Duration. The following symbols and abbreviations are used within this description.

a) State

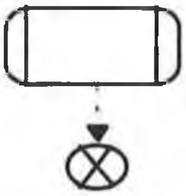
b)  Signal reception

c)  Signal generation

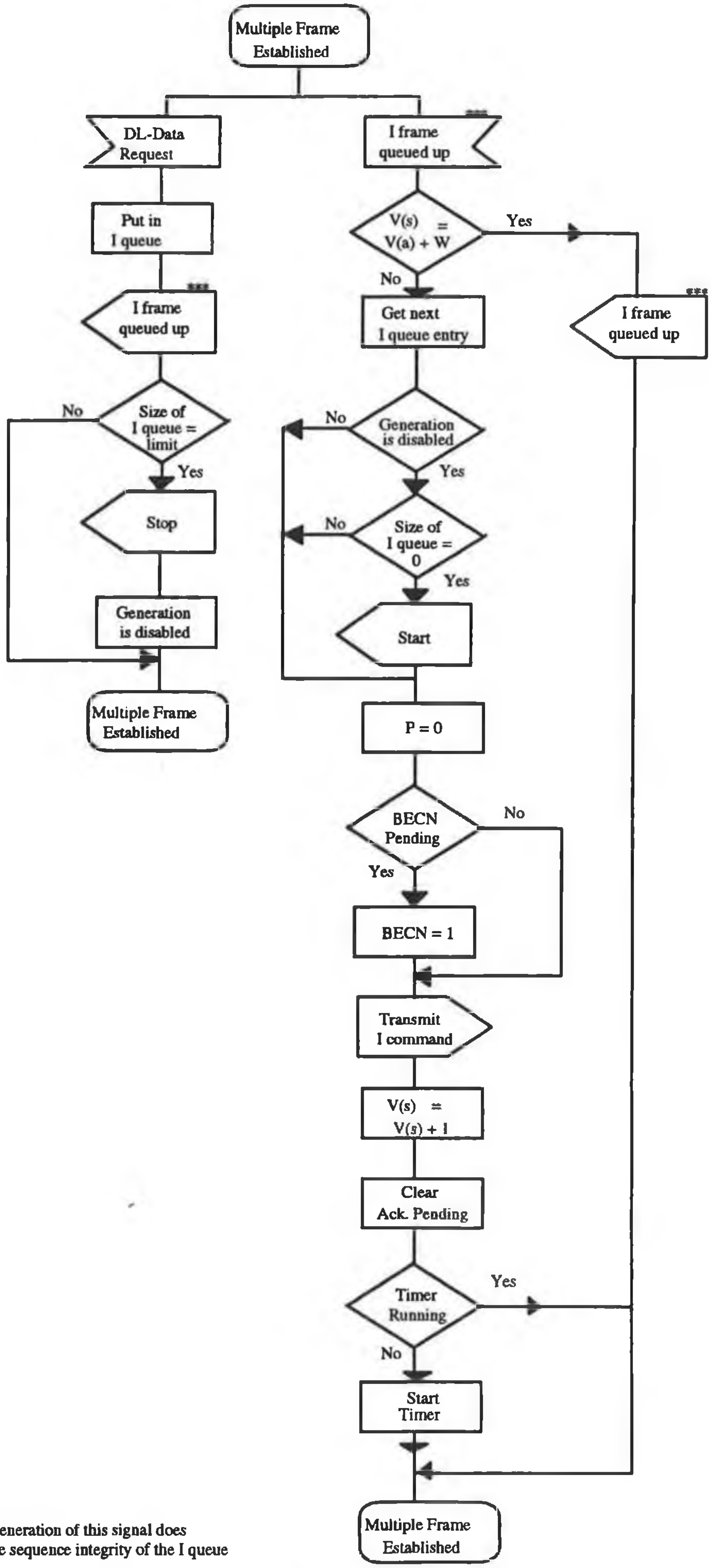
d) Process description

e)  Test

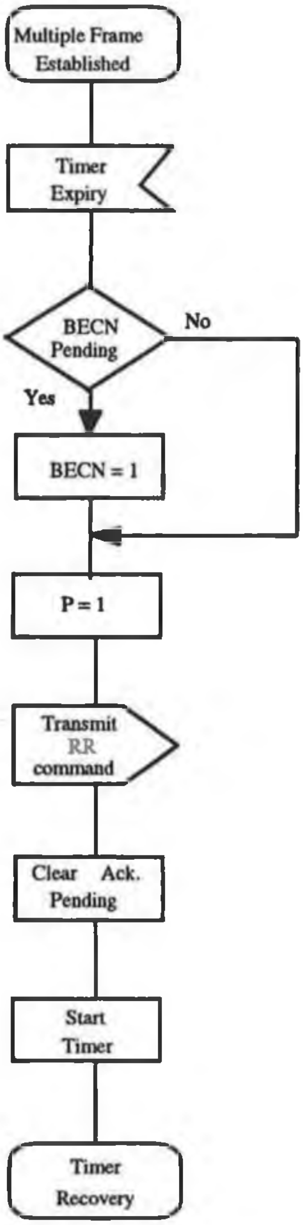
f) Procedure call

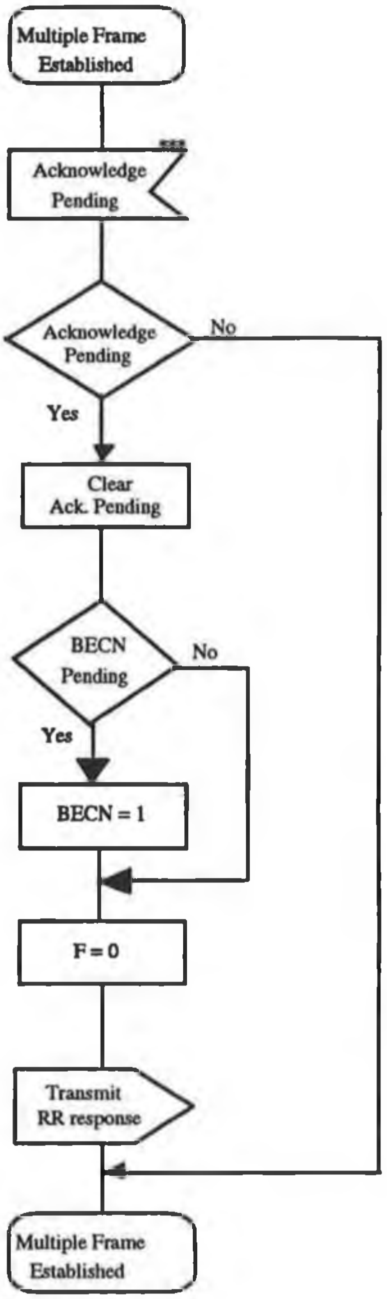
g)  Procedure definition

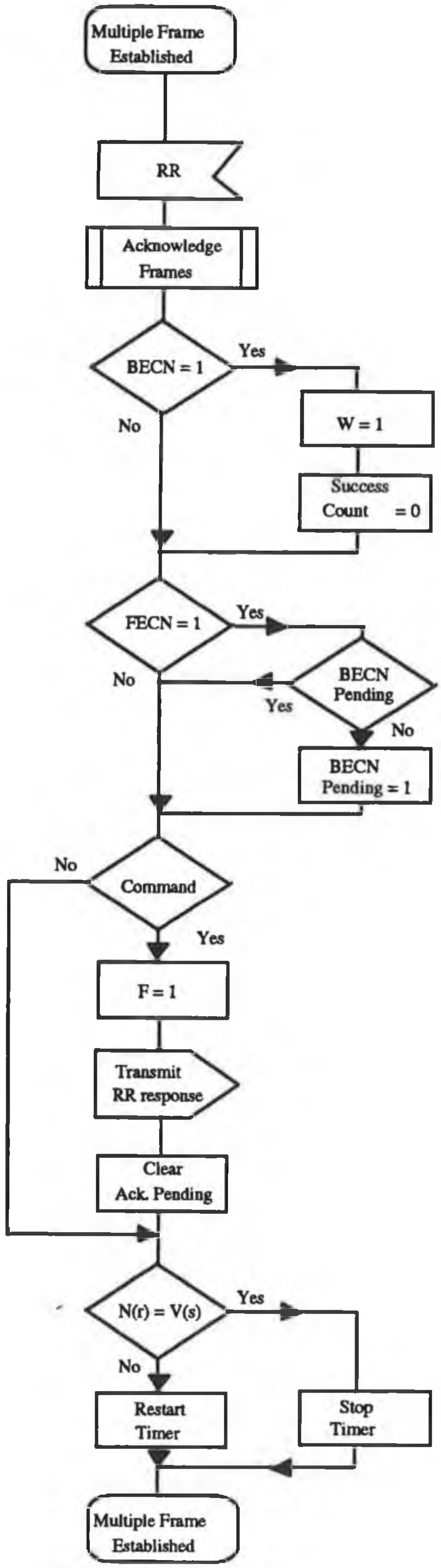
h) *** To mark an event or signal required as a result of the representation approach adopted, which is local to the data link layer entity

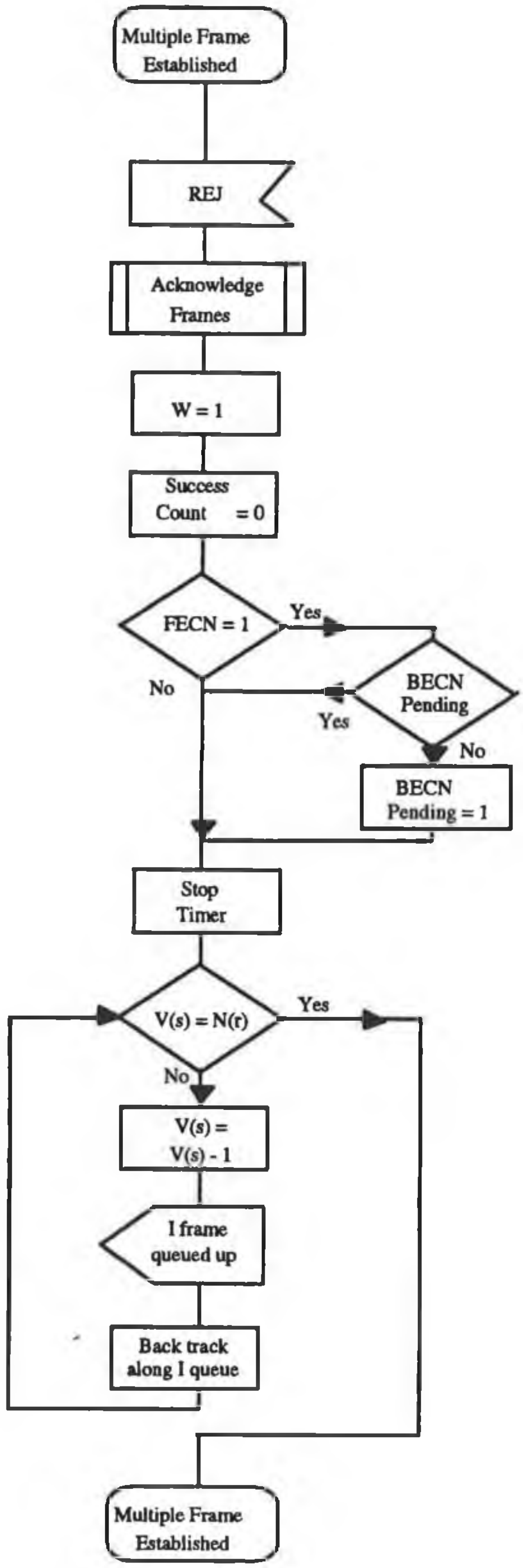


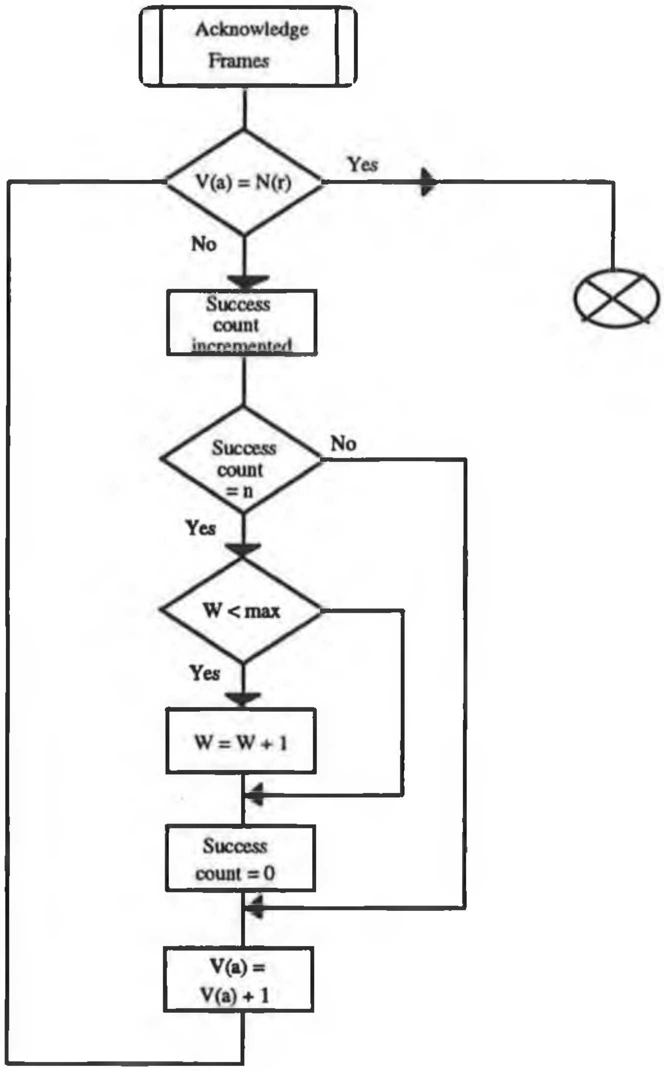
Note- The regeneration of this signal does
not affect the sequence integrity of the I queue

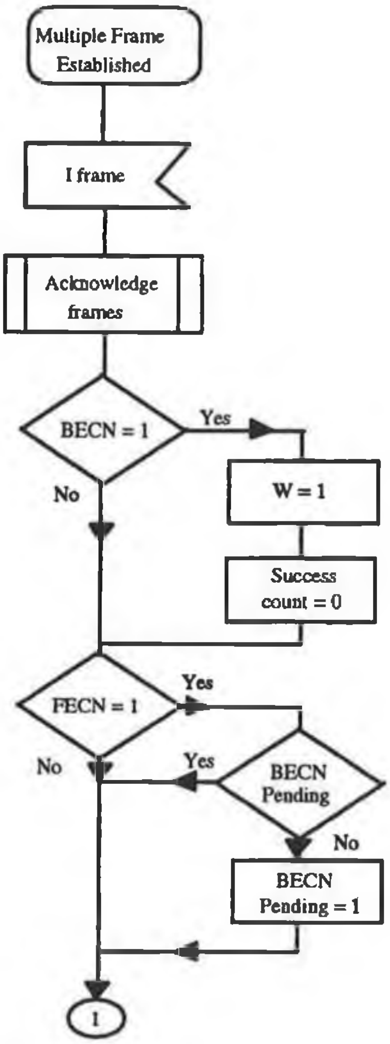

Multiple Frame Established
Timer Expiry
BECN Pending
No
Yes
BECN = 1
P = 1
Transmit RR command
Clear Ack. Pending
Start Timer
Timer Recovery

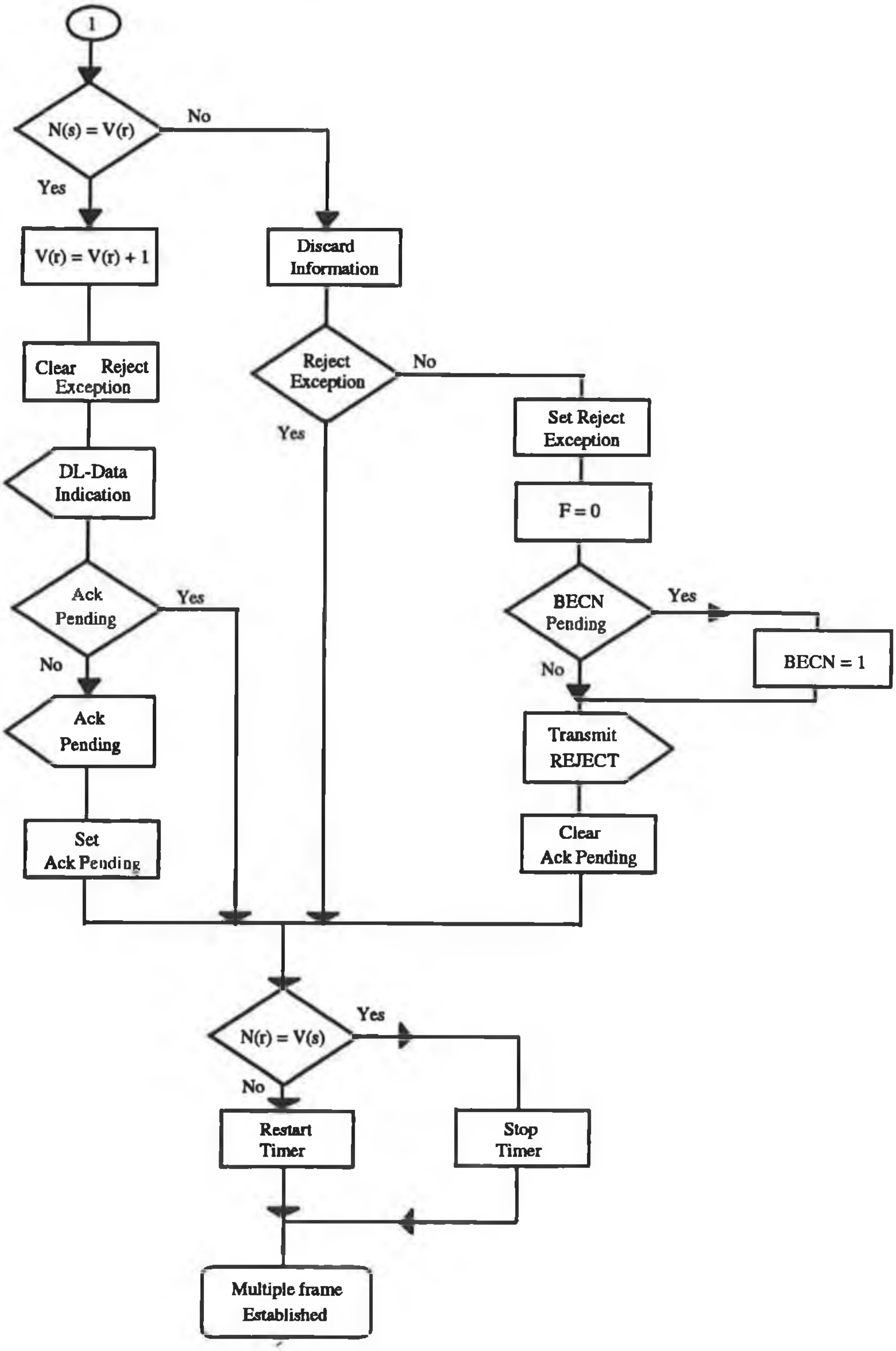

Multiple Frame Established
Acknowledge Pending
Acknowledge Pending
No
Yes
Clear Ack. Pending
BECN Pending
No
Yes
BECN = 1
F = 0
Transmit RR response
Multiple Frame Established

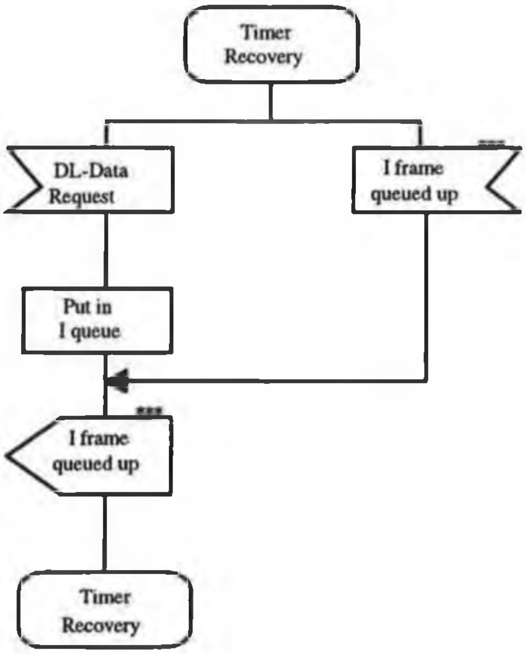

Multiple Frame Established
RR
Acknowledge Frames
BECN = 1
Yes
No
W = 1
Success Count = 0
FECN = 1
Yes
No
BECN Pending
Yes
No
BECN Pending = 1
No
Command
Yes
F = 1
Transmit RR response
Clear Ack. Pending
N(r) = V(s)
Yes
No
Restart Timer
Stop Timer
Multiple Frame Established

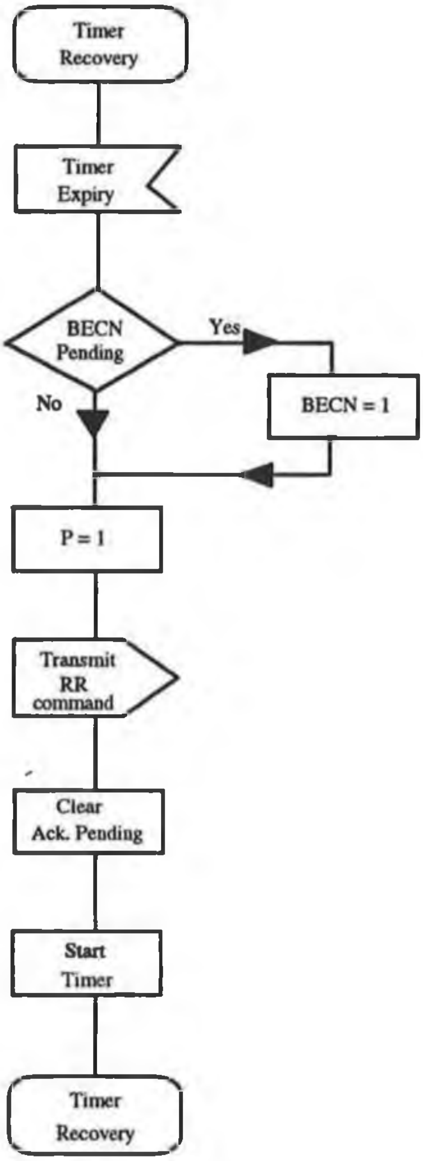

Multiple Frame Established
REJ
Acknowledge Frames
W = 1
Success Count = 0
FECN = 1
Yes
No
BECN Pending
Yes
No
BECN Pending = 1
Stop Timer
V(s) = N(r)
Yes
No
V(s) = V(s) - 1
I frame queued up
Back track along I queue
Multiple Frame Established

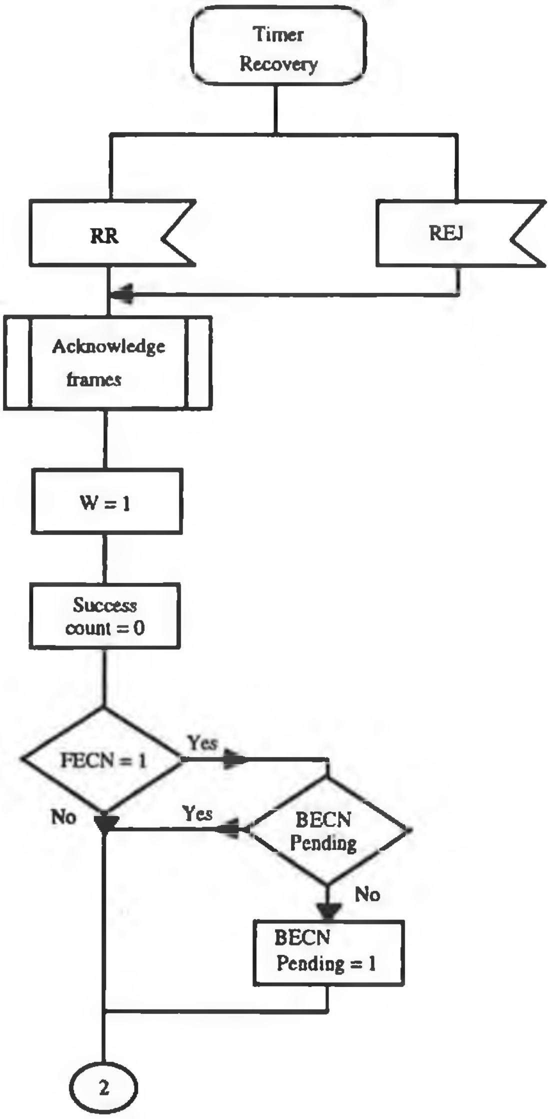

Acknowledge Frames
V(a) = N(r)
Yes
No
Success count incremented
Success count = n
No
Yes
W < max
Yes
W = W + 1
Success count = 0
V(a) = V(a) + 1

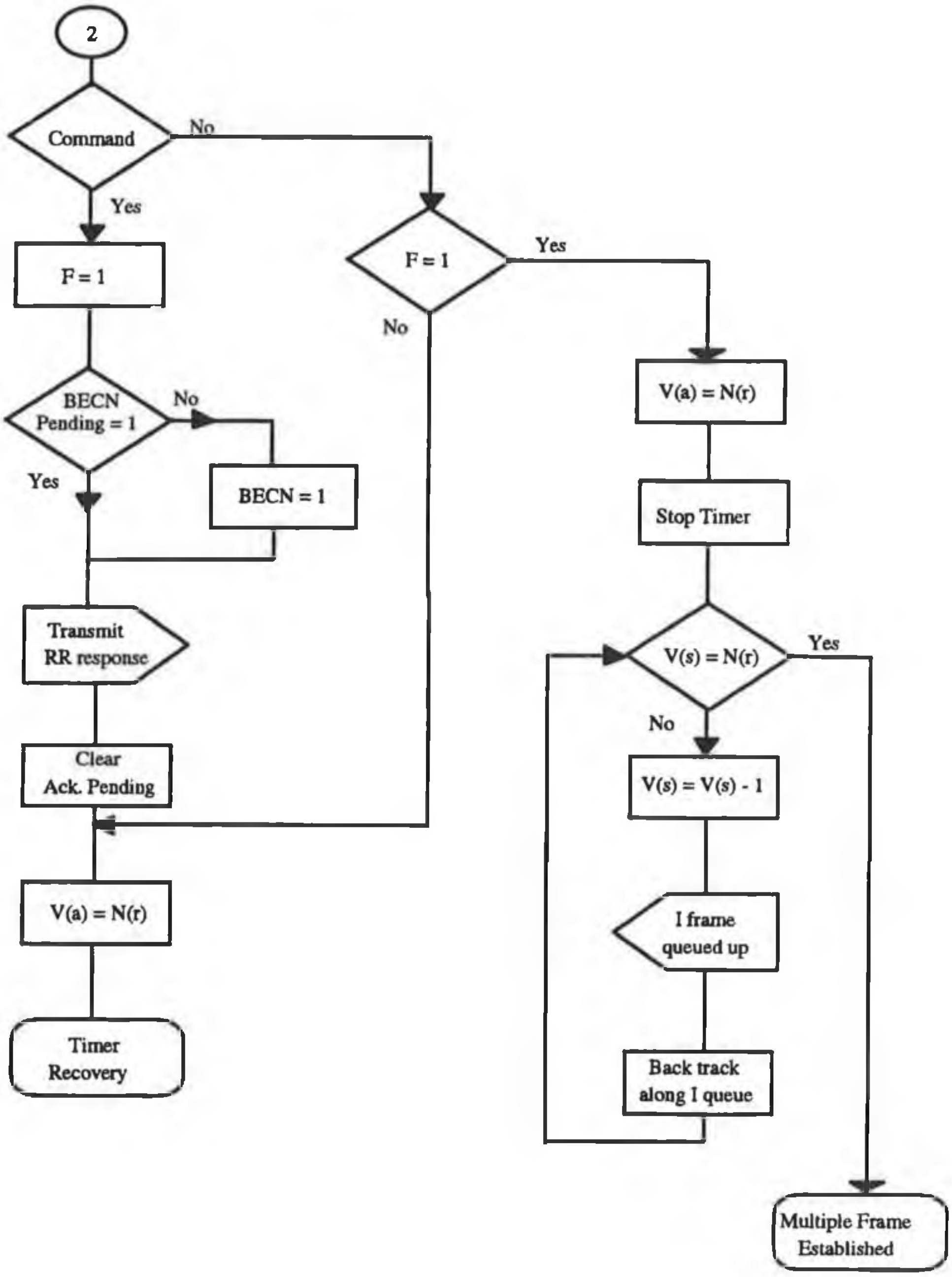

Multiple Frame Established
I frame
Acknowledge frames
BECN = 1
Yes
No
W = 1
Success count = 0
FECN = 1
Yes
No
Yes
BECN Pending
No
BECN Pending = 1
1

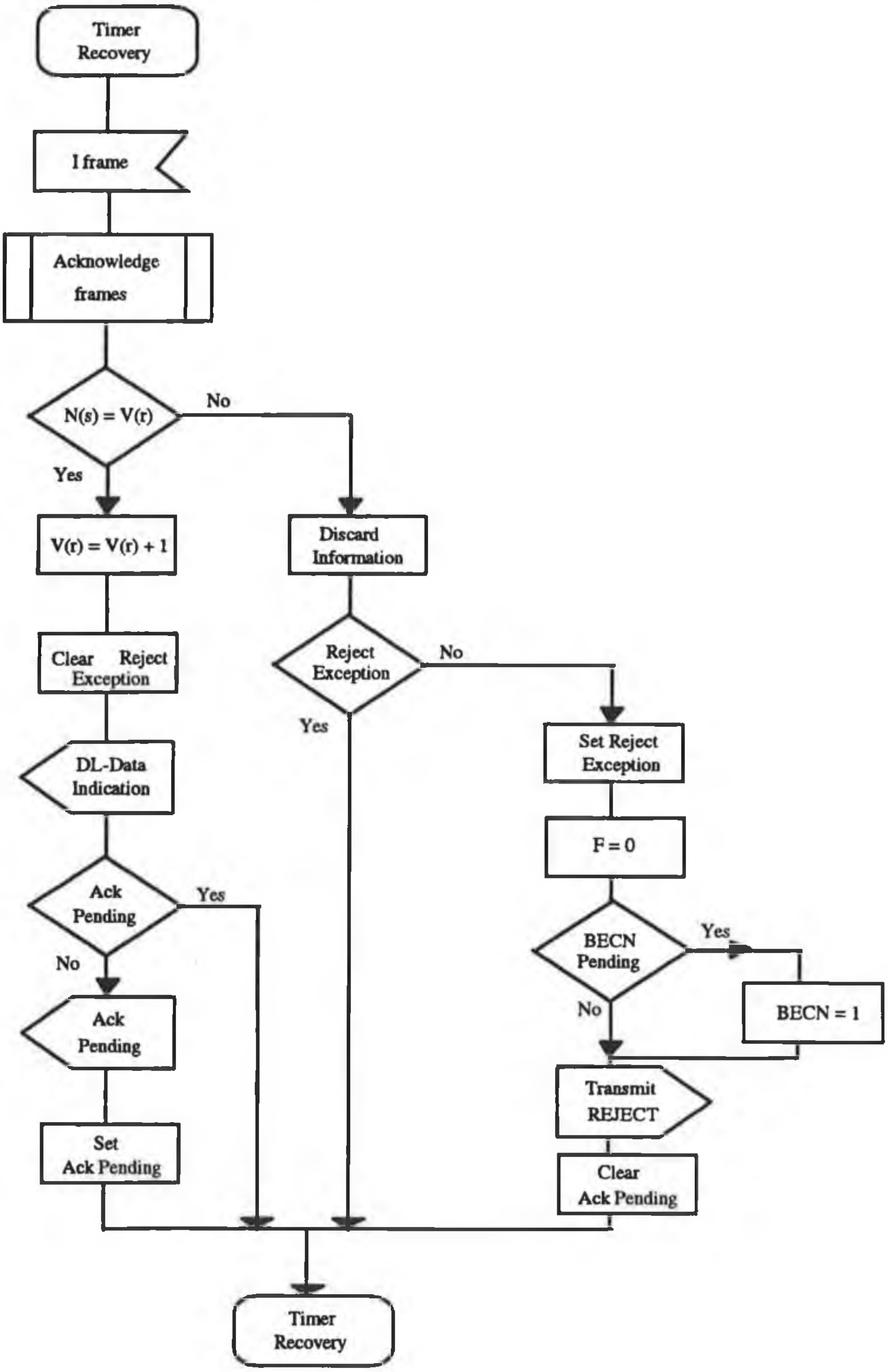

1
N(s) = V(r)
No
Yes
V(r) = V(r) + 1
Discard Information
Clear Reject Exception
Reject Exception
No
Yes
Set Reject Exception
DL-Data Indication
F = 0
Ack Pending
Yes
No
BECN Pending
Yes
No
BECN = 1
Ack Pending
Transmit REJECT
Set Ack Pending
Clear Ack Pending
N(r) = V(s)
Yes
No
Restart Timer
Stop Timer
Multiple frame Established

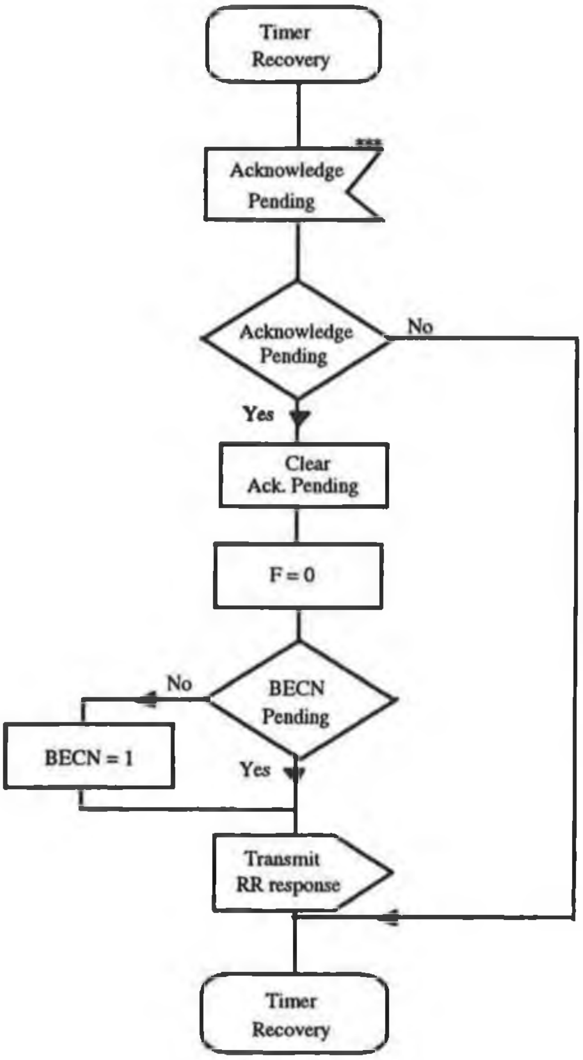

Timer Recovery
DL-Data Request
I frame queued up
***
Put in I queue
***
I frame queued up
Timer Recovery


Timer Recovery
Timer Expiry
BECN Pending
Yes
No
BECN = 1
P = 1
Transmit RR command
Clear Ack. Pending
Start Timer
Timer Recovery


Timer Recovery
RR
REJ
Acknowledge frames
W = 1
Success count = 0
FECN = 1
Yes
No
Yes
BECN Pending
No
BECN Pending = 1
2


2
Command
No
Yes
F = 1
F = 1
Yes
No
BECN Pending = 1
No
Yes
BECN = 1
V(a) = N(r)
Stop Timer
Transmit RR response
V(s) = N(r)
Yes
No
Clear Ack. Pending
V(s) = V(s) - 1
V(a) = N(r)
I frame queued up
Timer Recovery
Back track along I queue
Multiple Frame Established


Timer Recovery
I frame
Acknowledge frames
N(s) = V(r)
No
Yes
V(r) = V(r) + 1
Discard Information
Clear Reject Exception
Reject Exception
No
Yes
Set Reject Exception
DL-Data Indication
F = 0
Ack Pending
Yes
No
BECN Pending
Yes
No
BECN = 1
Ack Pending
Transmit REJECT
Set Ack Pending
Clear Ack Pending
Timer Recovery


Timer Recovery
Acknowledge Pending
Acknowledge Pending
No
Yes
Clear Ack. Pending
F = 0
BECN Pending
No
Yes
BECN = 1
Transmit RR response
Timer Recovery

Appendix B

List of Acronyms

AAL	ATM Adaptation Layer
ARQ	Automatic Repeat reQuest
ATM	Asynchronous Transfer Mode
BA	Basic Access
BECN	Backward Error Congestion Notification
BER	Bit Error Ratio
CAD	Computer Aided Design
CCITT	International Telegraph and Telephone Consultative Committee
CH	Congestion Handling
C/R	Command/Response
DCE	Data Circuit terminating Equipment
DE	Discard Eligibility
DLC	Data Link Connection
DLCI	Data Link Connection Identifier
DTE	Data Terminal Equipment
EA	Extended Address
ET	Exchange Termination
FC	Flow Control
FCFS	First Come First Served
FCS	Frame Check Sum
FDDI	Fiber Distributed Data Interface
FECN	Forward Error Congestion Notification
FR	Frame Relay
FSM	Finite State Machine
HDLC	High Level Data Link Control
IDN	Integrated Digital Networks
IP	Interworking Port
ISDN	Integrated Services Digital Networks
ISO	International Organisation for Standardisation
LAN	Local Area Network
LAPB	Link Access Procedure, B channel
LAPD	Link Access Procedure, D channel
LT	Line Termination
MAN	Metropolitan Area Networks
NT	Network Termination

NT1	Network Termination 1
NT2	Network Termination 2
OPNET	Optimised Network Engineering Tool
OSI	Open Systems Interconnection
PABX	Private Automated Branch Exchange
PBX	Private Branch Exchange
PC	Personal Computer
pdf	probability distribution function
PH	Packet Handler
PL	Packet Loss treatment
PLLC	Permanent Logical Link Connection
PSPDN	Packet Switched Public Data Networks
PTT	Post, Telegraph and Telephone
REJ	REJect
RNR	Receiver Not Ready
RR	Receiver Ready
RT	Routing
SAPI	Service Access Point Identifier
SDLC	Synchronous Data Link Control
SNA	Systems Network Architecture
TA	Terminal Adaptor
TCP/IP	Transmission Control Protocol/Internet Protocol
TE1	Terminal Equipment 1
TE2	Terminal Equipment 2
TEI	Terminal Endpoint Identifier
UNI	User Network Interface