IBM

International Technical Support Centers

High-Speed Networking Technology
An Introductory Survey

GG24-3816-00

High Speed Networking Technology
An Introductory Survey

Document Number GG24-3816-00

March 1992

International Technical Support Center
Raleigh

Take Note!

Before using this information and the product it supports, be sure to read the general information under "Special Notices" on page xv.

First Edition (March 1992)

This edition applies to Version 2.2 of the IBM Token-Ring Network Bridge Program (Product Number 53F7724) and Version 1.0 of the IBM PC Network Bridge Program (Product Number 96X5860).

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for reader's comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation, International Technical Support Center
Dept. 985, Building 657
12195
Research Triangle Park, NC 27709

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Abstract

Digitisation of public communications networks, the wide use of optical fibre technology and continuing advances in circuit technology have combined to produce very significant increases in speeds and decreases in cost of communications services. But in order to deliver practical benefits to the user new techniques and technologies are needed.

This publication presents a broad overview on the emerging technology of very high speed communications. It is written at the "technical conceptual" level with some areas of greater detail. It is intended to be read by computer professionals who have some understanding of communications (but who do not necessarily consider themselves experts).

The primary topics of the book are:

- Fibre Optical Technology
- Local Area Networks (Token-Ring, FDDI, MetaRing, CRMA)
- Metropolitan Area Networks (DQDB/SMDS)
- High Speed Packet Switches (Frame Relay, Paris)
- Networking Protocols for High Speed

CC CO EN ES SD VO (324 pages)

Contents

Figures

Special Notices

This publication is intended to help both customers and IBM systems engineers to understand the principles of high speed communications. The information in this publication is not intended as the specification of any programming interface provided by any IBM product.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

The following terms, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM
CallPath
ESCON
IBM
Micro Channel
VTAM

The following terms, which are denoted by a double asterisk (* *) in this publication, are trademarks of other companies.

Ethernet is a trademark of XEROX Corporation.
QPSX is a trademark of QPSX Communications, Inc.

Microsoft is a trademark of Microsoft Corporation.
UNIX is a trademark of UNIX System Laboratories, Inc.
DECnet is a trademark of Digital Equipment Corporation.
SNS SNA/Gateway is a trademark of Interlink Computer Sciences, Inc.

Preface

This publication is a systems engineering technical paper, NOT a product manual. Its purpose is to assist the reader in understanding the wider issues relating to the interconnection of IBM products. It should be regarded by the reader in the same way as a paper published in a professional journal or read at a technical conference.

Detailed information about IBM products is given here incidental to objectives of the document, and while every effort has been made to ensure accuracy, such information should not be considered authoritative. Authoritative information about IBM products is contained in the official manuals for the product concerned.

Audience

This publication is primarily intended for people who have an interest in the fields of data communications or voice networking. The information is presented at a "technical conceptual" level and technical detail is only introduced when essential to communicate a particular concept.

Technical planners in user organizations who wish to broaden their understanding of high speed communications and the direction product development in the industry is taking.

IBM systems engineers evaluating the potential of different systems approaches may find the information helpful in understanding the emerging high speed technologies.

Structure

The document is organized as follows:

- Chapter 1, "Introduction"

  This chapter presents a broad outline of the topics dealt with in detail later in the document.

- Chapter 2, "A Review of Digital Transmission Technology"

  This chapter discusses digital transmission over copper wire. Starting with the concept of sending a digital signal along a wire it describes:

  - The problems of transmission over electrical wires.
  - The methods that have been developed to cope with these problems.
  - The line code structures that are used in the more significant modern digital systems.
  - How the problems associated with data transmission in the public network (subscriber loop) and LAN environments are addressed and solved.

- Chapter 3, "An Introduction to Fibre Optical Technology"

  This chapter presents a general introduction to fibre optic technology.

- Chapter 4, "Traffic Characteristics"

  High speed networks will be used for many purposes. The traditional interactive data traffic will of course be important but voice, image and video will be carried in the new networks.

  But these new types of traffic (new to packet networks) have completely different characteristics and requirements from traditional data.

  The characteristics of voice, image and video traffic are examined and compared to those of traditional data traffic.

- Chapter 5, "Principles of High Speed Networks"

  This chapter outlines the broad principles that must be followed in building a high speed network (one that is able to take full advantage of high speed). The principles are discussed in relation to those used by traditional packet networks.

- Chapter 6, "High Speed Time Division Multiplexing Systems"

  Time Division Multiplexing (TDM) is an important way of dividing a fast communications channel into many slower ones. It can be "inefficient" in the sense that it wastes capacity but it is simple and cost effective. If high speed data transmission is low in cost then perhaps TDM systems are the way to share the capacity.

  Integrated Services Digital Network (ISDN) is a TDM method of access to a public network. The basics of ISDN are described.

  Sonet and Synchronous Digital Hierarchy are very important protocols for sharing very high speed optical trunks. While currently these interfaces are not available to the end user (they are internal to the PTT) it is likely that they will be made available in the near future. These protocols are described in concept.

- Chapter 7, "Cell-Based Networking Systems"

Cell Relay systems are almost universally regarded as the long term future of data (and voice communications). Broadband ISDN is being developed as a cell relay system.

The concepts of cell relay are described and an overview of Broadband ISDN presented.

- Chapter 8, "High Speed Packet Networking"

  The technique of packet networking, familiar to all because of its use in SNA and in X.25 needs to be updated to operate at the very high speeds becoming available.

  The concept of Frame Switching is introduced since although it is not a high speed technology, it illustrates a packet network approach to the transport of link level frames.

  Frame Relay is then described and positioned as an important technology for the here and now.

  Paris is an IBM experimental networking technology designed for operation at very high speeds. Paris is described as an example of the direction research is taking.

- Chapter 9, "Shared Media Systems (LANs and MANs)"

  This chapter deals with Local Area Networks and Metropolitan Area Networks (LANs and MANs). After an overview of available LAN technologies the chapter details what happens to various LAN systems as speed is increased. Token-Ring continues to operate but becomes less and less efficient. FDDI is better (and designed to run on optical media) but it too starts to degrade at speeds in the gigabit region.

  DQDB is the telephone industry's first LAN protocol. It is the basis for several operational MAN systems and for the user to public network interface called "Switched Multi-Megabit Data Service" (SMDS). DQDB also has problems with efficiency and fairness at high speeds.

- Chapter 10, "The Frontiers of LAN Research"

  There is an enormous amount of research going on around the world aimed at inventing a LAN protocol that will be optimal in the speed range above about two gigabits per second. Two IBM Research project LAN prototypes are discussed (Metaring and CRAM) and a further proposal, which integrates the desirable features of both (CRMA-II).

- Chapter 11, "Networks of LANs"

  This chapter discusses LAN interconnection via bridges, routers and packet switches. The discussion leads to the conclusion that the functions of all three types of interconnection are converging and that in the future a single device (perhaps based on an architecture like Paris) will perform all these functions.

- Appendix A, "Review of Basic Principles"

  This appendix is a review of the basic principles involved in multiplexing (or sharing) a link or switching device. It is included as background for readers who may be unfamiliar with this area.

- Appendix B, "Queueing Theory"

  Queueing theory is the basis of network design and also the basis for understanding much of the discussion in this document. This appendix

presents the important results of queueing theory and discusses its impact on high speed protocols.

- Appendix C, "Getting the Language into Synch"

  One problem in developing a document such as this is the inconsistency in language between different groups in the EDP and Communications industries. This chapter deals with words related to the word "synchronous" in order that the reader may understand the usage in the body of the text.

- Appendix D, "An Introduction to X.25 Concepts"

  This chapter presents an introductory overview of X.25. Throughout this document reference is made to X.25 concepts and this appendix is included to assist people who are not immediately familiar with the jargon.

Acknowledgments

This publication is the result of a residency conducted at the the International Technical Support Center, Raleigh.

The author of this document is:

Harry J.R. Dutton
IBM Australia Limited.

The project leader was:

Peter Lenhard
IBM International Technical Support Center,
Raleigh.

Special thanks go to the following people for assistance in obtaining information, comments and review:

Bob Thoday	IBM Australia Limited
David R. Irvin	IBM US Telecommunications Center, Research Triangle Park, North Carolina.
Roy Evans	IBM United Kingdom Limited
Peter Russell	IBM United Kingdom Limited
Michel Demange	IBM Advanced Telecommunications Systems, La Gaude, France.
Pitro Zafiropulo	IBM Research Division, IBM Zurich Research Laboratory, Ruschlikon, Switzerland.
Harman R. van As	IBM Research Division, IBM Zurich Research Laboratory, Ruschlikon, Switzerland.
Neville Golding	IBM Network Systems LOB, Research Triangle Park, North Carolina U.S.A.
Ian Shields	IBM US, Raleigh, North Carolina U.S.A.
Raif O Onvural	IBM Advanced Telecommunications Systems, Raleigh, North Carolina U.S.A.

Chapter 1. Introduction

The phrase "high speed communication" is a relative term. It seems not long ago (1970) that a 4,800 bits per second leased line was considered very high in speed. In 1991, two megabit wide area links and LANs using speeds of 10 and 16 megabits per second have become universal. In the 1990's very much higher speeds (hundreds of megabits per second) will be common.

The networking techniques and technologies currently in use are unable to operate efficiently at the newly available higher speeds. This publication describes the new approaches that are required to enable efficient use of the new high speeds. So rather than defining "high speed" to mean any particular speed, for the purposes of this document "high speed" is held to mean "any speed that requires the use of new networking techniques for efficient operation". In practice this dividing line is somewhere around 100 megabits per second for LANs and about 35 megabits per second for wide area communications.

1.1.1 Yet Another Revolution?

The 1990's promises to be a period of very rapid change in the field of data communications. Over the years the word "revolution" has been used so frequently in the EDP industry that when a genuine revolution comes along we are robbed of the right word to describe it. But "revolution" is the right word this time.

Since the inception of data communications in the late 1950s, techniques have steadily improved, link speeds have progressively got faster and equipment cost has gradually declined. In general terms, it seems that the industry has progressed in price/performance measures by about 20% per annum compound. But now the cost of long communications lines is scheduled to reduce not by a few percent but by perhaps 100 or even 1000 times over a very few years!

The main causes of this are:

- The universal use of digital technologies within public telecommunications networks.
- The maturing of fibre optical transmission within public network facilities.

Both digital and fibre technologies have been in use for some years but they have formed only a small proportion of public networks. Digital "islands" in an analogue "sea." But the real benefits of digital transmission can only be realised when circuits are fully digital from end-to-end and when the digital network is available everywhere. In many western countries that time has arrived.

The consensus among economists in the industry seems to be that bandwidth cost (though not, yet, price) is reducing at a compound rate of 80% per annum. Most agree that these enormous cost savings will be passed on to end users (indeed this is already happening) but it will take some time for the full effect to be felt. This is because it is extremely difficult for any industry to manage change at anything like the rate now being experienced.

1.1.2 User Demand

Abundant, cheap communications means that many potential applications that were not possible before because of cost are now becoming very attractive. There are four generic requirements being expressed by users in relation to the new technologies;

1. To implement new data applications using graphics and image processing etc.
2. To do existing applications in better ways. For example instead of using coded commands and responses (such as are typical of banking and airline applications) users would like fast response full screen applications so that staff do not need the specialised skills that previous systems demanded.
3. To rationalise the many disparate networks that major users have. Many users have SNA*, DECnet**, TCP/IP and X.25 data networks as well as logical networks of FAX machines and a voice network. For management reasons (and for cost optimisation, though this is becoming less and less of a factor) users want to have only one network to handle all traffic.
4. Integration of voice, video, and data. Most large users have a voice network and a separate data network, and for cost and manageability reasons, would like to integrate these. In addition there is the future possibility of voice and data integrated applications.

   Many users see video as an important opportunity and would like to be able to handle this traffic.

1.1.3 The New Environment

Bandwidth Cost

The cost of transmitting "x" bits per second continues to reduce exponentially.

Attachment Cost

The cost of connecting a link (be it copper wire or optical fibre) from a user's location to the public network continues to *increase* with inflation. Having any connection at all involves "digging up the road" and this cost continues to rise. Today's technology enables us to use a single connection for many different things simultaneously and to use significantly higher transmission rates, but the cost of having a single connection is increasing.

Error Rates

Error rates on communication links have improved dramatically. (On an analogue telephone line with a modem a typical rate might have been 10^{-5} - or one error in every 100,000 bits transmitted. On a fibre connection this rate may be as low as 10^{-11} - a million times better!

Error Characteristics

The kinds of errors have changed also. On a digital connection errors tend to occur in bursts; whereas, on an analogue line they usually occur as single or double bit errors.

Propagation Delay

At 80% of the speed of light, the speed of message propagation on a communication line hasn't changed.

Storage Effect

At very high speeds long communication lines store a large amount of data. For example, a link from New York to Los Angeles involves a delay of roughly 20 milliseconds. At the 100 megabits per second speed of FDDI this means that two million bits are in transit at any time *in each direction*. So the link has approximately half a million bytes stored in transit.

This has a critical effect on the efficiency of most current link protocols.

Computer Technology

Computers continue to become faster and lower in cost at an approximate rate of 30% per annum compounded. This is not as simple as it sounds however - some things (for example the cost of main storage), have reduced faster than others (for example the costs of power supplies, screens and keyboards).

Data links are now considerably faster than most computers that attach to them. In the past, the communications line was the limiting factor and most computer devices were easily able to load the link at perhaps 95% of its capacity. Today, very few computer devices are capable of sustaining a continuous transfer rate of anything like the speed of a fast fibre link.

The "state of the art" in public network fibre facilities today is a transmission rate of 550 megabits per second, but rates of many times this are functioning in laboratories.

1.1.4 The Traditional Packet Data Network

It is a generally agreed consensus in the industry that the structures and protocols associated with the traditional packet network are obsolete in the new technological (and economic) environment. For the purpose of this discussion the Wide Area Networking parts of "Subarea" SNA and of APPN can be included among the plethora of "X.25" based packet switching networks.

1.1.4.1 Objectives

It is important to consider first the aims underlying the architecture of the traditional packet network.

1. The most obvious objective is to save cost on expensive, low speed communication lines by statistically multiplexing many connections onto the same line. This is really to say that money is saved on the lines by spending money on networking equipment (nodes).

   For example, in SNA there are extensive flow and congestion controls which when combined with the use of priority mechanisms enable the operation of links at utilisations above 90%. These controls have a significant cost in hardware and software which is incurred in order to save the very high cost of links.

   As cost of long line bandwidth decreases there is less and less to be gained from optimisation of this resource.

2. Provide a multiplexed interface to end user equipment so that an end user can have simultaneous connections with many different destinations over a single physical interface to the network.

3. Provide multiple paths through the network to enable recovery should a single link or node become unavailable.

1.1.4.2 Internal Network Operation

There seems to be as many different ways of constructing a packet network as there are suppliers of such networks. The only feature that the commodity "X.25 Networks" have in common is their interface to the end user - internally they differ radically from one another. Even in SNA, the internal operation of subarea networks and of APPN are very different from one another. That said, there are many common features;

Hop-by-Hop Error Recovery

Links between network nodes use protocols such as SDLC or LAPB which detect errors and cause retransmission of error frames.

Implicit Rate Control

Because the data link is far slower than any computer device (the end user devices and the network nodes could handle all of the data that any link was capable of transmitting) the link provides implicit control over the rate at which data can be delivered to the network. (This is separate to the explicit controls contained in the link control.)

Software Routing

Software is used for handling the logic of link control, for making routing decisions on arriving data packets and for manipulating queues.

Connection Orientation

Most (but not all) networks are based on the concept of an end-to-end connection passing through a set of network nodes. In X.25 these are called virtual circuits, in SNA there are routes and sessions. Within "intermediate nodes" a record is typically kept of each connection and this record is used by the software to determine the destination to which each individual packet must be directed.

Throughput

Based on the available link speeds in the 1970s and 1980s the fastest available packet switching nodes have maximum throughput rates of a few thousand packets per second.

Network Stability

In SNA and APPN (though not by any means in all packet networks) there is an objective to make the network as stable internally as possible thereby removing the need for attaching devices to operate stabilising protocols.

In SNA there is no end-to-end error recovery protocol across the network.[1] In the environment of the 1970s and early 1980s this would have involved crippling extra cost (storage, instruction cycles and messages) in every attaching device. Instead, extra cost was incurred within the network nodes because there are very few network nodes compared to the number of attaching devices and the total system cost to the end user was minimised.

---

[1] That is, there is no ISO "layer 4 class 4" protocol.

With recent advances in microprocessors and reductions in the cost of slow speed memories (DRAMs), the cost of operating a stabilising protocol within attaching equipment (or at the endpoints of the network) has reduced considerably.

Packet Size

Blocks of user data offered for transmission vary widely in size. If these blocks are broken up into many short "packets" then the transit delay for the whole block across the network will be considerably shorter. This is because when a block is broken up into many short packets, each packet can be processed by the network separately and the first few packets of a block may be received at the destination before the last packet is transmitted by the source.

Limiting all data traffic to a small maximum length also has the effect of smoothing out queueing delays in intermediate nodes and thus providing a much more even transit delay characteristic than is possible if blocks are allowed to be any length. There are other benefits to short packets; for example, it is easier to manage a pool of fixed length buffers in an intermediate node if it is known that each packet will fit in just one buffer and if packets are short and delivered at a constant, relatively even rate then the amount of storage needed in the node buffer pool is minimised.

Also, on the relatively slow, high error rate analogue links of the past, a short packet size often resulted in the best data throughput because when a block was found to be in error then there was less data that had to be retransmitted.

However, there is a big problem with short packet sizes. It is a characteristic of the architecture of traditional packet switching nodes that switching a packet takes a certain amount of time (or number of instructions) *regardless* of the length of the packet! That is, a 1000-byte block requires (almost) the same node resource to switch as does a 100-byte block. So if you break a 1000-byte packet up into 10, 100-byte packets then you multiply the load on an intermediate switching node by 10! This effect wasn't too critical when nodes were very fast and links were very slow. Today, when links are very fast and nodes are (relatively) slow this characteristic is the most significant limitation on network throughput.

It should be pointed out that SNA networks do not break data up into short blocks for internal transport. At the boundary of the network there is a function (segmentation) which breaks long data blocks up into "segments" suitable to fit in the short buffers available in some early devices. In addition there is a function (chaining) which enables the user to break up very large blocks (say above 4000 bytes) into shorter ones if needed but in general, data blocks are sent within an SNA network as single, whole, blocks.

Congestion Control

Every professional involved in data communication knows (or should know) the mathematics of the single server queue. Whenever you have a resource which is used for a variable length of time by many requesters (more or less) at random then the service any particular requester will get is determined by a highly predictable but very unexpected result. (This applies to people queueing for a supermarket check out just as much as messages queuing for transmission on a link.)

As the utilisation of the server gets higher the length of the queue increases. If requests arrive at random, then at 70% utilisation the average queue length will be about 3. As utilisation of the resource approaches 100% then the length of the queue tends towards infinity!

Nodes, links and buffer pools within communication networks are servers and messages within the network are requesters.

The short result of the above is that unless there is strict control of data flow and congestion within the network the network will not operate reliably. (Since a network with an average utilisation of 10% may still have peaks where utilisation of some resources exceeds 90% this applies to all traditional networks.) In SNA there are extensive flow and congestion control mechanisms. In an environment of very high bandwidth cost this is justified because these control mechanisms enable the use of much higher resource utilisations.

When bandwidth cost becomes very low then some people argue that there will be no need for congestion control at all (for example if no resource is ever utilised at above 30% then it is hard to see the need for expensive control mechanisms). It is the view of this author that congestion and flow control mechanisms will still be needed in the very fast network environment but that these protocols will be very different from those in operation in today' networks.

1.1.5 Is a Network Still Needed?

If the purpose of a network is to save money by optimising the use of high cost, low speed transmission links then if the transmission links are low cost, high capacity why have a network at all.

In practice, there are many other reasons to build a network.

- The reason that bandwidth costs so little is that individual links can be very fast. In a practical situation this means that single links need to be shared among many users.
- Users with many locations and devices typically need a single device to communicate with multiple other devices simultaneously. Without a network there would have to be a point-to-point link from each device to each other device that it needed to communicate with. This means that each device would need a large number of connections - all of which add to the cost. With a network each device needs only one connection.
- For reliability reasons we often need multiple routes to be available between end using devices (in case one "goes down" etc.).

The network is still very necessary. The question is really whether "packet" networks (or cell based networks) are justified when "Time Division Multiplexed (TDM) " networks may end up as more cost effective.

There is a question about who should own the network. A typical user may well buy a virtual network of point-to-point links from the PTT (telephone company) but the PTT will provide these links by deriving TDM channels from its own wideband trunks. From the user point of view they don't really have a network but from the PTT viewpoint this is a very important case of networking.

1.1.6 Alternative Networking Technologies

Within the industry there is considerable debate over the best way to satisfy the user requirements outlined above (integration of data, voice, video, image...). The only point of general agreement is that conventional data networking techniques are inadequate. Digital high speed networking techniques may be summarised as follows:

Frequency Division Multiplexing (FDM)

FDM is an analogue technique and is obsolete in the context of modern high speed communications. But techniques very similar to FDM are being researched for sharing of fibre optical links. In the context of fibre optics this is called Wavelength Division Multiplexing (WDM). Since wavelength is just the inverse of frequency (times some constant) the principles are very much the same.

Time Division Multiplexing (TDM)

If transmission bandwidth is to be very low in cost, then why spend money on expensive packet switching nodes. Why not use a simple time division multiplexing scheme for sharing the physical links and tolerate the "inefficiency" in link utilisation? Intelligent TDMs will be needed to set up and clear down connections and to provide a multiplexed connection to the end user, but the cost of these may be considerably lower than the packet node alternative.

Within public communications networks there are new multiplexing standards called SDH (Synchronous Digital Hierarchy) and SONET (Synchronous Optical Network). These provide standards for a significant simplification of the multiplexing techniques of the past (the "multiplex mountain"). See section 6.2, "SDH and Sonet" on page 117.

Fast Packet Switching, Frame Switching, Frame Relay

As explained above under "packet switching", every time a block of data is broken up into smaller packets or cells the overhead incurred to process it within the network is increased.

One of the aims of the new high speed network architectures is to mitigate this characteristic such that throughput in bytes per second is constant regardless of the frame size used. However, disassembling a logical record into a stream of cells and reassembling at the other end incurs overhead in the end nodes and there is additional overhead in end-to-end protocols for sending many blocks when one would do.

The conclusion to be reached from this is that data should be sent through the network as variable length frames. The principles of fast packet switching are:

1. Simplify the link protocols by removing the error recovery mechanism. That is check for errors and throw away any error data but do nothing else (rely on network end-to-end protocols for data integrity). This is acceptable since the new digital links have far fewer errors than previous analogue links.
2. Design the network such that all of the link control and switching functions can be performed in hardware logic. Software based switching systems cannot match the speed of the new links.

There is an international standard for "frame relay" called CCITT I.122 (see section 8.2, "Frame Relay" on page 146). A rather different system being prototyped by IBM research is called "Paris" (see section 8.3, "Packetised Automatic Routing Integrated System (PARIS)" on page 158).

Cell Relay

The big problem with supporting voice and video traffic within a data network is that of providing a constant, regular, delivery rate. Packet networks tend to take data delivered to them at an even rate and deliver it to the other end of the network in bursts.[2] It helps a lot in the network if all data is sent in very short packets or "cells". In addition the transit delay for a block of data through a wide area network is significantly shorter if it is broken up into many smaller blocks.

There is a standardised system for cell switching called "ATM" (Asynchronous Transfer Mode) which provides for the transport of very short (53 byte) cells through a SONET (SDH) based network.

The IEEE 802.6 standard for Metropolitan Area Subnetworks uses a cell based transfer mechanism.

LAN Bridges and Routers

Local Area Networks (LANs) are the most popular way of interconnecting devices within an office or work group. Many organisations have large numbers of LANs in geographically dispersed locations. The challenge is to interconnect these LANs with each other and with large corporate database servers.

LAN bridges and routers are a very popular technology for achieving this interconnection. The problem is that it is very difficult to control congestion in this environment. Also, neither of the popular LAN architectures (Ethernet** and token-ring) can provide sufficient regularity in packet delivery to make them usable for voice or video traffic. FDDI-II (Fibre Distributed Data Interface - 2) is a LAN architecture designed to integrate voice with data traffic. See section 9.3, "Fibre Distributed Data Interface (FDDI)" on page 183.

A remote LAN bridge that has several links to other bridges (the multilink bridge) is logically the same thing as a frame switch with a LAN gateway attached. Most proposed frame switch architectures handle LAN bridging as a part of the switch.

Metropolitan Area Networks

A MAN is just like a very large LAN (really a set of linked LANs) that covers a city or a whole country. (There are many MAN trials going on in various parts of the world but the first fully commercial service has been announced in Australia where a country-wide MAN called "Fastpac" will have "universal availability" over an area of three million square miles!)

MANs are different from LANs. In a LAN network, data from one user passes along the cable and is accessible by other users on the LAN. In a MAN environment this is unacceptable. The MAN bus or ring must pass through PTT premises only. End users are attached on

---

[2] There are techniques to help overcome this problem, however.

point-to-point links to nodes on the MAN bus. Data not belonging to a particular user is not permitted to pass through that user's premises.

MAN networks as seen by the end user are thus not very different from high speed packet networks. However, LANs and MANs are "connectionless" - the network as such does not know about the existence of logical connections between end users and does not administer them. Thus each packet or frame sent to the network carries the full network address of its destination. Most packet networks (fast or slow) recognise and use connections such as sessions or virtual circuits.

It should be noted that there is no necessary relationship (except convenience) between the internal protocol of the MAN and the protocol used on the access link to the end user. These will often be quite different.

1.1.7 Traditional End User Devices

The common data processing terminal devices used in the 1970s and 1980s were all designed to be optimal in the cost environment of those times. The IBM 3270 display subsystem is an excellent example. One of the key factors in the 3270 design was the assumption that "high speed" communication meant 4,800 bits per second! In addition, multidrop connection was considered vital because of the high cost of long lines. This determined the design of the formatted data stream, the method of interaction with a host user program and the assumed characteristics of user transactions.

What really happened was we decided to spend money on the device to save money on communications. Today an optimal solution may well be very different to traditional devices.

1.1.8 Traditional End User Systems

In the end analysis, the application being performed is the reason for existance of the whole system. There are many ways of performing the same application on a computer system but when most applications were designed this design and conception was done in a technological environment very different from that of today. The cost of communication is only one factor here, there is the cost of disk storage, the cost of executing instructions etc.

One very broad example would be the question of whether "distributed processing" is justified. Of course, there are many reasons for distributed processing but one key reason quoted in the past was to minimise communication cost (which was very high). With communication cost declining, it is no longer the driving force for distributed processing that it was in the past. Other things such as ease of system operation and management become very much more significant.

Another question entirely in application design is the amount of interaction between the end user and the processor. In the past we tried to minimise the amount of data sent - today that is perhaps not the most sensible thing to do. There is no reason today that a host processor should not operate with a remote screen keyboard device in the same way that a personal computer operates with its screen keyboard - and with the same (effectively instantaneous) response characteristics.

Of course the possibilities in application design that are opened up by high speed communication are the real reason to be interested in such communication in the first place. There are so many new techniques available in image and in interactive video that the application possibilities are endless.

Chapter 2. A Review of Digital Transmission Technology

2.1 Introduction

Over the last twenty years, the continued development of two technologies, Pulse Code Modulation (PCM) and Fibre Optics[3] have together provided a significant increase in available bandwidth for communication. In the case of PCM transmission, existing wires can be used to carry vastly increased amounts of information for little increase (often a decrease) in cost.

Among all that has been written about these technologies, the important facts to be remembered are:

1. Any (standard) telephone twisted pair of copper wires can carry data at a speed of 2 million bits per second (Mbps) in one direction.[4]
2. A standard telephone channel is 64 Kbps (thousand bits per second).
3. Two pairs of copper wires that currently carry only one call each now have the ability to carry 30 calls. (There are methods of voice compression that will double this at the least and potentially multiply it by 16. That is, 512 simultaneous voice calls on a single copper wire pair.)
4. A single optical fibre as currently being installed by the telephone companies is run at either 140 or 565 Mbps. At 140 Mbps, around 2,000 uncompressed telephone calls can be simultaneously handled by a single fibre WITHOUT the help of any of the available compression techniques. However, it is important to note that a single fibre can only be used in one direction at a time, so that two fibres are needed. The common optical cable being used between exchanges in the United States has 24 fibres in the cable.

Many, if not most, EDP communications specialists have been unprepared for these new technologies. This comes about because of the development of data communications using telephone channels. EDP specialists became used to the assumed characteristics of the telecommunications environment. It was thought by many that these characteristics were "laws of nature" and would remain true forever. It was quite a surprise to discover that far from being laws of nature, the characteristics of telecommunications channels could benefit from technological advance just as EDP systems could.

In the "early days" of data communications (the early 1960s), data signaling was done at what is now regarded as very low speed, 100 to 30 0 bits per second. In many countries when the PTT was asked for a leased line between two locations it was able to provide "copper wire all the way". Thus data was signaled in a very simple way. A one bit was a positive voltage (such as +50 volts) and a

---

3 The use of a light beam to transmit information was first demonstrated by Alexander Graham Bell in the year 1880. He demonstrated equipment to transmit speech. It took 100 years and the advent of glass fibre transmission for the idea to really become practical.

4 The actual speed that can be achieved here is variable depending on things like the length of wire and the environment in which it is installed. Over very short distances (up to 45 meters), TTP (Telephone Twisted Pair) can be used at 4 Mbps (it is used in this way by the IBM Local Area Network). **All numeric examples used in this paper are intended only to illustrate concepts and therefore must NOT be construed to be exact.**

zero bit was a negative voltage (such as -20 volts). This is much the same way as "telegraph" and "telex" equipment have operated since their inception.

Then two things happened. The first was that data communication requirements increased (users were finding applications for many more terminals) and terminal designs became more sophisticated. This resulted in a requirement for faster transmission. The second was that PTTs found themselves unable to provide end-to-end copper wire as they had in the past, and telephone channels had to be used. Most PTTs used interexchange "carrier systems" that used "frequency division multiplexing" between exchanges. Data users were given these channels and had to find ways of using them for data.

The important characteristic of these "multiplexed telephone channels" is that while they will pass alternating current between about 200 and 3,400 cycles, they will not pass direct current. So the old methods of direct current signaling would not work any more. (It is also worth noting that more modern telephone exchange switches were developed that used switching techniques that also limited bandwidth and would not pass direct current.) A device called a MODEM (MOdulator-DEModulator) was developed to send bits as a sequence of audio tones. In its simplest form a modem might signal a zero bit by sending a 1,200 cycle tone and a one bit by sending a 700 cycle tone.

Modems developed technologically very quickly indeed and became very sophisticated, complex and expensive devices. Sending 9600 bps through a telephone channel relies on many different techniques of "modulation" (variation of the signal to encode information) and the achievement borders on the theoretical limits of a telephone channel. In general, it is necessary to (very carefully) "balance" lines for this kind of modem and the line cannot actually be "switched" through an exchange. The line is simply a sequence of wires and frequency multiplexed channels from one end to the other. With a telephone channel 9,600 bps has been the limit. (Newer techniques can increase this to an amazing 16,800 bps.) Wider channels and higher speeds were obtained by combining several voice channels into one within the exchange such that the new channel has a "wider" carrying capacity. This is quite expensive.

Many people assumed that because of the need to limit the "bandwidth" within the telephone system, there was something that meant the ordinary copper wire twisted pairs played a part in this limitation. Expected problems included radio frequency emissions and "print through" of signals in one wire onto adjacent wires in the same cable by inductive coupling.

In fact, these, while always a consideration, were never the limiting factor. New technology has enabled the sending of data in just the way that it was in the early 1960s. A current in one direction for a one bit and in the opposite direction for a zero bit. (More accurately, complex changes of voltage are used.) The technique of Pulse Amplitude Modulation (PAM) is much more complex but the principle is the same and there is no need for the complex and expensive modem nor for "balancing" the wires in the circuit.

2.2 Electrical Transmission

As noted in the introduction, digital information can be transmitted over copper wires by encoding the information quite directly as changes of voltage on the wire. This is called "baseband" transmission and should be contrasted with "broadband" transmission where a carrier signal is "modulated" with the digital information (as for example in a modem).

The methods of encoding the information in the baseband technique are often grouped under the heading "Phase Amplitude Modulation" (PAM).[5] When analogue information (for example voice or video) is converted to digital form for transmission the most common technique is called "Pulse Code Modulation" (PCM). PCM coded voice is almost always transmitted on a wire (baseband) using PAM.

High speed digital bit streams can, of course, be sent by modulation of a high frequency (radio frequency) signal and this is sometimes done (such as in a broadband LAN) but this is relatively expensive compared to the simpler baseband method. In this chapter, only digital baseband PAM transmission is discussed.

The objective of data transmission is to transfer a block of data (or a continuous stream of bits) from one device to another over a single pair of wires. Because there are only two wires (and we are describing baseband transmission) transmission must take place one bit (or perhaps a small group of bits) at a time.

From the perspective of the transmission system we may insist that the device present a stream of bits for transmission. The problem then, is to transmit that bit stream unchanged from A to B.

2.2.1 Non-Return to Zero (NRZ) Coding

If the bit stream is to be sent as changes of voltage on a wire then the simplest coding possible is NRZ.

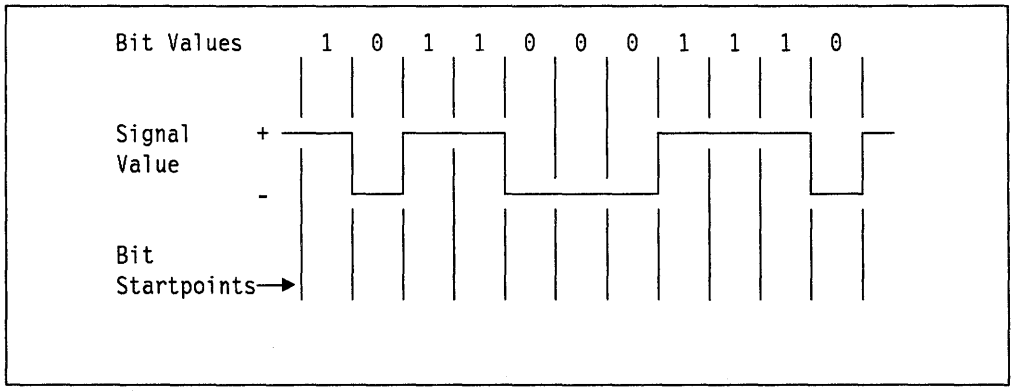


Figure 1. NRZ Coding

[5] Strictly, the word "modulation" means imposing changes onto a "carrier" signal in order to transmit information. In baseband transmission there is no carrier signal to be modulated, the digital bit stream is placed directly on the wire (or fibre). Nevertheless, most authors use the term modulation to describe the coding of a baseband digital signal.

Here a "1" bit is represented as a + voltage (or current in one direction) and a "0" bit is represented as a - voltage (or current in the opposite direction).[6]

This method of coding is used for short distance digital links such as between a data terminal device and a modem (the RS.232 or V.24 interface uses NRZ coding).

If a transmitter places a stream of bits on a wire using the NRZ coding technique, how can it be received? This is the core of the problem. Transmission is easy, recreating the original bit stream at a receiver can be quite difficult.

On the surface it looks simple. All the receiver has to do is look at its input stream at the middle of every bit time and the state of the voltage on the line will determine whether the bit is a zero or a one.

But there are two important problems:

1. There is no timing information presented to the receiver to say just where the middle of each bit is.

   The receiver must determine where a bit starts and then use its own oscillator (clock) to work out where to sample the input bit stream. But there is no economical way of ensuring that the receiver clock is running at exactly the same rate as the transmitter clock. That is, oscillators can be routinely manufactured to a tolerance of .005% but this is not close enough.

2. The signal will be changed (distorted) during its passage over the wire. See Figure 14 on page 33.

The signal will have started off as a sharp "square wave" at the transmitter and will be decidedly fuzzy when it gets to the receiver. The signal will no longer be just a positive or negative voltage. Instead, the voltage will change from one state to the other "slowly" passing through every intermediate voltage state on the way.

The receiver must now do two things:

1. Decide what line state is a zero and what state is a one.

   A simple receiver might just say "any positive voltage represents a one and any negative voltage a zero. This will be adequate in many situations, but this is by no means adequate in all situations as will be seen later.

2. Decide where bits begin and end.

   As a zero bit changes to a one bit the voltage will rise (perhaps quite slowly) from one state to the other. Where does one bit end and the next begin.

3. Decide where a new bit begins and an old one ends even if the line state does not change!

   When one bit is the same as the one before then the receiver must decide when one bit has finished and another begun. Of course, in data, it is very common for long strings of ones and zeros to appear, so the receiver must

---

[6] In fact, there is a simpler way of representing the information. A 1 bit might be the presence of a voltage and a zero bit the absence of a voltage. (Early morse code systems did use the absence of a voltage to delimit the start and end of each "dot" or "dash".) This technique is not used in modern digital signaling techniques because it is "unbalanced". The need for direct current (DC) balancing in digital codes is discussed later.

be able to distinguish between bits even when the line state hasn't changed for many bit times.

With simple NRZ coding this is impossible and something must be done to the bit string to ensure that long strings of zeros or ones can't occur.

A simple receiver might operate in the following way:

1. Sample the line at defined intervals faster than the known bit rate on the line (say seven times for every bit).

   When there is no data being sent, the line is usually kept in the one state.

2. When a state change is detected, this could be the start of a bit. Start a timer (usually a counter) to wait for half a bit time.
3. When the timer expires, look at the line. If it is the same as before then receive the bit. If not then the previous state change detected was noise - go back to 1 (looking for the start of a bit).
4. Set the timer for one full bit time.
5. Monitor the line for a change of state. If a change is detected before the timer expires then go back to step 2.
6. When the timer expires receive the bit.
7. Go to step 4

In the jargon the above algorithm is called a "Digital Phase Locked Loop" (DPLL). Consider what's happening here:

- The receiver is using the change of state from one bit to another to define the beginning of a bit (and the end of the last).
- When there are no changes, the receiver's clock is used to decide where the bits are.
- Whenever there is a state change, the receiver re-aligns its notion of where the bits are.

Successful operation is clearly dependent on:

- How good the receiver is at deciding when a state change on the line has occurred. (Since this is often gradual voltage change rather than an abrupt one, this is a judgement call on the part of the receiver.)
- How accurate the receiver's clock is in relation to the transmitters.
- How many times per bit the stream is sampled.

  Some practical systems in the past have used as few as five samples per bit time. The IBM 3705 communications controller (when receiving bits without an accompanying timing signal) sampled the stream 64 times per bit.

  Today's systems, using a dedicated chip for each line, often sample the line at the full clock frequency of the chip. The more frequent the sampling, the more accurate will be the result.

The above technique (the DPLL) is very simple and can be implemented very economically in hardware. But it is also very rough.

Notice here that the bit stream has been recovered successfully but the exact timing of the received bit stream has not. This doesn't matter to the example since the objective was to transfer a stream of bits, not synchronise timing. Later however, there will be situations where accurate recovered timing is critical to system operation.

Frequent state transitions are needed within the bit stream for the algorithm to operate successfully. The maximum number of bits without a transition is determined by the quality of the transmission line and the complexity of the receiver. Typical values for the maximum length of strings of ones or zeros in practical systems are between three and six bits.

2.2.2 Non-Return to Zero Inverted (NRZI) Coding

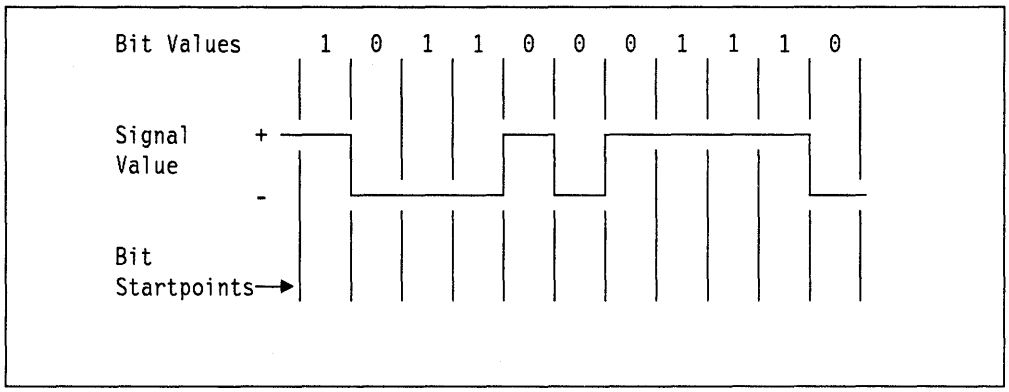


Figure 2. NRZI Coding

In order to ensure enough transitions in the data for the receiver to operate stably, a coding called Non-Return to Zero Inverted (NRZI) is often used. In NRZI coding, a zero bit is represented as a change of state on the line and a one bit as the absence of a change of state. This is illustrated in Figure 2.

This algorithm will obviously ensure that strings of zero bits do not cause a problem. But what of strings of one bits? Strings of one bits are normally prevented by insisting that the bit stream fed to the transmitter may not contain long strings of one bits. This can be achieved in many ways:

- By using a "higher layer" protocol (such as SDLC/HDLC) that breaks up strings of one bits for its own purposes. The HDLC family of protocols for example insert a zero bit unconditionally after every string of five consecutive ones (except for a delimiter or abort sequence).
- By using a code translation that represents (say) four data bits as five real bits. Code combinations which would result in insufficient numbers of transitions are not used. This is the system used in FDDI (section 9.3.5.2, "Data Encoding" on page 190) for example. Code conversion from 4-bit "nibbles" to 5-bit "symbols" is performed before NRZI conversion for fairly obvious reasons.

2.2.3 Coupling to a Line

While the above coding schemes are used in many practical situations (such as between a modem and a data processing device) they will not work properly over any significant distance using electrical baseband signaling. (They work well on optical links, however.)

It seems obvious but it must not be forgotten that when devices are connected together using electric wire there is an electrical connection between them. If that connection consists of direct connections of components to the wire there are several potential problems.

- Both the signaling wires and the grounds (earths) must be electrically connected to one another. If the devices are powered from different supplies (or the powering is through a different route),[7] then a "ground loop" can be set up.

  Most people have observed a ground loop. In home stereo equipment using a record turntable, if the turntable is earthed and if it is plugged into a different power point from the amplifier to which it is connected, then you often get a loud "hum" at the line frequency (50 or 60 cycles). The hum is caused by a power flow through the earth path and the power supplies.

  Ground loops can be a source of significant interference and can cause overload on attaching components.

- If the connection is over a distance of a few kilometers or more (such as in the "subscriber loop" connection of an end user to a telephone exchange) it is not uncommon for there to be a difference of up to 3 volts in the earth potential at the two locations. This can have all kinds of undesirable effects.

- It is not good safety practice to connect the internal circuitry of any device to a transmission line. In the event of a power supply malfunction it may be possible to get the main's voltage on the line.

  This does not help other equipment connected to the line and could leave a lasting impression on any technician who happened to be working on the line at the time.

So, it is normal practice to isolate the line from the equipment by using either a capacitor or a transformer. There are other reasons for using reactive coupling:

- A transformer coupling matches the impedance of the transmission line to the device and prevents reflection of echoes back down the line.
- The received pulses are re-shaped and smoothed in preparation for reception.
- The transformer filters out many forms of line noise.
- Neither a capacitor nor a transformer will allow direct current (DC) to pass.

  This means that the line is intentionally isolated from the user equipment. It becomes possible to put an intentional direct current onto the line. For example, in basic rate ISDN, a direct current is used on the same line as the signal to provide power for simple receivers. Also, in token-ring systems, a "phantom voltage" is generated in the attaching adapter and is used to signal the wiring concentrator that this device should be connected to the ring.

Transformer coupling is generally used in high speed digital baseband transmission systems. There are other advantages to transformer coupling:

- If the code is designed carefully, the transmission leads can be reversed (swapped) at the connection to the device without affecting its ability to correctly interpret the signal.
- Interference caused by "crosstalk" (the acquisition of a signal through inductive and capacitive coupling from other wires in the same cable) usually affects each signal wire equally. The fact that crosstalk signals tend to be

---

[7] More accurately, if there is any resistance in the connection between the grounds - there usually is.

equal on both wires means that when they are put through the transformer coupling they tend to cancel one another out.

The net is that crosstalk interference is greatly reduced.

2.2.4 DC Balancing

Any transmission code that can cause the line to spend more time in one state than the other has a direct current (DC) component and is said to be unbalanced. The presence of a DC component causes the transmission line and the coupling device to distort the signal. This phenomenon is technically called "baseline wander" and the effect is to increase the interference caused by one pulse with a subsequent pulse on the line. (See Intersymbol on page 34.)

In addition DC balancing simplifies transmit level control, receiver gain and receiver equalisation.

Both the NRZ and NRZI codes described above are unbalanced in this way and so are unsuitable for high speed digital transmission **on electrical media** (its fine on optical media).[8] So we need a different kind of code.

2.2.5 Pseudoternary Coding

Pseudoternary coding is the simplest DC balanced code and is used at the "S" and "T" interfaces in basic rate ISDN. This code uses three line states to represent two binary conditions (0 or 1) - hence the name.

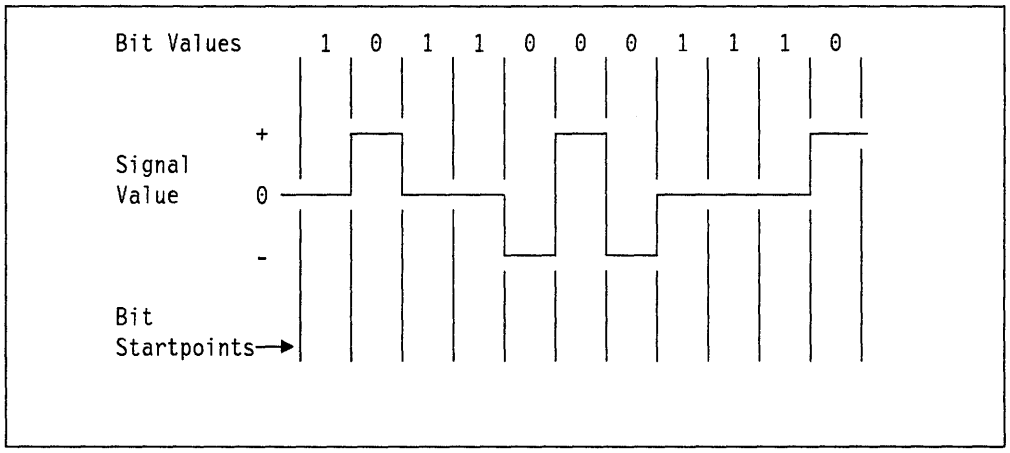


Figure 3. B_ISDN Pseudoternary Coding Example

- A one bit is represented by no voltage present on the line.
- A zero bit is represented by either a positive or a negative pulse.

Pulses representing zero bits must strictly alternate in state. Successive zero pulses (regardless of how many one bits are in between) must be of opposite states. Hence the coding is DC balanced. If they are the same state then this is regarded as a code violation.

[8] The suggestion of DC balancing on a fibre seems absurd. However, this is not always so. Optical transmissions are sent and received electronically. It is often advantageous to have an AC coupled frontend stage to a high gain receiver. So some systems do balance the number of ones and zeros sent to give a DC balance to the electrical coupling stage in the receiver.

The properties of this code are used in an elegant and innovative way by the Basic Rate ISDN "S" interface. See the discussion in section 6.1.3.3, "The ISDN Basic Rate Passive Bus ("S" Interface)" on page 108.

The drawback of using this code compared to NRZI is that (other things being equal) it requires 3dB more transmitter power than NRZI. It also requires the receiver to recognise three states rather than just a transition - this means a slightly more complex receiver is required. Notice also that it is the one bit that is represented as null voltage and the zero that is a voltage rather than the opposite.

Of course, the synchronisation problem discussed above is still present. A long string of one bits will cause the receiver to loose synchronisation with the transmitter. (This problem is overcome in B_ISDN by the regular occurrence of zero bits in the framing structure, the relatively slow speed of the interface (144 Kbps) and the use of a much better timing recovery system than the simplistic DPLL described earlier.)

2.2.5.1 Alternate Mark Inversion

Pseudoternary code is also called "AMI" for Alternate Mark Inversion or "Bipolar" code. The only difference between AMI and the form of pseudoternary used in B_ISDN is that the representation of the one and zero states are reversed. Thus in AMI, a no voltage state is interpreted as a zero bit and alternating voltage pulses are interpreted as ones. Of course, this is just a convention and makes no real difference at all. B_ISDN reverses the convention for an excellent reason related to the operation of the "passive bus". See section 6.1.3, "ISDN Basic Rate" on page 105.

2.2.6 Timing Recovery

There are many situations where a receiver needs to recover a very precise timing from the received bit stream *in addition* to just reconstructing the bit stream. This situation happens very frequently:

- In B_ISDN a frame generated by the DTE must be in very nearly exact synchronisation with the framing sent from the DCE to the DTE.
- In P_ISDN a similar but a little less critical requirement exists.
- In token-ring, the all important ring delay characteristic is minimised by maintaining only minimal (two bits)[9] buffering in each ring station. This requires that the outgoing bit stream from a station be precisely synchronised with the received bit stream (to avoid the need for elastic buffers in the ring station).

In order to recover precise timing not only must there be a good coding structure with lots of transitions but the receiver must use a much more sophisticated device than a DPLL to recover the timing. This device is an (analogue) phase locked loop.

---

[9] In the current IBM Token-Ring adapter this is actually two and a half bits or five bps. However, in principle it is possible to cut this down to one and a half bits.

2.2.6.1 Phase Locked Loops (PLLs)

Earlier in this section (page 15) the concept of a simple "digital phase locked loop" was introduced. While DPLLs have a great advantage in simplicity and cost they suffer from three major deficiencies:

- Even at quite slow speeds they cannot recover a good enough quality clocking signal for most applications where timing recovery is important.
- As link speed is increased, they become less and less effective. This is due to the fact alluded to earlier in this book, that circuit speeds have not increased in the same ratio as have communication speeds.

  A DPLL needs to sample the incoming bit stream many times per bit. With a link speed of 2,400 bits per second this isn't very difficult to do even by programming. But at multi-megabit speeds it becomes more and more costly and then (as speed becomes too great), impossible.
- As digital signals increase in speed (where speed begins to be limited by circuit characteristics), they start behaving more like waveforms and less like "square waves" and the simplistic DPLL technique becomes less appropriate.

What is needed is a continuous time, analogue, PLL.

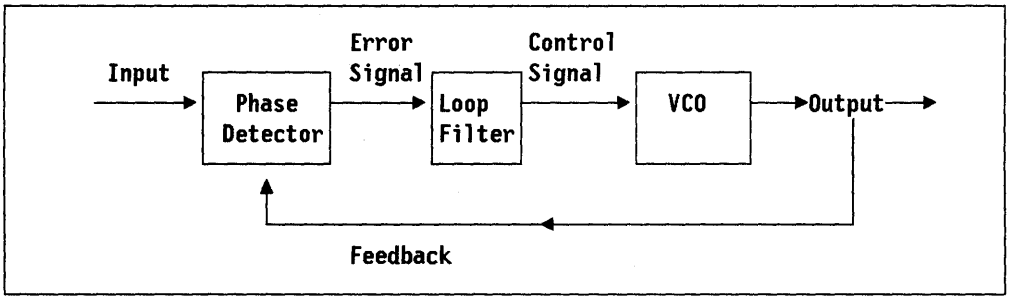


Figure 4. Operating Principle of a Continuous (Analogue) PLL

The concept is very simple. In Figure 4 the VCO is a Voltage Controlled Oscillator and is the key to the operation.

- The VCO is designed to produce a clock frequency close to the frequency being received.
- Output of the VCO is fed to a comparison device (here called a phase detector) which matches the input signal to the VCO output.
- The Phase Detector produces a voltage output which represents the difference between the input signal and the output signal.

  (In principle, this device is a lot like the tuner on an AM radio.)
- The voltage output is then used to control (change) the frequency of the VCO.

Properly designed, the output signal will be very close indeed to the timing and phase of the input signal. There are two (almost conflicting) uses for the PLL output.

1. Recovering the bit stream. That is, providing the necessary timing to determine where one bit starts and another one ends.
2. Recovering the (average) timing. That is, providing a stable timing source at exactly the same rate as the timing of the input bit stream.

Many bit streams have a nearly exact overall timing but have slight variations between the timings of individual bits.

The net of the above is that quite often we need two PLLs: one to recover the bit stream and the other to recover a precise clock. This is the case in most Primary Rate ISDN chip sets.

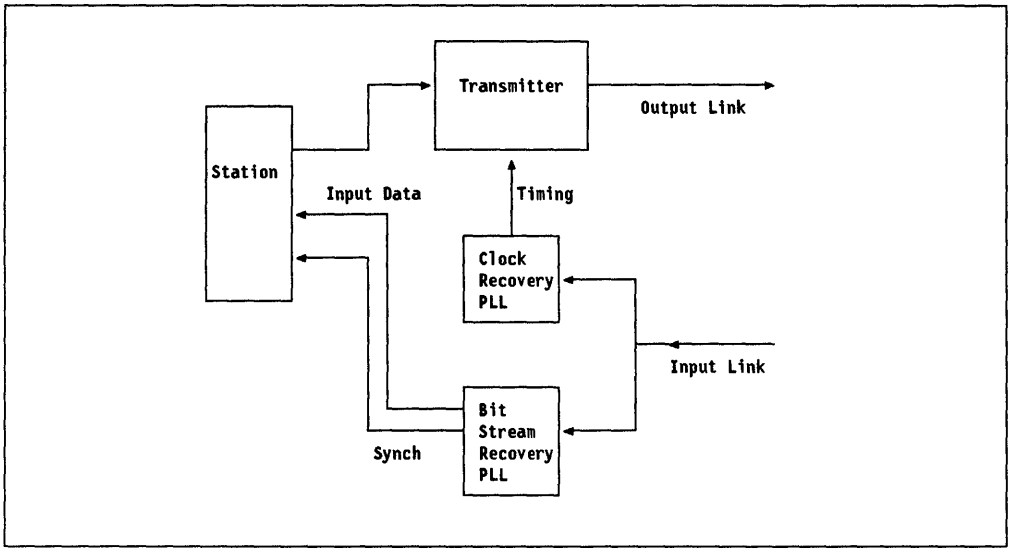


Figure 5. Clock Recovery in Primary Rate ISDN. Two PLLs are used because of the different requirements of bit stream recovery and clock recovery.

PLL design is extremely complex and regarded by many digital design engineers as something akin to black magic. It seems ironic that the heart of a modern digital communications system should be an analogue device.

PLL quality is extremely important to the correct operation of many (if not most) modern digital systems. The Basic Rate ISDN "passive bus" and the token-ring LAN are prime examples.

2.2.6.2 Jitter

Jitter is the generic term given to the difference between the (notional) "correct" timing of a received bit and the timing as detected by the PLL. It is impossible for this timing to be exact because of the nature of the operation being performed. Some bits will be detected slightly early and others slightly late. This means that the detected timing will vary more or less randomly by a small amount either side of the correct timing - hence the name "jitter". It doesn't matter if all bits are detected early (or late) provided it is by the same amount - delay is not jitter. Jitter is a random variation in the timing either side of what is correct.

Jitter is minimised if both the received signal and the PLL are of high quality. But although you can minimise jitter you can never quite get rid of it altogether.

Jitter can have many sources such as distortion in the transmission channel or just the method of operation of a digital PLL. Sometimes these small differences do not make any kind of difference. In other cases, such as in the IBM Token-Ring, jitter accumulates from one station to another and ultimately can result in the loss or corruption of data. It is jitter accumulation that restricts the maximum number of devices on a token-ring to 260.

2.2.6.3 Repeaters

The ability to use repeaters is one of the principal reasons that digital transmission is so effective.

As it travels along a wire, any electrical signal is changed (distorted) by the conditions it encounters along its path. It also becomes weaker over distance due to energy loss (from resistance and inductance) in the cable. After a certain distance it is necessary to boost the signal.

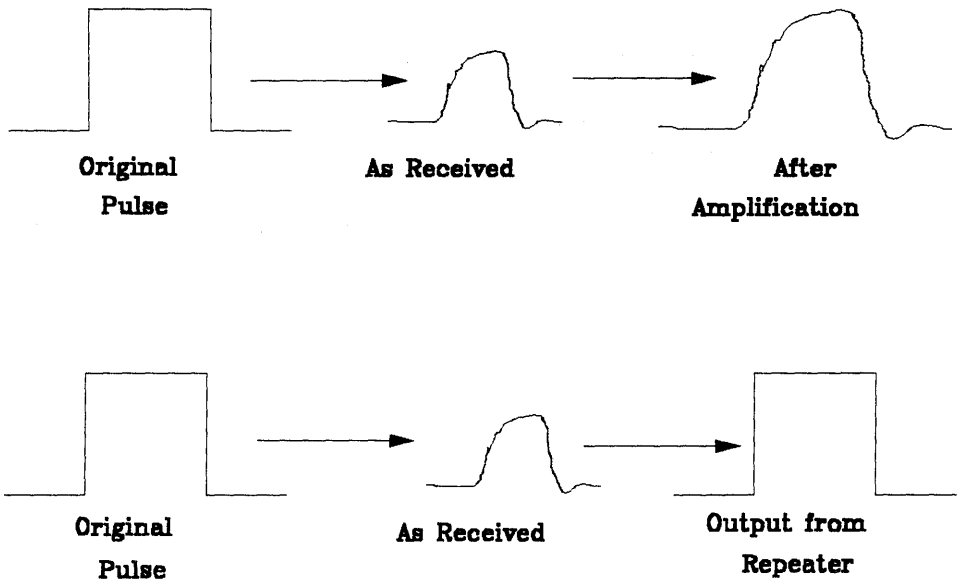


Figure 6. Function of a Repeater

The signal can be boosted by simply amplifying it. This makes the signal stronger, *but* it amplifies a distorted signal. As the signal progresses further down the cable it becomes more distorted and all the distortions add up and are included in the received signal at the other end of the cable. Analogue transmission systems must not allow the signal to become too weak anywhere along their path. This is because we need to keep a good ratio of the signal strength to noise on the circuit. Some noise sources such as crosstalk and impulse noise have the same real level regardless of the signal strength. If we let the signal strength drop too much the effect of noise increases.

In the digital world things are different. A signal is received and (provided it can be understood at all) it is re-constructed in the repeater. A new signal is passed on which is completely free from any distortion that was present when the signal was received at the repeater.

The result of this is that repeaters can be placed at intervals such that the signal is still understandable but can be considerably weaker than would be needed were the signal to be amplified. This means that repeaters can be spaced further apart than can amplifiers and also that the signal received at the far end of the cable is an exact copy of what was transmitted with no errors or distortion at all.

2.2.7 High Speed Digital Coding (Block Codes)

In the above discussion of pseudoternary (AMI) codes it was seen that AMI codes are DC balanced and are relatively simple to implement. However they suffer from the problem that long strings of zeros do not cause any transitions and so there is no way for a receiver to recover either the bits or the timing.

We need to do something to the bit stream to add transitions without destroying the advantages of the code. There are many things that could be done such as bit-stuffing (as in SDLC/HDLC) or 4B/5B code translation (as in FDDI) but these techniques add overhead in the form of extra bits to the stream.[10]

The general solution adopted here is to replace a fixed length string of zeros with a different string (one containing transitions). But of course, if that is done using valid bit combinations, when such a string is received, the receiver has no way of knowing whether the string is a substitution or not. The answer is to make use of the redundancy of the AMI code by introducing (hitherto illegal) signal combinations called code violations.

A code violation is where there are two successive voltage pulses of the same polarity.

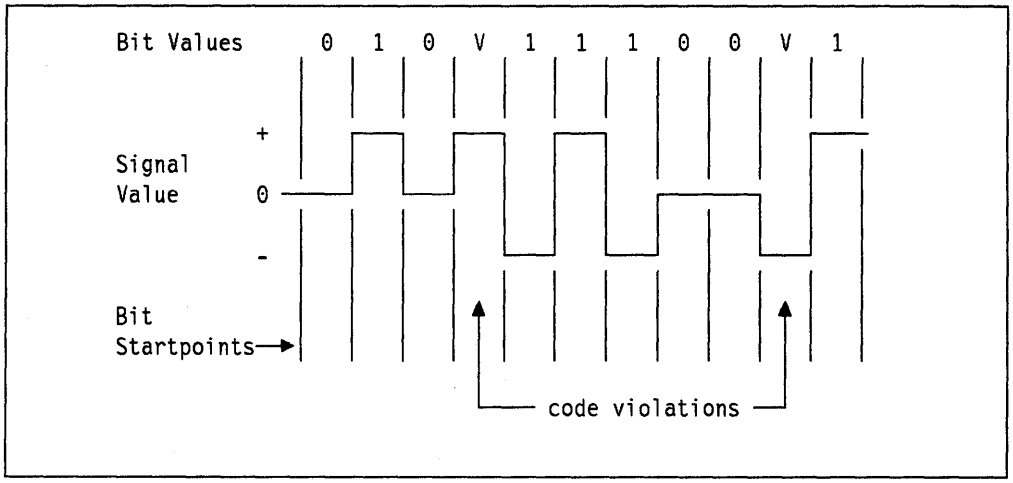


Figure 7. Code Violation in AMI Coding. A violation is a bipolar pulse of the same polarity as the preceding bipolar pulse.

The code violation breaks the rule that alternate voltage pulses (representing ones) must have opposite polarity.

In order to introduce the necessary transitions the general technique is to replace a string of k bits with a predetermined (unique) bit pattern, also k bits long, but including a code violation or two to ensure its uniqueness.

Problem solved - but what about DC balancing? Code violations by definition introduce an unbalanced DC component. Depending on the code used, the polarity of the code violations alternates in order to achieve an overall balance.

There are many suggested codes. BkZS, HDBk and kBnT are names of families of codes (k=3, 4, 5... n= 2, 3..) which have great academic interest but two

[10] Bit stuffing has another much worse effect: It adds a variable number of bits to the frame destroying the strict frame synchronisation required for TDM systems.

members of these families are very important commercially. These are HDB3 and B8ZS because they are used by Primary Rate ISDN.

2.2.8 High Density Bipolar Three Zeros (HDB3) Coding

This code is used for two megabit "E1" transmission outside the US and is defined by the CCITT recommendation G.703. It is the code used in Primary Rate ISDN at the two megabit access speed.

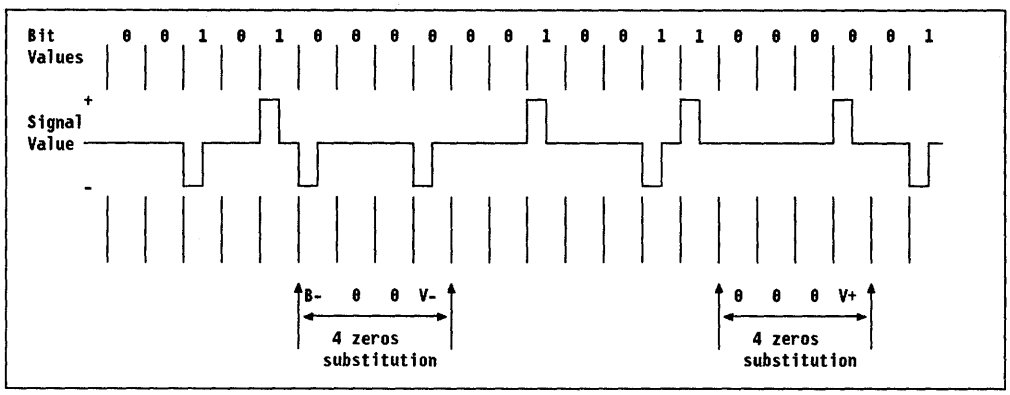


Figure 8. Zeros Substitution in HDB3

The basic concept is to replace strings of four zero bits[11] with a string of three zeros and a code violation.[12]

```
0000 is replaced by 000V  (v = a code violation)
```

But the code violation destroys the very thing that we are using the AMI code for - its DC balanced nature. Using the above rule a long string of zeros would be replaced with

```
000V000V000V...
```

Each code violation pulse would have to have the same polarity otherwise it would be a one bit - not a code violation. Even short strings of zeros within arbitrary data would cause an unbalance as the polarity of the preceding one bit would not be predictable or controllable.

What is needed is to organise things so that successive code violations are of alternating polarity. This can be done by using a different substitute string beginning with a phoney "one" bit. This phoney one bit is a valid bipolar pulse which could represent a one bit, but in this context (followed two bits later by a violation) is interpreted as a zero.

```
0000 can be replaced by B00V (B = a valid bipolar pulse)
```

The whole is organised so that the polarity of code violations alternates. The code word transmitted when a string of four zeros is detected is then chosen to make the number of "B"s between successive "V"s odd.

The rule is simple: if the polarity of the last violation is the same as that of the most recent one bit then send the string B00V (that is, invert the polarity of the violation), if the polarities are different send 000V. This rule is summarised below.

---

[11] The size of the bit string replaced, in this case 4 bits is the "k" in the name of the code plus one. Thus in HDB3, k=3 and the number of bits replaced is k+1=4.

[12] The + or - suffixes attached to the B and V notations in the figure denote the polarity of the pulse.

		Polarity of last Code Violation	
		+	-
Polarity of preceding one bit	+	B-00V-	000V+
	-	000V-	B+00V+

The + and - signs denote the polarity of the preceding pulse.

The result can be seen in Figure 8 on page 24. Reading the bit stream from left to right the first pattern is B-00V- (indicating that the previous violation must have been positive). The next violation is 000V+ and is of opposite polarity.

2.2.8.1 The Duty Cycle

In Figure 8 on page 24 the pulses were represented as only being in either + or - state for one half of one bit time. In this case the code is said to have a 50% duty cycle.[13] This is commonly done in digital codes for two reasons:

1. It reduces "intersymbol interference", that is, the distortion of one pulse by the preceding pulse. The line is allowed to settle to a zero state before the next bit time.
2. It means that only half the transmitter power is required compared to the power needed for transmitting for the full bit time.

The cost is that the nominal bandwidth occupied is increased (not a problem in most baseband environments) and that a slightly more precise receiver is needed.

[13] This is sometimes referred to as "half bauded coding".

2.2.9 Bipolar with Eight Zeros Substitution (B8ZS) Coding

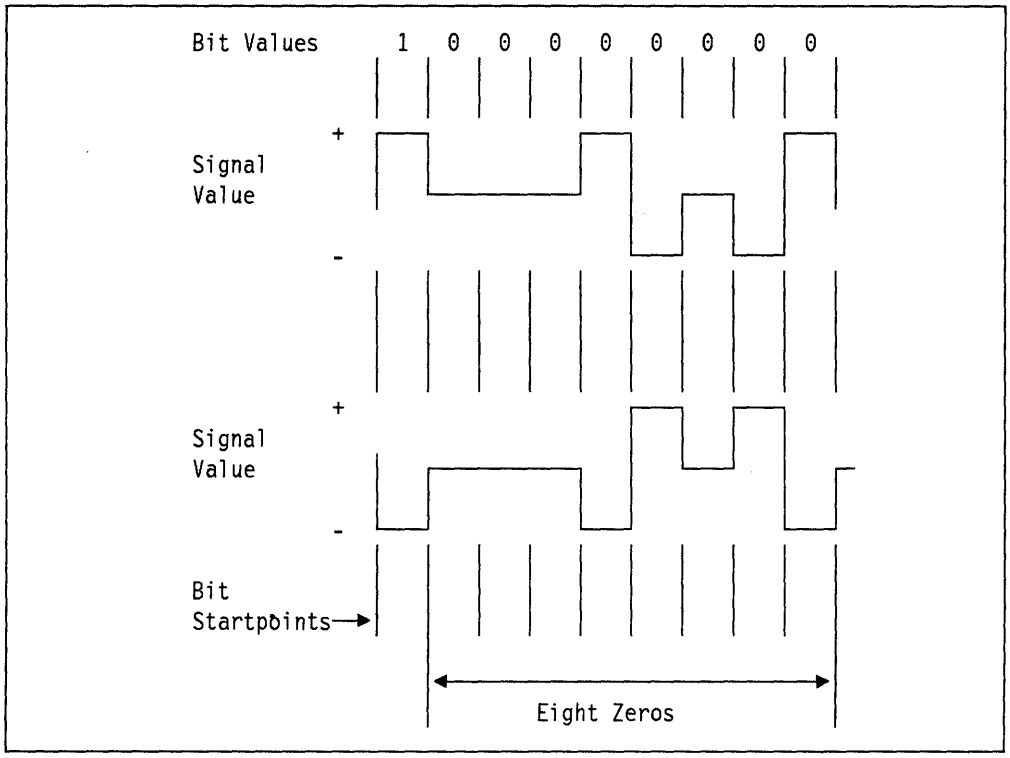


Figure 9. B8ZS Substitutions. Either of two alternative strings are used depending on the polarity of the preceding one bit.

This code is used for the 1.544 megabit US Primary Rate ISDN access link. In principle this is the same as HDB3 in the sense that a predetermined pulse string replaces a string of zeros. In this case, the string is 8-bits long and like HDB3, may be anywhere in the transmitted data block - it is not confined to byte boundaries. This means that in B8ZS the maximum length of a string of zeros (after substitution) is 7. In HDB3 the maximum number of consecutive zeros allowed in the substituted string is 3.

Because the substitution is of eight zero bits, there is room in the string for more than one code violation and a couple of valid bipolar pulses thrown in. The substituted string is <000VB0VB>. There are two strings. Selection of one to be used depends on the polarity of the preceding one bit (you have to create a violation).

Note that because both strings are themselves DC balanced, multiple repetitions of the same string are possible.

2.2.10 4-Binary 3-Ternary (4B3T) Code

This code is another ternary code designed to be used in an environment where bandwidth is somewhat limited. In the codes discussed above, a two-state binary bit was mapped into a three-state ternary code. This gave a number of advantages in error detection and simplicity in the receiver, but there are some occasions where bandwidth limitation, while not severe, still must be considered.

One such environment is the "U" interface of Basic Rate ISDN. As discussed in section 6.1.3, "ISDN Basic Rate" on page 105, this interface is *not* internationally

standardised. Different countries adopt different techniques. 4B3T code is used in Germany.

The problem here is full-duplex transmission over a two-wire line. (A very good trick by any test!) The challenge is to get high quality transmission between the PTT exchange and the end user *at very low cost*.

Four binary bits (as a group) are transmitted as three ternary states (also as a group), thus there is a 25% reduction in the baud rate of the line compared with AMI codes. This helps noise immunity but at the cost of complicating the receiver. In the B_ISDN environment this means that 160 Kbps is transmitted as 120 kbaud.

Four binary bits represent 16 different combinations. Three ternary digits represent 27 combinations. What happens is that we map groups of four binary bits to a group of three ternary bits (hence the name of the code).

It is not quite as simple as a one-to-one mapping. Some care is needed in choosing the mappings. There are still the twin problems of getting enough transitions and making sure the code is DC balanced. The trick is that some code blocks (strings of three ternary digits) have two different (logically opposite) representations. In Figure 10 ternary line states are represented by +, - or 0.

Binary String	Ternary String Mode 1	Mode 2	Digital Sum
0000	+0-	+0-	0
..	.	.	.
0011	+-0	+-0	0
0100	++0	--0	±2
..	.	.	.
0111	+++	---	±3
..	.	.	.
1100	00+	00-	±1
..	.	.	.
Unused	000	000	0

Figure 10. Principle of 4B3T Code

Notice the two modes. Ternary strings that are DC balanced in themselves (such as the +-0 string) represent the same binary block in either mode. Ternary blocks that are not DC balanced have two representations (one with positive DC bias and its inverse with negative bias). The amount of bias is noted in the table.

What happens is that the sender keeps track of the RDS (Running Digital Sum) of the block being transmitted. When the sum is positive, the next code word sent is from the Mode 2 column; if the sum is negative the next code word sent is from the Mode 1 column. The receiver doesn't care about digital sums; it just takes the next group of three states and translates it into the appropriate four bits. The combination of 000 is not used because a string of consecutive groups would not contain any transitions (the strings of --- and + + + are possible because they alternate with each other if the combination 01110111 occurs in the data).

The code achieves a statistical DC balance rather than a balance in each transmitted symbol. This has been found to help in self-cancellation of echoes and ISI but it is not as good as DC balancing within each symbol.

2.2.11 Differential Manchester Coding

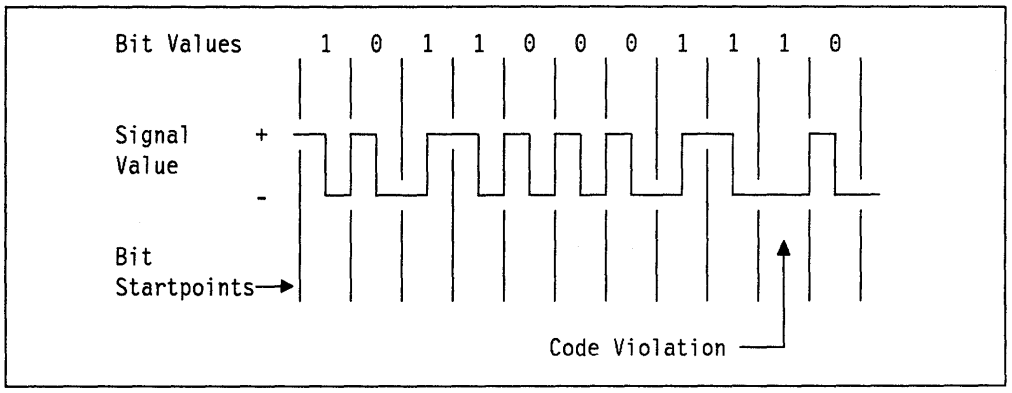


Figure 11. Differential Manchester Coding Example. The presence or absence of a transition **at the bit start point** *determines whether the bit is a one or a zero. A code violation occurs when the signal state does not change at the middle of a bit time.*

Differential Manchester[14] is the electrical level coding used in the IBM Token-Ring LAN.

The most significant factor in the choice of this code for the IBM Token-Ring LAN was the desire to minimise the "ring latency" (the time it takes for something to travel around the ring). The mechanism used to achieve this is to minimise necessary buffering (delay) in each attaching ring station. (IBM Token-Ring adapters contain 2½ bits of delay.) To do this you need to eliminate the "elastic buffer" which would be required if each station had an independent clock. To do this each station's transmitter must run at the same rate as the data it is receiving. This means that each station must be able to derive a highly accurate timing source from its received data stream. To do this you need a code with a lot of transitions in it. Hence the choice of Differential Manchester Code.

The next significant desire was to minimise the cost of each attaching adapter albeit that function should not be sacrificed purely for cost saving. (It was felt that as chip technology improved over the years the initial high cost of TRN chip sets would reduce significantly - and this has indeed happened.)

The desired speed range of around ten megabits per second (in 1981) dictated the use of shielded cable. The shielded twisted pair that was decided upon for transmission can easily handle signals of several hundred megabits per second, so there was no advantage to be gained from limiting the baud rate (signaling frequency).

Differential Manchester intentionally uses a high baud rate in order to minimise the cost of the adapter and the amount of buffering required in each ring station.

The baud rate is twice the bit rate. On a 4 Mbps token-ring the "baud rate" (that is the number of state changes on the line) is 8 megahertz. A 16 Mbps token-ring runs at 32 megabaud (32 megahertz). Some people consider that this is a waste of bandwidth since there are codes in use that allow up to 6 bits for

[14] Differential Manchester coding is a member of a class of codes called "biphase modulated codes"

each state change.[15] FDDI, for example, uses a 4B/5B code that allows 100 megabits to be sent on the line as 125 megabaud.

Differential Manchester coding provides an elegant way of satisfying the requirements.

1. Because it provides at least one state transition per bit, a receiver can derive a very stable timing source from the received bit stream with minimal circuitry. This means that the station can be a simple repeater with minimal buffering (one bit only) and hence minimal delay.

   Had it not been possible to derive such a stable timing source, the alternative was to use a structure like the one used in FDDI. This would mean a separate transmit oscillator with an elastic buffer in each station - a solution which adds to both the node delay and the cost.

2. The inbuilt DC balanced characteristic of the code meant that there is no data code translation required before transmission. In addition it means that additional DC balance bits are not required to be added to the data - as is required in some other codes.

3. The stability of the derived timing minimises "phase jitter" and thus allows more stations to be connected to a single ring segment.

4. A code violation is used as a frame delimiter and thus no special delimiting function is needed in the data.

5. Because the modulation technique has such a large amount of redundancy, if a link error occurs it is very likely that a code violation will result. This provides an elementary form of error detection.

2.2.12 Multi-Level Codes

Multi-level codes are not often used in digital baseband transmission because they complicate the design of the receiver (that is, add to cost). They do however, use less bandwidth (this just means that the pulses are longer) than binary or ternary codes and are attractive in a bandwidth limited environment. There are other advantages such as a reduced susceptibility to noise interference but this tends to be offset by the fact that multi-level codes have much less redundancy than other codes (AMI for example) and therefore are not as good at error detection.

2.2.12.1 2-Binary 1-Quaternary (2B1Q) Code

An important example of a multi-level code is the 2-binary 1-quaternary (2B1Q) code used in the US for the B_ISDN "U" interface. As mentioned above, this is a two-wire full-duplex environment. This code is important because (in the US) the U interface is available for the direct attachment of end user equipment. This means that many manufacturers will build equipment using this code directly.

2B1Q code uses 4 line states to represent two bits. In the B_ISDN environment this means that the 160 Kbps "U" interface signal is sent at 80 kbaud.

---

[15] In an analogue carrier system where available bandwidth is severely restricted then perhaps the word waste would be justified. In a baseband LAN environment where the signal is the only one on the wire this is difficult to call waste since were it not used for this purpose, the additional capability of the wire would be unused.

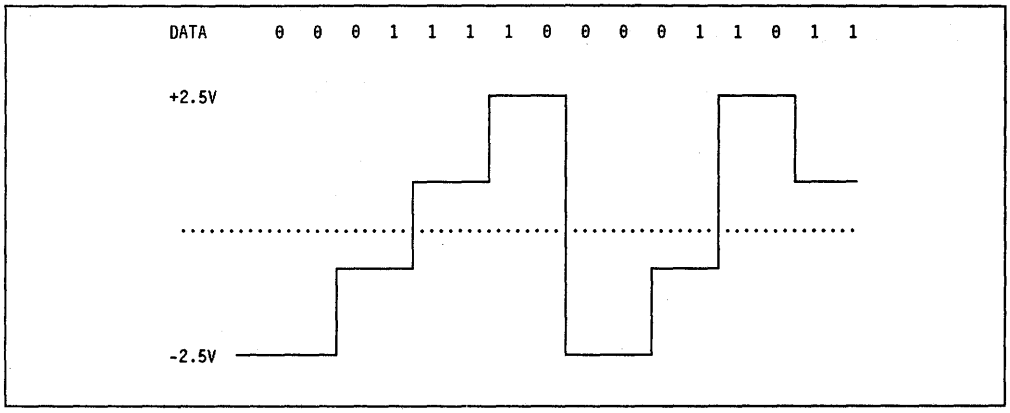


Figure 12. 2-Binary 1-Quaternary Code

In B-ISDN PAM (Pulse Amplitude Modulation) is used to carry the 2B1Q signal. This is just ordinary baseband digital signaling *but* the receiver must now be able to distinguish the amplitude of the pulses (not just the polarity). There are four voltage states with each state representing two bits.

Another point to note is that this code is *not* DC balanced! Further it does not inherently contain sufficient transitions to recover a good clock. For example the sequences:

```
00000000  or 0101010101  or 1010101010  or 11111111
```

all result in an unchanging line state leaving nothing for the receiver to synchronise on.

To overcome this difficulty data is "scrambled" (converted by a procedure that ensures pseudo randomness in the data) before transmission and "unscrambled" after it is received.

2.3 Practical Transmission

There are two important environments where baseband digital transmission is used. These are:

1. The connection between an end user (such as a private home) and the local telephone exchange. This is called the "subscriber loop".
2. Local area networks.

These environments are very different but nevertheless share many common characteristics.

2.3.1 Types of Transmission Media

In digital baseband communication the medium is always a pair of wires (or rather, two conductors). There are five configurations which will be discussed more fully later:

Open Wire

Open wire is the historic form of telephone transmission. Two wires are kept apart by some four feet and strung on "telephone poles".

In 1991 this transmission medium is all but extinct. *But it is still the best* from the perspective of sending analogue voice signals over a distance. It has much lower losses and distortion than other methods. For example, in the past it was common to have a telephone subscriber loop of up to 70 kilometers or so without amplification. Sadly, it is now impractical in most situations.

Twisted Pair

Two wires are held together by twisting them and are insulated with either paper or plastic. There are many types and grades of twisted pair cables. The most common type is used by telephone companies for the "subscriber loop" (see section 2.3.3, "The Subscriber Loop" on page 35) and is called "telephone twisted pair" (TTP).

Twisted pair is also very popular in some quarters for LAN cabling but (depending on the grade and quality of the particular cable) can have significant limitations in that environment. See section 2.3.6, "LAN Cabling with Unshielded Twisted Pair" on page 41.

Shielded Twisted Pair

This is the best medium for digital baseband communication and is used for many LAN installations. It consists of multiple twisted pairs of wires in a single cable within which each pair is surrounded by a conductive shield. The whole cable is surrounded by an earthed, metal mesh shield. This works well for digital baseband transmission at speeds up to 300 megabaud and beyond.

The IBM specification for shielded twisted pair is called "Type One Cable" and is shown in Figure 13 on page 32.

Screened Twisted Pair

Screened twisted pair is the case where multiple twisted pairs are bound together in the same cable but are not shielded from one another. However, the whole cable is surrounded with a (braided) metal mesh shield.

Coaxial Cable

Coaxial cable consists of a single conductor running down the centre of an earthed cylindrical metal mesh shield. The conductor is usually separated from the shield by a continuous plastic insulator.[16] Coax is an "unbalanced" medium and is commonly used more for analogue than for digital communications. (Cable TV is typically reticulated on coaxial cables).

For digital communication it is used:

- In "Token Bus" (broadband) LANs where digital information is sent on an analogue carrier.
- In the public telephone network for exchange-to-exchange communication. Digital information is again sent on an analogue carrier.
- In some situations for baseband transmission over short distances such as in the IBM 3270 controller to device link.

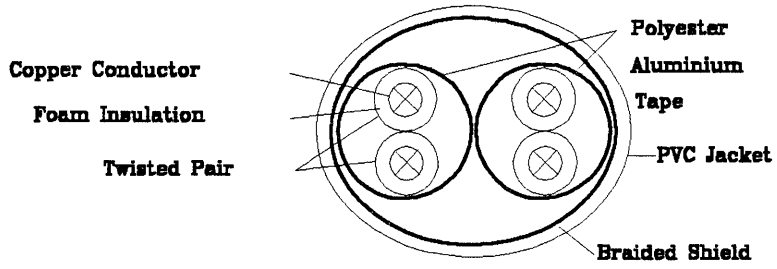


Figure 13. IBM Type 1 Shielded Twisted Pair. Each of two twisted pairs is shielded from the other by an earthed polyester aluminium tape shield. The whole is further shielded in copper braid.

2.3.2 Characteristics of Cables

Specific cable configurations vary enormously in their ability to carry a signal, in their usefulness for a particular situation and in their cost. Nevertheless they all share a common set of characteristics:

Resistance

All wire has resistance to the flow of electrical current and this is one factor in limiting the distance over which a signal may travel. The thicker the wire, the less resistance. While it reduces the signal (and also the noise) resistance does not cause distortion of the signal.

A typical resistance value for 20 gauge (.8118 mm) wire at 70 deg F is 10 ohms per thousand feet. For 26 gauge (.4094 mm) wire this figure is 40 ohms per thousand feet.

[16] Coaxial cables used for long line telephone transmission in the past did not have a continuous plastic insulator but rather small plastic insulators spaced at regular intervals. This type of cable has many better characteristics than the ones with continuous plastic but is very hard to bend around corners and to join.

Leakage

This is caused by conduction between the wires through the "insulator". This is another source of signal loss.

Inductance

Current flowing in a wire always produces a magnetic field around the wire. This field changes with the flow of current. These changes in magnetic field absorb power at some times and replace it back into the circuit at other times. This causes distortion of the digital signal. (Inductance is not all bad, since it tends to compensate for distortion caused by capacitance effects.)

Capacitance

Capacitance is the effect caused by electrostatic attraction between opposite charges in conductors which are physically close to one another. Like inductance, capacitance takes power from the circuit at some times and discharges it back into the circuit at other times - causing distortion. However, it is extremely important to understand that capacitance and inductance work in different (in simple terms, opposite) ways.

Capacitance is most influenced by the distance between conductors and the material between them. This material is called the "dielectric". Other things being equal, if wires are separated by air the capacitance is about half of what it would be if the wires were separated by PVC. This is the reason that many cables use plastic foam for an insulator.

In all cable situations *except* the "open wire" case, capacitance is "large" and is considered the major factor limiting a cable's ability to carry a signal.

The above factors can be lumped together into a single characteristic called "impedance". Impedance is defined as the ratio of voltage over current at any point on the line and is a constant for a particular cable.

A normal transmission channel will change any digital pulse sent along it (see Figure 14).

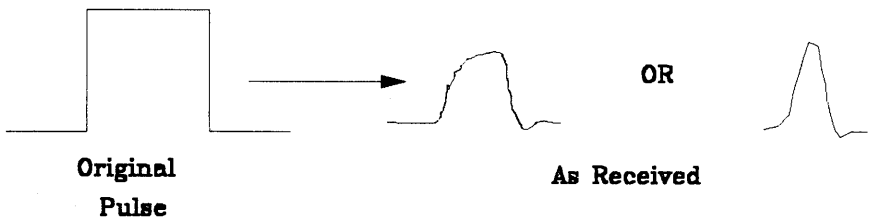


Figure 14. Distortion of a Digital Pulse by a Transmission Channel. Both of the illustrated alternatives are perfectly normal pulse shapes after transmission down a good transmission channel.

2.3.2.1 Impairments in Transmission Channels

Thermal Noise

Heat (at any temperature) causes electrons to "bounce around" at random. This causes small random variations in the signal and is usually called background noise.

Physical Environment

Cable characteristics can vary significantly with temperature sometimes in the space of an hour or so. Other environmental factors such as the presence of moisture can also change the transmission characteristics of the cable.

Impulse Noise

When wires (such as regular telephone twisted pairs) pass through a telephone exchange or pass close to other wires they can pick up short transient voltages (impulses) from the other wires in the same cable. This is caused by unintended capacitive and inductive coupling with the other wires in the cable.

In some telephone environments the major source of impulse noise is the relatively high voltage pulses (up to 50 volts) used for dialing in old style (step-by-step) telephone exchanges. Some telephone environments use as many as two thousand twisted pairs in the same cable.

In other environments, impulse noise can arise if an (unshielded) twisted pair passes close (in the same duct) to normal electrical power wiring. Whenever a high current device (such as a motor) is switched on there is a current surge on the wire. Through stray coupling, a power surge can cause impulse noise on nearby communications cabling.

Reflected Signals

If a signal traveling on a wire encounters a change in the impedance of the circuit, part of the signal is reflected back towards the sender. Of course, as it travels back to the sender, it could be reflected back in the original direction of travel if it encounters another change in impedance.

Reflected signals (sometimes called echoes) can cause a problem in a unidirectional environment because they can be reflected back from the transmitter end, but this is not always very serious.

The real problem comes when a two-wire line is used for full-duplex communication (such as at the B_ISDN "U" interface). Here reflections can cause serious interference with a signal going in the other direction.

Impedance changes over the route of a circuit can have a number of causes such as a change in the gauge or type of wire used in the circuit or even a badly made connection. However, the worst impedance mismatches are commonly caused by the presence of a "bridged tap". A bridged tap is simply where another wire (pair of wires) is connected to the link but its other end is not connected to anything. Bridged taps are quite common in some telephone environments.

Intersymbol Interference Interference.

Intersymbol interference takes place when a particular line state being transmitted is influenced by a previous line state. This is usually a result of the characteristics of the circuit causing the signal to be "smeared" so that the end of one symbol (bit or group of bits) overlaps with the start of the next.

Crosstalk

Crosstalk is when a signal from one pair of wires appears on another pair of wires in the same cable - through reactive (capacitive or inductive) coupling effects. There are two kinds of crosstalk which can have different amounts of significance in different situations. These are called "Near End Crosstalk" (NEXT) and "Far End Crosstalk" (FEXT).

The most common type of NEXT occurs when the signal transmitted from a device interferes with the signal in the other direction being received by that device. This can be a problem in the LAN environment using unshielded cable since the transmit pair and the receive pair are bound closely together in the cable.

Another type of NEXT is interference from any transmitter at the same end of the cable as the receiver being interfered with. (This is common in the telephone situation where there can be hundreds of wire pairs in the same cable.)

FEXT is caused by interference from transmissions by other devices at the far end of the cable. In most (though not all) digital signaling situations, FEXT is not a significant problem (because it happens at the transmitter end of a cable where the signal is strong relative to the interference).

2.3.3 The Subscriber Loop

The subscriber loop is the two-wire connection between the telephone exchange and customer premises. This environment was designed for analogue telephone connection and it poses some severe problems for digital communication. Nevertheless, it is of immense economic importance.

In the US there is something of the order of 80 million subscriber loop connections. To replace them (meaning digging up the road to install new ones) would cost something like $ 1,500 per connection on average. A way of making more productive use of them (such as is done in B_ISDN) could provide a large economic benefit. Even selection of "good" wire pairs to use for a special purpose or the conditioning of a nominated pair, are very labour intensive procedures and are to be avoided if possible.

There are wide differences between countries (and even within individual countries) in the characteristics of this wiring.

Maximum Length

One of the most important criteria is the length of the wire. The maximum length varies in different countries but is usually from four to eight kilometers.

Wire Thickness

All gauges of wire have been used, from 16 gauge (1.291 mm) to 26 gauge (.405 mm). The smaller gauges are typically used for short distances so that it is rare to find 26 gauge wire longer than 2½ kilometers (a little more than 8000 feet).

Material

Most installed wire is copper but some aluminium wire is in use.

Insulation

The majority of this cable uses tightly wound paper insulation, but recent installations in most countries tend to use plastic.

Twists in Cable

Telephone wire is twisted essentially to keep the two wires of a pair together when they are bundled in a cable with up to 1000 other pairs. Twists do, however, help by adding a small inductive component to the cable characteristic.

The number of twists per meter is different for different pairs in the same cable. (This is deliberately done to minimise crosstalk interference.) Also, the uniformity of twists is not generally well controlled. These have the effect of causing small irregularities in impedance which can cause reflections and signal loss due to radiation etc.

Different Gauges on the Same Connection

It is quite common to have different gauges of wire used on the same connection. Any change in the characteristics of the wire causes an impedance mismatch and can be a source of reflections.

Bridged Taps

The worst feature of all in typical subscriber loops is the bridged tap. This is just a piece of wire (twisted pair) connected to the line at one end only with the other end left unconnected. In other words, the circuit between the end user and the exchange has another (unconnected) wire joined to it somewhere along its path. This happens routinely when field technicians attach a user to a line without removing the wires that attached previous users.

In practical situations subscriber loops with as many as six bridged taps have been reported.

Bridged taps cause a large impedance mismatch (reflect a large amount of the signal) and radiate energy (causing loss of signal strength and potential problems with radio frequency emission).

Loading Coils

On typical telephone twisted pair cable, the effects of capacitance between the two conductors dominates the characteristics of the circuit and limits the transmission distance. For many years it has been a practice to counteract some of the effect of capacitance by adding inductance to the circuit. This was done (especially over longer distances) by the insertion of "loading coils" into the loop.

It is estimated that up to 25% of the subscriber loops in the US have loading coils in the circuit.

There is no available digital transmission technique which will work in the presence of loading coils. They need to be removed if the circuit is to be used for digital transmission.

2.3.4 Echo Cancellation

As mentioned previously in this discussion, echoes are a significant source of interference. This is particularly true wherever bi-directional transmission takes place on two wires (such as in the subscriber loop).

In the traditional analogue telephone environment, two-wire transmission is used from the subscriber to the nearest exchange and "four wire" (true full-duplex) transmission is used between exchanges. In this traditional environment, echoes can be a major source of annoyance to telephone users.

A historic way of handling echoes was to use "echo suppressors". An echo suppressor is a device that detects transmission in one direction (such as someone speaking) and suppresses *all* transmissions arriving from the opposite direction. Sometimes in circuits with long delays (such as in satellite circuits) echo suppressors are used at both ends. Of course, echo suppressors are useless for full-duplex transmission since they prevent it. Even in voice applications echo suppressors have problems because they "clip" speech when both parties attempt to speak at the same time and they tend to suppress low level speech as well.

A better way of handling echoes is to use an echo canceller. The basic principle of echo cancellation is shown in Figure 15.

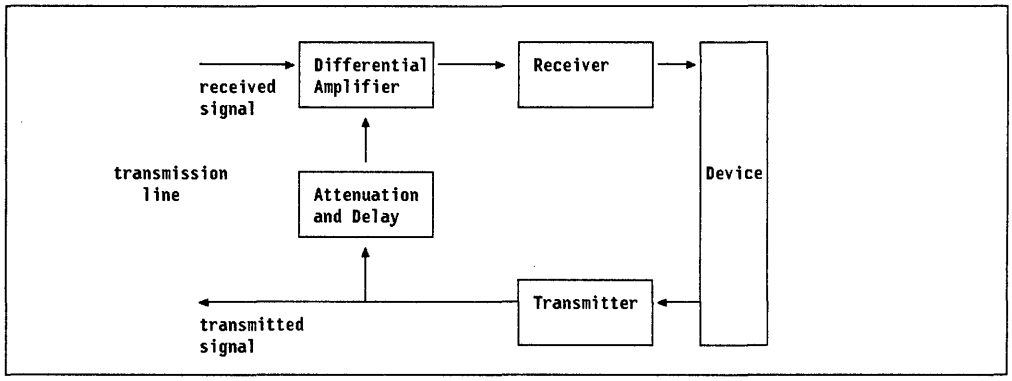


Figure 15. Concept of an Echo Canceller

The concept behind an echo canceller is that as a signal is transmitted, it is copied and put into a variable delay mechanism. The delayed signal is then reduced and subtracted from the signal being received in a differential amplifier. If the delay and signal levels are set correctly, then this device will effectively remove strong echoes. Echo cancellers may be analogue or digital in nature. This device is, however, only effective at removing echoes from a single source. If (as is typical), there are many echoes the device will only remove one of them.

When transmitting full-duplex onto a two-wire line there is a much larger potential problem than echo. It is the full signal from the transmitter. After all, the output from the transmitter is connected to the same two wires as the receiver. This problem is solved by using a device called a Hybrid Balanced Network. This is simply a number of inductive couplings organised so that the transmitter signal is cancelled out (subtracted from the signal) before the signal reaches the receiver. This technique has been used historically to split the signal from a two-wire subscriber loop onto a four-wire trunk circuit. The problem with this is that it is difficult in practice to match impedances exactly and so some of the transmitter's signal does reach th receiver (a form of near end crosstalk).

In recent times devices called "Adaptive Filters" have become practical with improvements in VLSI technology. An adaptive filter automatically adjusts itself to its environment.[17] Echo cancellers can't handle interference which is systematic but not an echo (such as crosstalk from adjacent circuits) nor can they handle the effects of impulse noise or (for that matter) multiple echoes. Adaptive filters can do a very good job of cleaning up a signal when transmitting full-duplex on a two-wire line.

Of course, echo cancellers can be digital and adaptive also, so that they can adjust automatically to the characteristics of the circuit they are attached to.

A practical system will then use:

1. An analogue hybrid to connect to the line.
2. Analogue filters for both transmit and receive to clean up the signal a bit.
3. A digital adaptive echo canceller.
4. A digital adaptive filter.

In conjunction with an appropriate line code, this system can process full-duplex data at rates up to about 200 Kbps, and half-duplex at rates in excess of 1.8 megabits per second over distances of up to 12 kilometers without the need for repeaters.

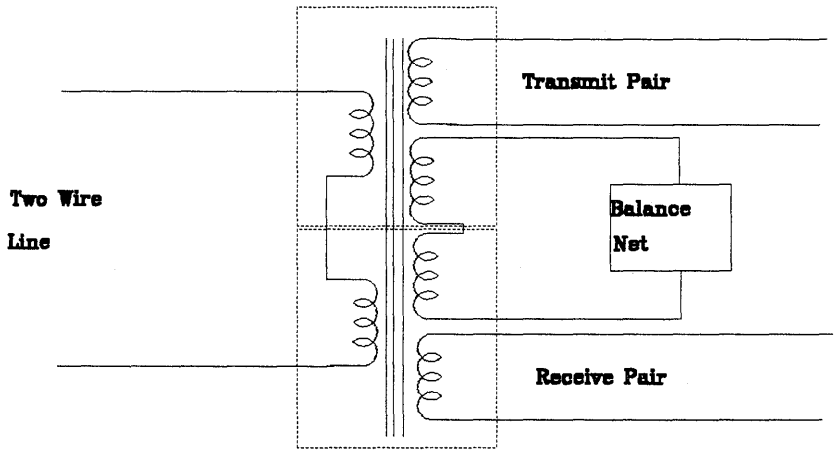


Figure 16. Hybrid Balanced Network. This device is used to convert a signal from two-wire to four-wire operation. Two transformers each consisting of at least three tightly coupled windings and an impedance balancing network are used. The balance network is adjusted to match the impedance of whatever device is connected to the two-wire line. With careful impedance matching, the device can reduce the level of the transmitted signal reaching the receive pair by as much as 50 dB.

[17] An excellent account of adaptive filtering in the loop plant is give in *Waring D.L et. al. 1991*.

2.3.5 Digital Transmission State of the Art

Many people think that because the use of fibre optics is growing very rapidly that there is no progress being made in digital transmission on copper media. Nothing could be further from the truth.

Better circuitry (higher density, higher speed, lower cost) enables us to use much more sophisticated techniques for digital transmission than have been possible in the past. As mentioned above, there is a vast number of copper "subscriber loops" installed around the world and there is a big incentive to get better use out of them.

Earlier in this chapter, and later in section 6.1.3, "ISDN Basic Rate" on page 105, we have discussed the principles of basic rate ISDN. This was an enormously successful research effort. So much so that there are a number of submissions before the American National Standards Institute (committee T1.E1.4) aimed at increasing the link speeds even further.

Two projects are in progress:

1. Asymetric Digital Subscriber Line (ADSL).

   This project is aimed at producing a standard for use of a two-wire subscriber loop for 1.544 Mbps in one direction with a much slower (say 64 Kbps) channel in the opposite direction. Many potential services, for example for image retrival, do not require high speed in both directions.

2. High-bit-rate Digital Subscriber Lines (HDSL).

   This project is examining a number of alternative architectures for full-duplex transmission on *unconditioned* subscriber loop connections at speeds up to 1.544 Mbps.

This is a considerable challenge! The unshielded telephone twisted pair of the subscriber loop has many impairments and as the transmission rate increases the losses on the line increase exponentially. The subscriber loop is a very limited channel. Existing "T1 and E1" circuits use selected pairs in one direction only (two pairs are used) and they are amplified every mile.

There are four HDSL loop architectures under consideration:

Dual Duplex (Two-Pair Full-Duplex)

Two subscriber loops are used (that is, four wires) to carry a data rate of 1.544 Mbps. This is not the same as existing T1s because it is proposed to use unconditioned loops without repeaters.

Each wire pair carries 784 Kbps full-duplex using an echo cancelled hybrid transmission method similar to that used for B_ISDN.

Dual Simplex (Two-Pair Simplex)

Two pairs are used as above, but each pair carries a full 1.544 Mbps signal in one direction only.

One-Pair 1.5 Mbps Full-Duplex

Full-duplex T1 over a single unconditioned subscriber loop pair. This is listed "for further study" (for now, it is too difficult).

Provisioned Single Loop 768 Kbps Transport

This is really like a half of the dual duplex case, albeit that it will probably be quite different in detailed operation. The user interface is

proposed to be standard T1 but with 12 channels disabled. That is, a 24 slot T1 frame with 12 of the slots disabled and filled with idle characters.

Three approaches to transmission are under consideration:

2B1Q Line Code

In principle this is the same technique as is used for the US version of Basic Rate ISDN, although it is a little different in detail. 2B1Q line code as described in section 2.2.12.1, "2-Binary 1-Quaternary (2B1Q) Code" on page 29, is used in conjunction with advanced echo cancellation and adaptive filtering techniques (see section 2.3.4, "Echo Cancellation" on page 37).

Carrierless AM/PM (CAP)

This architecture has been suggested for the "dual duplex" configuration.

The dual duplex configuration is attractive because by splitting the signal over two parallel subscriber loops and running the links (loops) in a coordinated way you can cancel a significant proportion of the NEXT.

Multicarrier (Discrete Multitone - DMT)

The principle behind the discrete multitone system has been in use for many years in modems. It is *not* a digital baseband system. Rather, a single data stream is broken up into a number of components and sent on several separate carriers (of different frequency) within the usable frequency band. The usable bandwidth is divided up into a set of subchannels that may be modulated and demodulated independently. So the full bandwidth is used by different frequency signals each carrying a part of the total.

But the actual system used is a lot more complex than this.

- A digital bit stream is constructed from the data to be transmitted.
- This stream is then broken up into its underlying frequency components. (A "Fast Fourier Transform" performs this function.)
- The output is a set of parameters that describe the original signal.
- These parameters are then coded and the output sent on the multicarrier system.

All three of these systems are quite complex. What is happening is that the signal is being coded optimally to fit into the available frequency bands.

A number of papers on this are referenced in the bibliography.

2.3.6 LAN Cabling with Unshielded Twisted Pair

2.3.6.1 The FCC versus the Laws of Physics

The two fundamental issues in telecommunications copper cable choice are easy to express. First of all, cable utility is fundamentally limited by the laws of physics. The second fundamental limitation is that a premises's wiring must meet the laws of man, notably regulatory requirements governing radiation of electronic signals into the environment. These two sets of laws often interact in ways that are disturbing to the transmission engineer. In other words, not everything that is possible in the realm of physical law is permitted by regulatory law.

All other things being equal, we know that as data rate increases the drive distance available on copper wire decreases approximately proportionally to the square roots of the data rates. So a 16 Mbps signal can travel only one-half of the distance of a 4 Mbps signal using a copper wire with constant specifications. However, all other things are almost never equal among signaling methods of devices using different protocols, implementations, or data rates. All of these considerations are further complicated by the fact that increasing data rates cause increasing radiation. The regulatory agencies reasonably set an absolute standard for radiation emmission that does not vary with the data rate of the signal.

Recently, cable manufacturers have introduced a variety of high performance UTP cables that are manufactured to much more rigid specifications than common UTP used for voice transmission in internal building wiring. These cables are expected to provide a much more stable platform for high speed data transmission than earlier UTP cables.

2.3.6.2 UTP for 16 Mbps Token-Ring

In a recent (1991) submission to the IEEE 802.5 Token-Ring Network committee, IBM presented the results of a study into 4 and 16 Mbps transmission on unshielded twisted pair cable of various specification.

Using good quality telephone twisted pair (IBM Type 3 media),[18] the study found that this cable allows a high level of crosstalk and high attenuation that together place severe restrictions on both drive distance and number of stations attached.

On this (TTP) cable, near end crosstalk (NEXT) can be as much as -23dB at 16 megahertz, relative to the transmitted signal level. The signal attenuation on this cable can be as high as 13dB over a transmission distance of 100 meters. On a token-ring, these two phenomena act in concert to degrade the signal quality at the station receivers. Consequently, for a 100-meter distance, the signal-to-noise ratio due solely to near end crosstalk can be as low as 10dB (or about 3 to 1) on cabling that conforms to the IBM Type 3 media (EIA-568) specification.

A conclusion of the study was that using standard IBM components and a prototype media filter to meet the FCC requirements, 16 Mbps transmission is possible in a single ring of up to 72 devices with lobe cables that do not exceed 30 meters. This very restrictive solution was improved by using a prototype

[18] Similar to the EIA-568 Commercial Building Telecommunications Wiring Standard.

active 8230 Lobe Attachment Module that utilizes equalization and linear amplification to offset the degradation associated with transmission on UTP.

The study further examined the operation of 16 Mbps token ring over cable conforming to a high grade UTP specification. The salient information in that specification for the purposes of this discussion is that the cable is to meet the following nominal characteristics when measured on a 1000-foot length of cable at 20 degrees Centigrade using a 16 MHz signal: a maximum of 27dB of attenuation, and near end crosstalk (NEXT) that is at least 39dB below the transmitted signal. This specification would allow the use of a wide variety of cables that are currently being marketed as high performance, super, or data grade UTP. A number of cables meeting this specification were tested. The study found that this cable will allow significant improvement in drive distance and number of stations attached over that possible with standard UTP.

The signal-to-noise ratio due solely to crosstalk, on a 100-meter lobe using this cable is about 30dB, or, put another way, the noise due to crosstalk is less than 3% of the normal data signal amplitude at the receiver. This amount of near-end crosstalk is negligible in comparison to the EIA-568 cable case where lobe lengths of only 30 meters exhibit a near end crosstalk amplitude approaching 40% of the received signal amplitude based on a signal-to-noise ratio of about 19dB.

This suggests that while 30 meter lobe lengths of Type 3 media are feasible in a ring using existing IBM components and prototype filters, the parameter variations are so great that, if many stations or longer lobe lengths are required, high performance UTP is clearly a superior choice.

Testing showed that, using standard IBM components and a prototype media filter to meet the FCC requirements, 16 Mbps transmission is possible in a single ring of up to 72 devices with lobe cables that do not exceed 90 meters.

The study examined ways of improving the situation further. Using a prototype active 8230 Lobe Attachment Module, testing showed that lobe lengths of up to 100 meters in a 72-station ring are possible. It is believed that the prototype Lobe Attachment Module can be improved to permit additional station attachment.

This doesn't solve all the problems of UTP however. As far as the issue of Electromagnetic Compatibility (EMC) is concerned, neither of these choices offers the protection provided by shielded twisted pair (STP) by the mere existence of the shielding. In order to prevent unacceptable levels of EMC, radiation filters are needed at every station and the cost of these must be considered in the overall network media cost. In addition, when using UTP, impulse noise and crosstalk from nearby cables can be a source of significant performance degradation. (Do *not* route UTP cable close to the power cables in an elevator well!)

Many users are requesting the ability to run 100 megabits per second FDDI token rings over UTP cable (in fact even TTP cable). This will prove a considerable (but doubtless not insurmountable) challenge.

Chapter 3. An Introduction to Fibre Optical Technology

The idea of using light to send messages is not new. After all, they used fires for signaling during wars in biblical times and native peoples have used smoke signals for thousands of years. The idea of sending light through a glass fibre as a means of communication goes back to Alexander Graham Bell. However, widespread use had to wait for better glasses and much lower cost electronics to make it possible. Over the decade of the 1980s optical communication in the public communications networks went from being a curiosity to the normal accepted way of doing things.

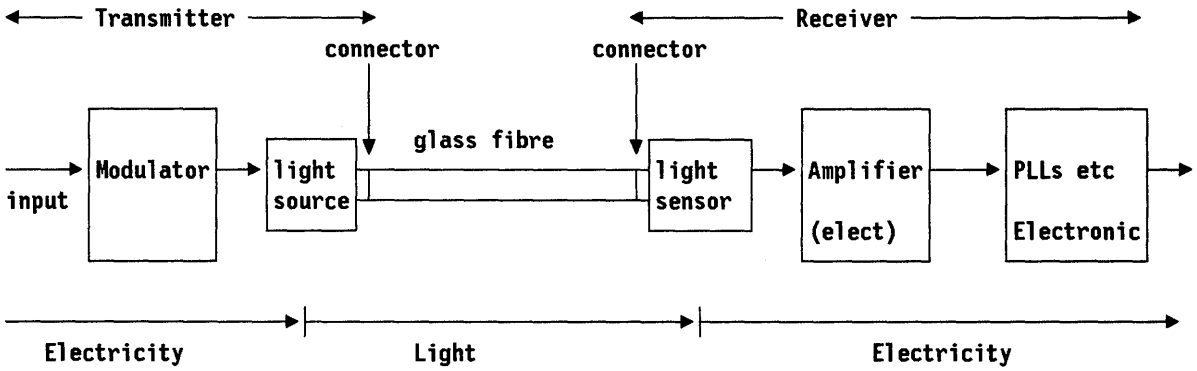


Figure 17. Optical Transmission - Schematic

3.1.1 Concept

The basic components of an optical communication system are shown in Figure 17, above.

- A serial bit stream in electrical form is presented to a modulator, which encodes the data appropriately for fibre transmission.
- A light source (a Laser or Light Emitting Diode - LED) is driven by the modulator and the light focused into the fibre.
- The light travels down the fibre (during which time it may experience dispersion and loss of strength).
- At the receiver end the light is fed to a detector and converted to electrical form.
- The signal is then amplified and fed to a detector, which isolates the individual bits and their timing.
- The timed bit stream so received may then be fed to a using device.

Optical communication has many well known advantages:

Weight and Size

Fibre cable is significantly smaller and lighter than electrical cables to do the same job. In the wide area environment a large coaxial cable system can easily involve a cable of several inches in diameter and weighing many pounds per foot. A fibre cable to do the same job could be less than one half an inch in diameter and weigh a few ounces per foot.

This means that the cost of laying the cable is dramatically reduced.

Material Cost

Fibre cable costs significantly less than copper cable for the same transmission capacity.

Information Capacity

The data rate of systems in use in 1992 is generally 150 or 620 Mbps on a single (unidirectional) fibre. This is very high in digital transmission terms.

In telephone transmission terms the very best coaxial cable systems gave about 2,000 analog voice circuits. A 150 Mbps fibre connection gives just over 2,000 digital telephone (64 Kbps) connections. But the 150 Mbps fibre is at a very early stage in the development of fibre optical systems. The coaxial cable system with which it is being compared is much more costly and developed to its fullest extent.

But fibre technology is in its infancy. Researchers have trial systems in operation that work at speeds of five gigabits per second and the limits seem almost endless.

No Electrical Connection

Such an obvious point seems almost silly to raise, but electrical connections have problems.

- In many electrical systems there is always the possibility of "ground loops" causing a serious problem. When you communicate electrically you often have to connect the grounds to one another. This is discussed in section 2.2.3, "Coupling to a Line" on page 16.
- Optical connection is very safe. Electrical connections always have to be protected from high voltages because of the danger to people touching the wire.

No Electromagnetic Interference

Because the connection is not electrical you can neither pick up nor create electrical interference. This is one of the reasons that optical communication has so few errors. There are very few sources of things that can distort or interfere with the signal.

In a building this means that fibre cables can be placed almost anywhere electrical cables would have problems. For example near a lift motor or in a cable duct with heavy power cables. In an industrial plant such as a steel mill this gives much greater flexibility in cabling than previously available.

In the wide area networking environment there is much greater flexibility in route selection. Cables may be located near water or power lines without risk to persons or equipment.

Distances Between Repeaters

In long line transmission cables now in use by the telephone companies, the repeater spacing is typically 24 miles. This compares with 8 miles for the previous coaxial cable electrical technology. The number of required repeaters and their spacing is a major factor in system cost.

Some systems planned in 1992 have much greater repeater spacings brought about by better transmission techniques.

Open Ended Capacity

Data transmission speed may be changed (increased) whenever a new technology becomes available. All that must be done is change the equipment at either end and change the repeaters.

Better Security

It is possible to tap fibre optical cable. But it is very difficult to do and the additional loss caused by the tap is relatively easy to detect. There is an interruption to service while the tap is inserted and this can alert operational staff to the situation. In addition, there are fewer access points where an intruder can gain the kind of access to a fibre cable necessary to insert a tap.

Insertion of active taps where the intruder actually inserts a signal is even more difficult.

There are some disadvantages however:

Joining Cables

Joining fibres is still a skilled task which requires precision equipment. It is particularly difficult to do outdoors in very low temperatures such as in the North American or European winter.

In the last few years patch cable systems for fibres with thicker cores ("multimode fibres") have improved things a lot but installation costs are still high.

The cost of coupling a fibre to an electronic adapter is still quite high. In the LAN environment, 1992 FDDI adapter market prices in the US are somewhere around $ 8,000 per ring connection. Some suppliers are suggesting that FDDI adapters operating over a copper medium will soon be available for less than half of the cost of their fibre counterpart. The cost difference is in the optical tranceiver and the coupling.

No Stable Standards Yet

This is nobody's fault. Development is happening so quickly, and getting worldwide agreement to a standard is necessarily so slow that standards setting just can't keep up. Things are improving considerably and very quickly, however. Cable sizes and types are converging toward a few choices, albeit the way they are used is still changing almost daily.

Standards agreement has begun in the LAN world for FDDI and in the wide area world for Sonet/SDH.

Optical Transmission Only

Until very recently there was no available optical amplifier. The signal had to be converted to electrical form and put through a complex repeater in order to boost its strength. Recently optical amplifiers have emerged and look set to solve this problem (see 3.1.2.10, "Optical Amplifiers" on page 53).

However, optical logic processing and/or switching systems seem to be a few years off yet.

Gamma Radiation

Gamma radiation comes from space and is always present. It can be thought of as either a high energy X-ray or a low energy β particle. Gamma radiation can cause some types of glass to emit light (causing

interference) and also gamma radiation can cause glass to discolor and hence attenuate the signal. These effects are minimal, however.

Very high voltage electrical fields also affect some glasses in the same way as gamma rays. One proposed route for fibre communication cables is wrapped around high voltage electrical cables on transmission towers. This actually works quite well where the electrical cables are only of 30,000 volts or below. Above that, (most major transmission systems are many times above that) the glass tends to emit light and discolor. Nevertheless, this is a field of current research - to produce a glass that will be unaffected by such fields. It is a reasonable expectation that this will be achieved within a very few years.

Sharks Eat the Cable(?)

In the 1980s there was an incident where a new undersea fibre cable was broken on the ocean floor. Publicity surrounding the event suggested that the cable was attacked and eaten by sharks. It wasn't just the press, this was a serious claim. It was claimed that there was something in the chemical composition of the cable sheathing that was attractive to sharks!

Other people have dismissed this claim as a joke and suggest that the cable was badly laid and rubbed against rocks. Nevertheless the story has passed into the folklore of fibre optical communication and some people genuinely believe that sharks eat optical fibre cable.

Most people evaluate the advantages as overwhelming the disadvantages for most environments. But advantages and disadvantages need to be considered in the context of the environment in which the system is to be used. The types of fibre systems appropriate for the LAN environment are quite different from those that are optimal in the wide area world. This will be shown in the remainder of this chapter.

3.1.2 Transmitting Light through a Fibre

When light is transmitted down a fibre, the most important consideration is "what kind of light?". The electromagnetic radiation that we call light exists at many wavelengths.[19] These wavelengths go from invisible infrared through the visible spectrum to invisible ultraviolet.

In order to signal with light we have to be able to create it, to transmit it down a fibre and to detect it at the other end. But there are many different ways of creating light, several ways of detecting it, and most of them are not compatible at all with fibre transmission.

3.1.2.1 Light Transmission Down a Fibre

If a short pulse of ordinary light from an incandescent bulb is sent down a narrow fibre it will emerge (depending on the distance) as a "fuzzy" pulse. The reasons for this are as follows:

1. Ordinary incandescent light contains a mixture of wavelengths (much of it in the invisible, infrared part of the spectrum). Different wavelengths travel at different speeds in a fibre. Thus some wavelengths arrive before others.

[19] Another way of saying this is that light has many frequencies or colours.

2. Light "bounces around" within the fibre. That is it can travel over many different paths. Because some paths are shorter than others, the signal is dispersed even if it is all of one frequency. This is illustrated in Figure 19 on page 49 under the heading "Multimode Step Index".
3. Glass absorbs different frequencies of light at very different rates. This is called "chromatic dispersion". So a signal that starts with one set of spectral characteristics will end up at the other end of the fibre with very different characteristics.

None of these effects are helpful to engineers wishing to transmit information over long distances on a fibre. But much can be done about it.

1. Lasers transmit light at one wavelength only; furthermore, the light rays are parallel with one another and in phase. Light Emitting Diodes (LEDs) that emit light within only a very narrow range of frequencies can be constructed. So the problem of dispersion due to the presence of multiple wavelengths is avoided.
2. If you make the fibre thin enough, the light will have only one possible path - straight down the middle. Light can't disperse over multiple paths because there is only one. This kind of fibre is called monomode or single mode fibre and is discussed in some detail later.
3. The frequency of light used in a particular application may be carefully chosen to avoid using frequencies that are absorbed by the fibre.

From the above we may conclude that there are four critical things about any fibre transmission system:

1. The characteristics of the fibre itself. Its thickness, its refractive index, its absorption spectrum.
2. The wavelength of light used.
3. The type and characteristics of the device used to create the light (Laser or LED).
4. The type and characteristics of the device used to detect the light.

But there is another even more important parameter. That is, how the signal is modulated (systematically changed to encode a signal). There are many potential ways to do this and they too are discussed later.

3.1.2.2 Absorption Characteristics of Glasses

Figure 18 on page 48 shows the absorption spectrum of two glasses in the infrared range. Light becomes invisible (infrared) at wavelengths longer than 730 nm There are a wide range of glasses available and characteristics vary depending on the chemical composition. Over the past few years the transmission properties of glass have been improved considerably. In 1970 the "ballpark" attenuation of a silicon fibre was 20 dB/km. By 1980 research had improved this to 1 dB/km. In 1990 the figure is 0.2 dB/km. As the figure shows, absorption varies considerably by frequency and the two curves show just how different the characteristics of different glasses can be.

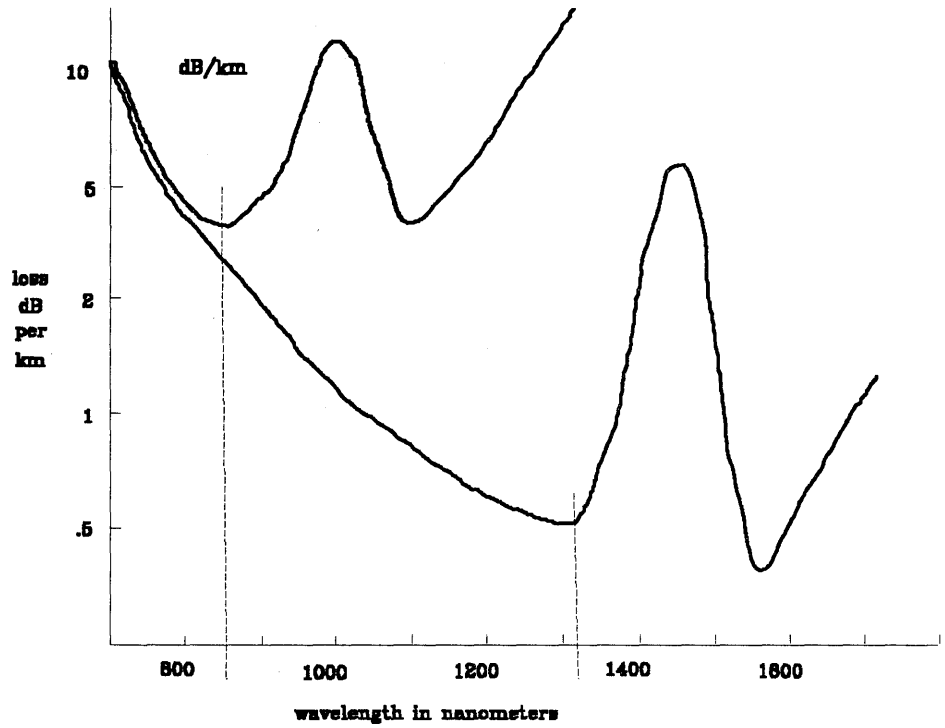


Figure 18. Typical Fibre Infrared Absorption Spectrum. The upper left-hand curve represents the characteristics of Silicon Dioxide (SiO_2) glass. The lower (much better) curve shows a curve for a glass made from a mixture of Germanium Dioxide (GeO_2) and Silicon Dioxide. The peak at around 1450 nm is due to the effects of traces of water in the glass.

The conclusion that can be drawn from the absorption spectrum is that some wavelengths will be significantly better for transmission purposes than others. For ordinary silica glass the wavelengths of 850 nm and 1100 nm look attractive. For the better quality Germanium Dioxide rich glass, wavelengths of around 1300 nm and 1550 nm look attractive. All this depends on finding light sources that will operate in the way we need at these wavelengths.

The wavelength used is an extremely important defining characteristic of the optical system.

3.1.2.3 Fibre Geometry

In Figure 19 on page 49, the top part of the picture shows the operation of "multimode" fibre. There are two different parts to the fibre. In the figure, there is a core of 50 microns (µm) in diameter and a cladding or 125 µm in diameter. (Fibre size is normally quoted as the core diameter followed by the cladding diameter. Thus the fibre in the figure is identified as 50/125.) The cladding surrounds the core. The cladding glass has a different refractive index than that of the core, and the boundary forms a mirror.

> *This is the effect you see when looking upward from underwater. Except for the part immediately above, the junction of the water and the air appears silver like a mirror.*

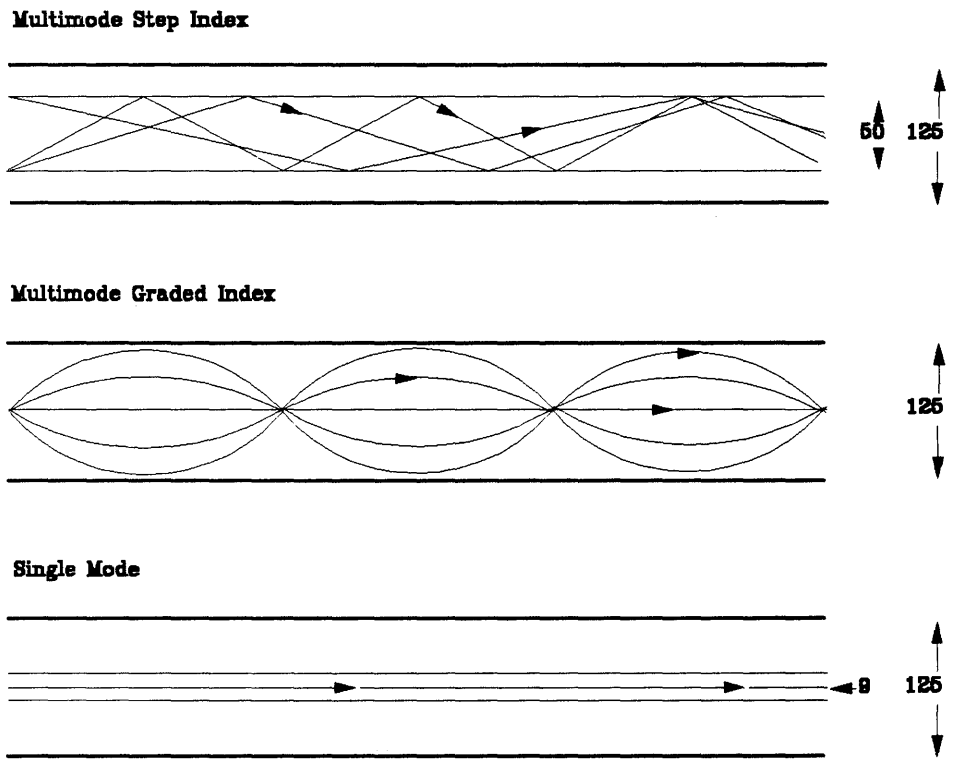


Figure 19. Fibre Types

Light is transmitted (with very low loss) down the fibre by reflection from the mirror boundary between the core and the cladding.

The expectation of many people is that if you shine a light down a fibre, then the light will enter the fibre at an infinitely large number of angles and propagate by internal reflection over an infinite number of possible paths. This is not true. What happens is that there is only a finite number of possible paths for the light to take. These paths are called "modes" and identify the general characteristic of the light transmission system being used. Fibre that has a core diameter large enough for the light used to find multiple paths is called "multimode" fibre. (For a fibre with a core diameter of 62.5 microns using light of wavelength 1300 nm the number of modes is around 2,400.)

The problem with multimode operation is that light traveling by different paths arrives at the other end at different times and so the pulse tends to spread out as it travels through the fibre. This restricts the distance that a pulse can be usefully sent over multimode fibre.

One way around the problem of multimode fibre is to do something to the glass such that there is no longer any defined boundary between the core and the cladding. The refractive index changes gradually from one to the other. This system causes the light to travel in a wavelike motion but light in different modes travels in glass of different refractive index at different speeds. The net is that the light stays together as it travels through the fibre and allows transmission for longer distances than does regular multimode transmission. This type of fibre is called "Graded Index" fibre.

If the fibre core is very narrow compared to the wavelength of the light in use then the light cannot travel in different modes and thus the fibre is called "single mode" or "monomode". It seems obvious that the longer the wavelength of light in use, the larger the diameter of fibre we can use and still have light travel in a single mode. The core diameter used in a typical monomode fibre is nine microns.

The big problem with fibre is joining it. The narrower the fibre, the harder it is to join and the harder it is to build such things as patch cables or plugs and sockets. Single mode fibre has a core diameter of around 9 µm. Multimode fibre can have many core diameters but in the last few years the core diameter of 62.5 µm in the US and 50 µm outside the US has become predominant.

3.1.2.4 Light Sources

There are two kinds of device that are used as light sources: Lasers and LEDs (Light Emitting Diodes).

Lasers

Laser stands for Light Amplification by the Stimulated Emission of Radiation. Lasers produce far and away the best kind of light for optical communication.

- Laser light is a single wavelength only. This is related to the molecular characteristics of the material being used in the laser.

  It is formed in parallel beams and is in a single phase. That is, it is "coherent".

- Lasers can be controlled very precisely (the record is a pulse length of 0.5 femto seconds[20]).
- Lasers can produce relatively high power. In communications applications, lasers of power up to 20 milliwatts are available.
- Because laser light is produced in parallel beams, a high percentage (50% to 80%) can be transferred into the fibre.
- Most laser systems use a monitor diode to detect back facet power for automatic power control. This can be used for diagnostic purposes.

There are disadvantages however. Lasers have typically been quite expensive by comparison with LEDs. (Recent development has helped this a lot.) In addition, the wavelength that a laser operates on is a characteristic of the material used to build the laser. You can't just say "I want a laser on x wavelength", or rather you can say it all you like. Lasers have to be individually designed for each wavelength they are going to use. (Tunable lasers exist but these are not yet available for communications type usage.)

Light Emitting Diodes (LEDs)

- LEDs are very low in cost (perhaps 1/10th that of a laser).
- The maximum light output is a lot lower than a laser (about 100 microwatts).

[20] 10^{-15} seconds.

- LEDs do not produce a single light frequency but rather a narrow band of frequencies. The range of the band of frequencies produced is called "spectral line width" and is typically somewhere between 40 and 120 nanometers.
- The light produced is not directional or coherent. This means that you have to have a lens to focus the light onto the end of a fibre. LEDs are not suitable for use with single mode fibre for this reason (it is too hard to get the light into the narrow core).
- LEDs cannot produce pulses short enough to be used at gigabit speeds. However, systems using LEDs operate quite well at speeds of up to around 300 Mbps.

3.1.2.5 Light Detectors

A number of different kinds of devices are used for light detection.

PIN Diodes

PIN diodes convert light directly to electric current. An ideal PIN diode can convert one photon to one electron of current. This means that the current output from a PIN diode is very small and an external amplifier is needed before the signal can be dealt with by a receiver.

Avalanche Photo Diodes (APDs)

APDs use a similar principle to the old "photomultiplier" tubes used in nuclear radiation detection. A single photon acting on the device releases a single electron which then releases more electrons as it travels through the device. APDs typically have an internal amplification of between 10 and 100 times.

The multiplication effect means that an APD can be very sensitive. The negative side is that APDs are inherently noisy as the multiplier effect applies to all free electrons including those made free by ambient heat. In most long distance wide area applications the internal gain is more important than other factors and the APD is the usual device used in long distance applications.

Phototransistors

Phototransistors are amplifiers where the amplification is controlled by the amount of light striking the device. These are much lower in noise and have a higher output than APDs but are less responsive than either APDs or PIN diodes.

Phototransistors are occasionally built as part of an integrated circuit. In this configuration they are referred to as "Integrated Preamplifier Detectors" (IPDs).

3.1.2.6 Making the Light Carry a Signal

In order to make light carry a signal you have to introduce systematic variations that represent the signal. This is called modulation. Then when the light is received you must decode it in such a way as to reconstruct the original signal.

All current optical transmission systems encode the signal as a sequence of light pulses in a binary form.

Sometimes this is described as Amplitude Modulation comparing it to AM radio for example. In fact the technique is nothing like AM radio.[21] In AM, the amplitude of the signal is continuously varied and the receiver recovers the signal from the variations. But in an optical system, it is much more like digital baseband transmission in the electronic world. The signal is there or it isn't, beyond this the amplitude of the signal doesn't matter.

Most digital communications systems using fibre optics use NRZ encoding. This means that when you have a pulse of light this means a "1" bit and when an expected pulse is absent this is a "0" bit. This is discussed in section 2.2.1, "Non-Return to Zero (NRZ) Coding" on page 13.

It is difficult to control (modulate) the frequency of a laser or an LED. However, frequency modulated light is being produced in laboratories. This leads to the hope of being able to use frequency modulation (FM) in fibre optical systems. FM has a major advantage. With FM, the receiver "locks on" to the signal and is able to detect signals many times lower in amplitude than AM detectors can use. This translates to greater distances between repeaters and lower cost systems. In addition FM promises much higher data rates than the pulse systems currently in use.

3.1.2.7 Directional Transmission

It is "possible" to use a single fibre for transmission in two directions at once. With very complex electronics it may even be possible to use the same light wavelength at least over short distances. But this is all far too much trouble. Tapping a fibre in such a way as to allow a receiver and a transmitter access to the fibre simultaneously is difficult and there is significant loss in the received signal. Experimental systems have been constructed that do this using a different wavelength in each direction.

In practical fibre optical transmission systems in 1992, fibre is a unidirectional medium. Two fibres are needed for bidirectional transmission. Given the size of a single fibre and the number that can conveniently be integrated into a cable, this looks certain to be the predominant mode of operation for the foreseeable future.

3.1.2.8 Joining Fibre Cables

The diameter of the core in an optical fibre is very small and any irregularity (such as a join) can result in significant loss of power. To get the maximum light transfer from the cut end of one fibre into another, both ends must be cut precisely square and polished flat. They must then be butted together so that there is minimal air (or water) between the butted ends and these ends must match up nearly exactly. In a practical situation outside the laboratory this is very difficult to do.

In the early days of optical data communication (1983), IBM specified an optical cable for future use which was 100/140 µm in diameter. The 100 micron core is very wide and is certainly not the best size for communication. However, at the time it was the best specification for making joints in the field. (This specification is still supported by some systems - including FDDI.)

---

[21] It is possible to modulate a laser signal in exactly the same way as regular AM radio. Researchers have used fibre communications to carry an analog television signal amplitude modulated onto optical fibre.

A cable join is a type of weld. The cable ends are cut, polished, butted up to one another and fused by heat. (Incidentally, with some silica fibres you need quite a high temprature - much higher than the melting point of ordinary soda glass.)

In data communication situations it is highly desirable to be able to change configurations easily. This means that we want to plug a cable into a wall socket and to conveniently join sections of cable. To do this connectors exist. They hold the fibre ends in exact position and butt them together under soft pressure to obtain a good connection. But this all depends on the precision to which connectors can be machined. Most mechanical devices are machined to tolerances much greater than the width of a fibre - so connectors are difficult to manufacture and hard to fit. This all results in a relatively high cost.

Expect to loose quite a lot of light (3 dB) at every connector. This is worse in the situation where two fibres of different diameters are being joined. It is common (for example in the IBM Escon channel system), for fibres with a 62.5 µm core to be connected to fibres with a 50 µm core. Loss of light in this situation is unavoidable.

3.1.2.9 Repeaters

Until the commercial availability of optical amplifiers in the last few months, the only way to boost an optical signal was to convert it to electrical form, amplify or regenerate it, and then convert it to optical form again. In the last few months optical amplifiers have become commercialy available (see section 3.1.2.10, "Optical Amplifiers").

Boosting the signal electrically either involves a simple amplifier or a more complex repeater. An amplifier just takes whatever signal it recieves and makes it bigger. This includes the noise and whatever dispersion of the signal that has already taken place. Amplifiers are however, simpler, cheaper and not sensitive to the coding used on the fibre. They are uncommon in fibre systems but they are used.

Repeaters have been the method of choice for boosting an optical signal. (Electronic repeaters were discussed in section 2.2.6.3, "Repeaters" on page 22.) A repeater is a full receiver which reconstructs the bit stream and its timing. This bit stream and timing is used to drive a transmitter. This means that the repeated signal has all dispersion and noise removed. Repeaters are more complex and more costly than simple amplifiers however.

In mulitmode systems where dispersion is the main distance limiting factor, electronic repeaters will continue to be the major way of boosting a signal. In single mode systems over long distances where dispersion isn't a problem, optical amplifiers look set to replace repeaters as the device of choice.

3.1.2.10 Optical Amplifiers

In all optical cable systems installed up to 1991, the signal was regenerated (about every 50 kilometers) by using a repeater.[22] Repeaters receive the old signal, retrieve the digital bit stream and then generate a new one. Thus noise and distortion picked up along the transmission path are completely removed by

[22] Repeaters in electrical communication systems are discussed in section 2.2.6.3, "Repeaters" on page 22.

a repeater; whereas, when an amplifier is used, these components are amplified along with the signal.

Repeaters are a problem in an optical system. The optical signal must be converted to electrical form, passed through the repeater and then converted to optical form again. The repeater is complex and subject to failure. Also, optical systems suffer from far fewer sources of signal interference, noise and distortion, than do electrical ones. When a signal on a single mode fibre arrives at its destination it is a lot weaker, but for practical purposes the signal is unchanged. So a device which just amplifies the signal will do just as well (or better) than a repeater in an optical system.

In the late 1980s a number of researchers around the world succeeded in developing an optical amplifier which is now (1992) beginning to go into commercial service. This device amplifies the signal *without* the need to convert the signal to electrical form - it is a wholly optical device. (Albeit, it is electrically powered.)

This is very significant because the amplifier is much less prone to failure than an electrical repeater, operates at almost any speed and is not dependent on the digital characteristics (such as the code structure) in the signal. It also costs significantly less. Many people believe that this device has begun a "new generation" in optical systems.

The device itself is illustrated below:

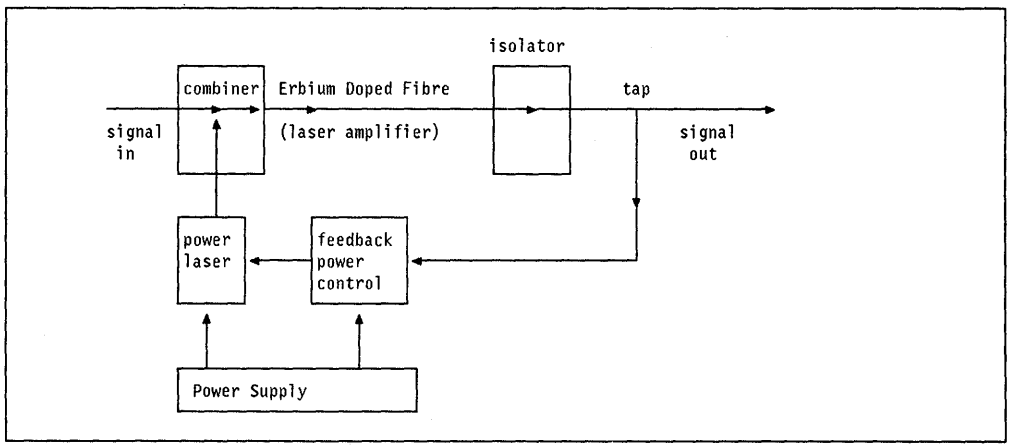


Figure 20. Erbium Doped Optical Fibre Amplifier. Although the device is powered electrically, the amplification process is totally optical and takes place within a short section of rare earth doped, single mode fibre.

The amplifier itself is simply a short (a few feet) section of fibre which has a controlled amount of a rare earth element (Erbium) added to the glass. This section of fibre is, itself, a laser.

The principle involved here is just the principle of a laser and is very simple. Atoms of Erbium are able to exist in several energy states (these relate to the alternative orbits which electrons may have around the nucleus). When an Erbium atom is in a high energy state, a photon of light will stimulate it to give up some of its energy (also in the form of light) and return to a lower energy (more stable) state. This is called "stimulated emission". "Laser" after all is an acronym for "Light Amplification by the Stimulated Emission of Radiation".

To make the principle work, you need a way of getting the Erbium atoms up to the excited state. The laser diode in the diagram generates a high powered (10 milliwat) beam of light at a frequency such that the Erbium atoms will absorb it and jump to their excited state. (Light at 980 or 1,480 nanometer wavelengths will do this quite nicely.) So, a (relatively) high powered beam of light is mixed with the input signal. (The input signal and the excitation light must of course be at significantly different wavelengths.) This high powered light beam excites the Erbium atoms to their higher energy state. When the photons belonging to the signal (at a different frequency which is not absorbed by Erbium) meet the excited Erbium atoms, the Erbium atoms give up some of their energy to the signal and return to their lower energy state.

The significant thing here is that Erbium *only* absorbs light (and jumps to a higher energy state) if that light is at one of a very specific set of wavelengths. Light at other wavelengths takes energy from the Erbium and is amplified.

So the device works this way. A constant beam of light (feedback controlled) at the right frequency to excite Erbium atoms is mixed with the input signal. This beam of light constantly keeps the Erbium atoms in an excited state. The signal light picks up energy from excited Erbium atoms as it passes through the section of doped fibre.

The optical amplifier has the following characteristics:

- It is significantly simpler than a repeater and will have a much longer mean time to failure.
- It is significantly lower in cost than a repeater.
- It will operate at almost any speed.
- It will amplify many different wavelengths simultaneously (with some small limitations).
- It doesn't need to understand the digital coding. Both amplitude (pulse) modulation and coherent (frequency modulated) light is amplified.

  This means that if amplifiers are installed in a long undersea cable for example, at some later time the transmission technique used on the cable may be changed without affecting the amplifiers.
- There is effectively no delay in the amplifier. The delay is only the time it takes for the signal to propagate through a few feet of single mode fibre.
- There is one drawback. The signal cannot be allowed to become too weak before reamplification. This results in the need to space amplifiers roughly every 30 kilometers where current systems place repeaters about every 50 kilometers.

Researchers have reported experimental results showing successful (simulated) transmission at 2.4 gigabits per second over a distance of 21,000 kilometers and higher speeds over shorted distances.

3.1.2.11 Fibre Cables

As we have seen, fibres themselves are generally 125 µm in diameter (very small indeed). Cables carrying fibres vary widely in their characteristics. One typical indoor/outdoor cable is shown in crossection below:

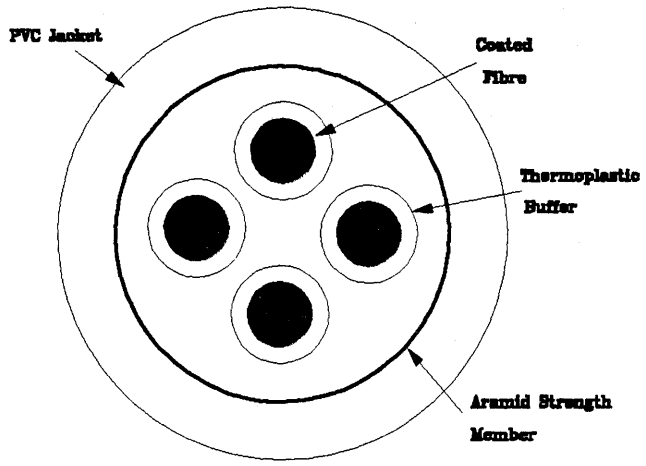


Figure 21. Typical Fibre Cable

Fibre cables are made to suit the application they are to perform and there are hundreds, perhaps thousands of types. The kinds of variations you see between cables are as follows:

- Number of fibres in a single cable. This typically ranges from two to around 100. Outdoor telephone company single mode fibre cables tend to have about 30 fibres in the single cable and since this is quite thin and light they will install multiple cables on routes where 30 fibres isn't enough.
- Strength members. Many cables, particularly for outdoor applications have steel wire at the core to provide strength.
- The optical characteristics of the fibre itself. Single mode or multimode, loss in dB per km, susceptability to temperature changes, etc.
- Electrical wire. In some cable situations it is necessary to power repeaters over long distances. One example of this is in submarine cables. Electrical power cabling is often included to deliver power to the repeaters.

3.1.2.12 Transmission System Characteristics.

The characteristics of various transmission systems are summarised in Figure 22 on page 57.

A universally accepted measure of the capability of a transmission technology is the product of the maximum distance between repeaters and the speed of transmission. In electrical systems (such as on the subscriber loop) maximum achievable speed multiplied by the maximum distance at this speed yields a good rule of thumb constant. It is not quite so constant in optical systems, but nevertheless the speed x distance product is a useful guide to the capability of a technology.

medium	source			speed.distance	product
copper				2Mbps 2km	4 M
multimode	LED	802.5	in use	32Mbps 2km	64 M
mono mode	LED	FDDI	in use	125Mbps 2km	250 M
mono mode	laser	long dist	in use	1.7Gbps 50km	85 G
mono mode	laser	amplitude mod.	in lab	8Gbps 100km	800 G
mono mode	laser	coherent	in lab	400Mbps 370km	150 G
mono mode	laser	solitons	in lab	2Gbps 4000km	8 T

Figure 22. Optical Fibre State of the Art

Figure 23 shows the attenuation characteristics of various transmission media and the maximum spacing of repeaters available on that medium. Of course this is very conceptual. Special coaxial cable systems exist with repeater spacings of 12 kilometers. In section 2.3.5, "Digital Transmission State of the Art" on page 39, we discussed systems capable of operatin at very high speed over telephone twisted pairs for distances of four to six kilometers without repeaters. Nevertheless the advantage of fibre transmission is obvious.

	attenuation	repeater spacing max 35db
Coaxial Cable	25 db/km	1.5 km
Telephone Twstd Pair	12-18 db/km	2-3 km
Window Glass	5 db/m	7 m
Silica - Installed	0.3-3 db/km	10-100 km
Silica - Development	0.16 db/km	250 km
Halide - Research	0.01 db/km	3500 km

Figure 23. Signal Loss in Various Materials

3.1.3 Fibre Optics in Different Environments

Optical systems are built to be optimal for the particular environment in which that system is to be deployed. The point here is that the local data communications environment and the wide area telecommunications environment are very different in their character and requirements. Hence we might expect that the systems built for each environment will themselves be different.

Wide Area Telecommunications Systems

In the wide area environment carrying very high capacity over long distances is the primary requirement. There are relatively few sources and receivers and few connectors. There is very little need to change a system once it is installed.

- Cable cost is very important (though the cost of burying it in the ground is many times higher than that of the cable itself).
- The cost of transmitters, receivers, repeaters, connectors etc. is much less important because that cost is only a tiny proportion of the total.
- High power is very important to achieve maximum distance between repeaters.
- Component reliability is also very important because these systems are typically multiplexed and a single failure affects many users.

Local Area Data Communications Systems

The most important thing about this kind of system is its need for flexibility.

- The cost of transmitters and receivers etc. is most critical (because there are a large number of these and they form a large proportion of the total).
- Cable cost is still important but much less so than in the wide area environment. Compared to electrical cables, fibre cables are much easier to install around a building (they are lighter and more flexible).
- The critical thing in this environment is joining the cable and the performance of connectors and patch cables.
- High power is important so that losses incurred in connectors and patch cables can be accomodated.
- Reliability is also very important because a single failure can disrupt the entire system.

So in both types of application it is important to have high power and reliability. These requirements lead to different system choices:

- For Wide Area Telecommunications, single mode fibre and long wavelength lasers constitute the system parameters of choice.
- For Local Data Communications, the choice is for shortwave lasers and multimode fibres.

3.1.4 Future Developments

3.1.4.1 Wavelength Division Multiplexing (WDM)

As was seen earlier, when a signal from a Laser is sent on an optical fibre only a single wavelength is used. The band of frequencies (wavelengths) occupied is very narrow. But the fibre itself can transport light over a very wide range of frequencies (expecially the new VAD fibres). Why not use the principles of frequency division multiplexing (described in Appendix A.1.1, "Frequency Division Multiplexing" on page 275) to provide much greater throughput on the fibre?

The potential is enormous. Perhaps 10,000 simultaneous signals could be easily handled by one, single mode fibre thus multiplying the throughput by 10,000.

Many researchers are working on exactly this idea and experimental systems have been demonstrated. But there are problems:

1. Light generated by many different sources (lasers) must be combined (with very little loss) and fed into the end of a single mode fibre. This is difficult to do.
2. At the other end of the fibre, the light must be separated into its component signals (again without losing too much signal strength). This is not as difficult. A prism or diffraction grating can separate wavelengths very precisely. Detectors must be then positioned and integrated so that each band of light coming from the prism can reach its proper detector.
3. Light at different frequencies is attenuated at different rates by the fibre. This doesn't bother an individual channel. But if some channels decay faster than others then all of the signal will have to be boosted (amplified or repeated) whenever the most attenuated signal in the group requires it. This means that repeaters etc. need to be closer together than they would otherwise be.
4. If repeaters are used, then the signal must be demultiplexed into its separate components, separately repeated and recombined at every repeater stage.

The above factors have combined to make FDM systems unattractive on the basis of cost. However, the advent of practical light amplifiers (these amplify all of the signal regardless of its wavelength) means that you don't have to demultiplex at every repeater stage.

In the next few years it is expected that WDM systems will see the beginning of commercial use.

3.1.4.2 Frequency Modulation

The current techniques of signaling with short bursts of light bear a startling similarity to the morse code transmitter and the spark coil!

If we can carry the information as changes in the frequency of light rather than in rough pulses there are two effects:

1. The signal can be detected at a much lower signal level than we can for pulses. This is because an FM receiver "locks on" to the received signal and does not need to detect the beginning and end of pulses.
2. The signal can be much higher data rate.

FM systems of modulating light have been demonstrated as long ago as 1984. There are problems in building a suitable laser and a suitable detector but these look like they will be solved in the relatively near future. The next generation of optical systems will probably use FM transmission.

3.1.4.3 Better Fibre Manufacture

Figure 18 on page 48 shows the infrared absorbtion characteristics of two typical commercial fibres. The lower (better) curve shows a very pronounced absorption peak at about 1450 nm because of the presence of hydroxyl ions (water). This curve relates to fibre produced by a process called "Modified Chemical Vapor Deposition" (MCVD).

There is a much better manufacturing process available called "Vapor-phase Axial Deposition" (VAD). This process is however, difficult to implement in commercial manufacture. However it produces a fibre with a significantly lower absorbtion than MCVD and with little or no absorption peak due to hydroxyl ions.

Over time as manufacturing processes improve it is expected that VAD fibres will become lower in cost and will be universally used.

3.1.4.4 Optical Logic

There is very strong current research activity in the area of providing optical computers. That is, computers where the logic is completely optical and the signal is carried optically rather than electrically. Much progress is being made but it is generally believed that the fruit of this research is many years off yet.

Chapter 4. Traffic Characteristics

4.1.1 User Requirements

Many organisations see the new lower communication cost structure as an opportunity for:

1. Doing old applications better.
2. Doing new applications that were not feasable (or indeed imaginable) before.

The first requirement is to integrate existing networks into a single network. The motivation for this is not only to save money on links but to provide a better networking service by integrating the many disparate networks into a single coherently managed unit.

The kinds of existing networks that users want to integrate can be summarised as follows:

- Traditional data networks
- Voice networks
- Interconnected LAN networks
- Multiprotocol networks

In addition there are opportunities for applications using:

- Image
- Full motion video

Traditional data networks were built to handle both interactive and batch data but were not built to handle image, voice or video traffic. The new types of traffic put a completely new set of requirements onto the network.

4.1.2 The Conflicting Characteristics of Voice and Data

It is attractive to think that when voice is digitised, it then becomes in some way "the same" as data. Or it "becomes" data. Up to a point this is true, but there are many differences between traditional data traffic and digitised voice which make the integration of the two a challenging technical problem.

Length of Connection (Call)

Traditionally, the most important difference between voice and data has been that voice calls are (on average) around three minutes and data calls can last for many hours. Telephone exchanges have been designed to have large numbers of external lines but relatively few "paths" through the exchange for calls. So it is possible, when the exchange is busy, for calls to "block". That is, for the caller to be attempting to make a connection and the called interface to be unused but the call cannot be made because all paths are "blocked" (in use by other calls).

Therefore, when data is passed through a traditional telephone exchange, all the paths can be used up very quickly and the rest of the exchange will become unavailable because all paths are "blocked". Modern digital PBXs have solved this problem by providing the capacity to handle more calls than there are interfaces. For example, an

exchange with 500 telephones can have a maximum of 250 simultaneous calls...but an internal capacity for perhaps 400 calls. This is because the internal data path equipment in a digital PBX represents only two or three percent of the total cost of the exchange; whereas, in the past, the function accounted for perhaps 30% of the cost. Nevertheless, while this difference is solved for the time being, in the future, the problem will appear again as the sharing of voice and data on the tails becomes more common. Hence, the number of connections to be made increases and the internal limitations imposed by bus speed become a factor.

Flow Control

The bandwidth required for voice is dictated by the digitisation technique and the circuit is either in use (using the full bandwidth) or not. Data can go at any speed up to the access line speed. Voice does not need (or want) flow control. (Voice must be either handled at full speed or stopped. You cannot slow it down or speed it up.) Data, on the other hand, must be controlled since a computer has an almost infinite capacity for generating data traffic.

Data has another problem, in that a data device, such as a terminal, can and will, establish a connection and use it in a very "bursty" manner. There may be minutes or hours with no traffic at all and then a few minutes of data at the maximum transmission rate. Added together statistically the traffic does NOT "average out". What happens in large data networks is that interactive traffic tends to "peak" at different times of the day, and on particular events (for example, after a computer system has been "down" and has just recovered).[23]

Voice does exist in bursts (talk spurts) also and in general only one party speaks at one time, but statistically it poses quite a different problem for a switching system than does data.

In the past, there have been several levels of flow control available to data devices. For example, link level controls (which are aimed at optimising the use of the link and not at network flow control), and network delivery type controls (like pacing in SNA or packet level flow control in X.25).

Delivery Rate Control

In data networking equipment in the past, there has also been another very important control, that of the link speed itself. Most equipment is designed to handle data at whatever speed the link can deliver it (at least at the link connection level). At the "box" (communication controller, packet switch) level, the switch is never designed for every link to operate simultaneously "flat out", but any individual link attachment must have that capability. Link speed provided an implicit control of the rate at which data could be sent or received.

---

[23] Another peculiarity here is that the difference between the peaks and the troughs in data traffic becomes greater as the network gets larger. This is not due to network size per se but rather is an effect that follows from the same cause. Networks get larger because terminal costs decrease. As the cost of terminals and attachments decreases, users are able to afford many for dedicated functions. An example is in the development of banking networks. In the early networks there was only one (expensive) terminal per branch and work was "queued" for it. It was in use all of the time with a dedicated operator (with others taking over during lunch). Thus there was very little variance in the traffic over time (though the mixture of transaction types changed quite radically). Now, with cheaper terminals, most bank branches have many terminals and they are operated by their direct users not by dedicated operators. Thus, in midmorning for example, after the mail arrives, there is a processing peak with every terminal in use. At other times there can be little or no traffic.

But the new technology allows link speeds which are very much faster than the attaching "box". For example, a link connected to a terminal (personal computer) might run at 64 Kbps but the device, while handling instantaneous transmission or reception of blocks at this speed may not allow for aggregate data rates much faster than (say) 500 characters per second. The same device might also be connected to a local area network at four Mbps with the same restriction that only a few hundred characters per second can be handled by the device. The same characteristic at higher speeds applies to the data switching processor itself.

This leads to the observation that if link speed is no longer to be a flow limiting mechanism, then others (adequate ones, such as those shown in section 8.3.5, "Flow and Rate Control" on page 163, exist) will have to be used.

Blocking Characteristics

Data exists in discrete blocks.[24] It is transmitted through the network in blocks. The two block sizes can be different (logical blocks can either be split up or amalgamated for transport). Telephone traffic is continuous or effectively so. It can be considered as very long indeterminate length blocks but the "real time" characteristic does not allow the network to receive a burst of speech as a single block and treat it that way.

Acceptable Transit Delay Characteristics

An acceptable network delay for even the most exacting real time data network is about 200 milliseconds (one way). More usual is a data interactive traffic delay of 500 milliseconds or so. Batch data does not have a problem with transit delays. Voice traffic, however, is marginal on a satellite where the transit delay is 250 milliseconds one way. For first quality voice, the transit delay should be no greater than 50 milliseconds.

Further, variable transit delays (variations in response time), while an annoyance in data traffic, make voice traffic impossible. Voice packets must be delivered to the receiver at a steady, uniform rate. They must not "bunch up" and get delivered in bursts (a characteristic of today's data networks).

A short interruption to the circuit (for example, caused by an aeroplane flying between two microwave repeaters) which could result in a one-second outage of the link will have quite different effects for voice than for data. For data, it is nearly always preferable to have a delay of a few seconds rather than losing the data. With voice, a packet that is one half a second old is just garbage. It is much better to discard delayed voice packets quickly, thus allowing the circuit to return to normal, than it is to build up a queue, particularly due to the fixed speed of the receiving (and of the transmitting) device.

Error Control

The most important thing about data traffic is that errors must be controlled, either detected, or (preferably) detected and corrected. This

---

[24] There is an unfortunate conflict here in the usage of the word "block". In the telephone world it describes the action of preventing a call being set up due to lack of resources. In the data world a "block" is a logical piece of data which is kept together for transport through the network.

correction mechanism can often only be done by context[25] (since you don't know who the sender is until you are sure there are no errors in the block), and will require retransmissions for recovery. Voice, on the other hand, cannot tolerate the time delays inherent in recoveries and does not care about occasional errors or bursts of errors. (Voice and human language are very redundant indeed.)

Power Demands

Problems caused by fluctuations in the demand for power, should not happen in modern digital systems.

Statistics shows us that when many variable (or varying) things are added up then the mean (average) becomes more and more stable (has less and less variation). For example, in voice calls if one takes the power demands on a trunk amplifier for a large number of calls, then the requirement is very stable indeed and well known. The dynamics of speech, when added up over many calls, produces remarkably stable system demands.

When data is used instead of voice, many things change. Call duration is usually cited (data calls are generally much longer than voice) but there are other problems. When modems are used for data communication over a telephone channel there are no "gaps between words". The modem produces a constant, high level signal. If too many modem calls happen to be multiplexed on a single interexchange (frequency division) trunk, then the additional electrical power, required by the multiplexors and amplifiers can be so great as to cause the device to fail. (Power supplies are designed to supply only enough power for voice calls.) This restriction will go away with the advent of digital systems, but it was the cause of PTT hesitancy about allowing modems to be connected arbitrarily around the telephone system without consideration of their effects on that system.

Volume of Data

If telephone calls are to be regarded as 64 Kbps full-duplex, then not even the largest organisation today (1991) transmits enough data to be more than ten percent of its telephone traffic. Most organisations transmit less than five percent, and of all the communications traffic carried over public communication lines perhaps one or two per cent is data. This is very important since anything that is done in the system to accommodate data traffic, if it adds cost to the voice part of the system, will be very hard to justify because of the large cost added to the total system for small benefit.

It is perfectly true that data traffic is growing rapidly and voice traffic is not, but there is a very long way to go. Particularly in that the number of interfaces to the public networks being used for voice versus the number of interfaces being used for data is a more important criteria than the number of bits sent. This ratio of the number of interfaces is even more biased in the direction of voice traffic.

---

[25] At the link level, the sender is always known regardless of the content of the block. Later when released from the context of the link, the only identification for the block is the routing information within the block itself.

Balanced Traffic

Most voice calls involve a two way conversation (albeit that some people talk more than others!). This means that for voice transmission, the traffic is usually reasonably well balanced.

Not so for data. Even without the obvious example of file transfer (which is one way), traditional (IBM 3270 style) interactive data traffic involves very short (perhaps 30 to 50 bytes) input and large output (typically 500 bytes but often 2000 bytes or more). In graphics applications the unbalance is even greater than this.

Echo Cancellation

In traditional (analogue) voice systems, the problem of suppressing echoes is extremely important. In a digital full-duplex system, it would seem that echos were no longer a consideration.

This is not completely true. Some echoes can be generated within a telephone handset and though this is a small problem compared with the problems of the past, it must still be considered. In a system where voice is packetised, the size of the packet determines the length of time that it takes to fill a packet before transmission (64 Kbps equals one byte per 125 µsec). As the delay in the circuit increases, then so does the problem caused by echoes.

These facts have fueled a debate over the optimal packet size for packetised voice. Some people contend that a packet of around 80 bytes or so will produce problems with echoes where packet sizes of 32 bytes will not. (This is because of the time needed to assemble a packet.)

There is a significant problem with echoes in the situation of a digital full-duplex backbone network with analogue subscriber loops. As noted in section 2.3.3, "The Subscriber Loop" on page 35 these loops can generate large echoes and this will be a problem if the network delay exceeds about 40 milliseconds.

4.1.3 Characteristics of Image Traffic

Image traffic is conceptually similar to traditional data traffic with one major difference - images are very large compared to traditional character screen images.

A traditional IBM 3270 character screen showing multiple fields and many colours averages about 2,500 bytes (the screen size is 1,920 bytes but other information relating to formatting and field characteristics is present. The same screen displayed as an image could be as much as 300KB. In this document, figure 123 on page 284 takes 126KB of storage. A character graphics diagram such as figure 24 on page 67 takes about 400 bytes of storage.

Images are therefore transmitted as groups of frames or packets (in SNA, as "chains"). Response time is important but only within normal human response requirements. Less than a second is goodness, up to perhaps five seconds is tolerable, above five seconds and users become more and more seriously inconvenienced.

Nevertheless, because image traffic is typically initiated by a human operator entering some form of a transaction, display will be relatively infrequent - because systems are such that a user typically needs to spend time looking at the display before looking at the next image. In the future this may not hold true.

Online books (with illustrations) for example may encourage users to "flick through" the pages looking for the information they need. "Flicking through the pages" of a book involves the consecutive display of many images perhaps a second or two apart. This could put a substantial unplanned load onto a data network.

4.1.4 Characteristics of Digital Video

At first thought video traffic appears to share many of the characteristic of voice traffic (you set up a connection, and transmit a continuous stream of data at a more or less constant rate until you no longer need the connection). In reality, while there are many similarities, transporting video in a packet network is a quite different problem from transporting voice.

Video systems display information as a sequence of still pictures called frames. Each frame consists of a number of lines of information. The two predominant broadcast television systems currently use 625 lines at 25 frames/sec (PAL) or 450 lines at 30 frames/sec (NTSC).

Data Rate

If a PAL signal is to be digitally transmitted we could perhaps break up a line into 500 points and encode each point in 12 bits (colour and intensity etc.). This becomes quite a high transmission rate:

625 (lines) × 25 (per sec) × 500 (points) × 12 (bits) = 93,750,000 bits/sec

In fact, for reasonable resolution we probably don't need 500 points in each line and maybe we can code each point as 8 bits, but whichever way you look at it the data rate is very high.

But this is altogether the wrong way to look at video. Over history we have broadcast video (a PAL signal requires about seven megacycles of bandwidth) over a fixed rate channel. Every point in the picture was sent (albeit via analogue transmission) in every frame. **But the information content of a video frame is inherently variable.** The point about video is that the majority of frames are very little different from the frame before. If a still picture is transmitted through a video system, all we need to transmit is the first frame and then the information content of each subsequent frame is *one* bit. This bit says that this frame is the same as the one before!

If a video picture is taken of a scene such as a room then only a data rate of one bit per frame is necessary to maintain the picture (that is, 25 bits/sec for PAL). As soon as a person enters and walks across the room then there is much more information required in the transmission. But even then much of the picture area will remain unaffected. If the camera is "panned" across the room then each frame is different from the one before *but* all that has happened is that the picture has moved. Most pixels (picture elements - bit positions) move by the same amount and perhaps we don't need to retransmit the whole thing.

There are many examples. The typical head and shoulders picture of a person speaking where most of the picture is still and only the lips are moving. But in a picture of a waterfall many pixels will be different from ones before *and* different in non-systematic ways. A video picture of a waterfall has a very high information content because it contains many non-systematic changes.

What is being discussed here is something a little different from what we traditionally regard as compression. When a still picture is examined, much of the picture area contains repetition. Any particular line (or point) will very likely have only small differences from the one either side of it. Within a line there will be many instances of repetition such as when crossing an area of uniform colour and texture. There are many algorithms available to compress a single image to a much smaller amount. So, although one can look for redundancy and compress it, a still picture contains a fixed amount of information (from an information theory viewpoint). A sequence of video pictures is different in the sense that from an information theory standpoint, each frame can contain from one to perhaps a few million bits!

The net result of the above is the conclusion that video is fundamentally variable in the required rate of information transfer. It suggests that a variable rate channel (such as a packet network) may be a better medium than a fixed rate TDM channel for video traffic. Consider the figure below:

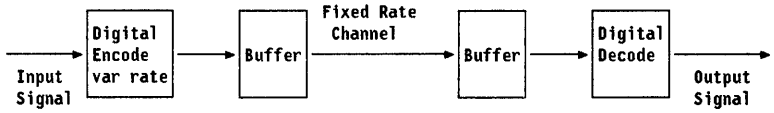


Figure 24. Transmitting Video over a Fixed Rate Channel

This is typical of existing systems that transmit video over a limited digital transmission channel. Systems exist where quite good quality is achieved over a 768 Kbps digital channel. When the signal is digitally encoded and compressed, the output is a variable rate. But we need to send it down a fixed capacity channel. Sometimes (most of the time) the required data rate is much lower than the 768 Kbps provided. At other times the required data rate is much higher than the rate of the channel. To even this out a buffer is placed before the transmitter so that if/when the decoder produces too much data for the channel it will not be lost. But when the data arrives at the receiver end of the channel data may not arrive in time for the next frame, if that frame contained too much data for the channel. To solve this, a buffer is inserted in the system and a delay introduced so there will be time for irregularities in reception rate to be smoothed out before presentation to the fixed rate screen.

Buffers however, are not infinite and if the demands of the scene are for a high data rate over an extended period of time then data is lost when the buffers are filled up (overrun). This is seen in "full motion" video conference systems which typically operate over a limited channel. If the camera is "panned" too quickly then the movement appears jerky and erratic to the viewer (caused by the loss of data as buffers are overrun).

It is easy to see from the above example that it is quite difficult to fit video into a limited rate channel. Always remembering that the average rate required in the example above will be perhaps ten times less than the 768 Kbps provided and that most of the channel capacity is wasted anyway!

The extreme variation in information transfer requirement means that if a fixed rate channel able to handle the fastest rate is used then there will be a large amount of wasted capacity. If a limited channel is used then there is less (but still significant) waste of capacity but more important, there is loss of quality when a high transfer rate is used for a longer time than the buffers can hold. (Typically, existing systems buffer for a few seconds of high activity in the picture

- if something such as the stereotype television car chase sequence occurs then the system can't handle it.)

Thinking about the matter statistically, if a number of video signals were able to share the same communications resource then it is likely that when one video channel requires a high bandwidth, others will require much less. The statistics of it say that the more signals there are sharing a resource the less variation there will be in the resource requirement.

> When there are only two users sharing, there is a reasonably high probability that there will be times when both signals will require a high transfer rate at the same time. When 50 signals share the resource there is still a finite probability that all 50 will require a high transfer rate at the same time, but that probability is tiny. This is discussed further in Appendix B, "Queueing Theory" on page 283.

This all leads to the conclusion that high speed packet networks and LANs are the natural medium for video transmission.

Timing Considerations

Video traffic is like voice in one important respect - it is isochronous. Frames (or lines) are delivered to the network at a constant rate and when displayed at the other end must be displayed at the same rate. But packet networks tend to deliver data at an uneven rate (this is sometimes called "packet jitter"). Something needs to be done at the receiver end to even out the flow of packets to a constant rate. As with voice, this can be done by inserting a planned delay factor (just a queue of packets) at the receiver.

Redundancy

Even more than voice, video is very redundant indeed. The loss or corruption of a few bits is undetectable. The loss of a few lines is not too much of a problem since if we display the line from the previous frame unchanged, most times the loss will be undetected. Even the loss of a frame or two here and there doesn't matter much because our eyes will barely notice. Of course it must be noted that when video is digitally coded and compressed loss or corruption of packets will have a much larger effect (because the data is now a lot less redundant).

Video Applications

Very often video applications are for one way transmission (as in viewing television or a movie). In this case the amount of delay that we may insert into the system without detriment can be quite great (perhaps ten seconds or more).

Interactive video is a little different in that this is the "videophone" application. People talking to one another accompanied by a picture. In this case although the voice communication is logically half-duplex (that is, hopefully only one person talks at one time), the video portion is continuous. Delay is still less stringent than for voice - although the voice component has all the characteristics of regular voice (without video). It appears that synchronisation of voice with the movement of lips is not too critical. Most people do not detect a difference of 120 milliseconds between the image and the sound in this situation.

Digital Video in a Packet Network

The discussion above concluded that packet networks are a natural medium for video transmission. But certainly we don't mean "traditional" packet networks. Many, if not most, existing packet networks don't have sufficient total capacity to handle even one video signal! In order to operate properly, a packet network processing video must have a number of important characteristics:

1. Sufficient capacity. The throughput capacity of the network must be sufficient to handle several video signals together - otherwise the benefit of sharing the resource is lost.
2. End-to-end delay appropriate to the application. This varies quite a bit with the application. One way traffic doesn't care about network delay too much. Interactive video needs a transit delay approximating that of voice (because voice accompanies it) but does not need to be exactly synchronised to the voice.
3. Minimal packet jitter. Irregularities in the rate of delivery of packets need to be smoothed out by inserting a buffer and a delay.

In addition there is the question of what to do when the network becomes congested and how to handle errors.

Hierarchical Source Coding

All networks of finite capacity encounter congestion at various times. But with video (as with voice) you can't slow down the input rate to the network in order to control congestion (as we do in data networks) because a video frame arriving too late is simply garbage. If the network is congested the best we can do is to throw some packets away until the network returns to normal. If this happens only very infrequently, then video and voice users will not get too upset but if it happens often then the system can become unuseable.

One approach to congestion is to code the information (video or voice) into packets in such a way that the information is split up. Information essential to display of the frame is coded into a separate packet from information that merely improves the quality. This means that some packets contain essential information and others less essential information. The packets can be marked in the header so that the network will discard only non-essential packets during periods of congestion. This technique (originally invented for handling packet voice) is called "Hierarchical Source Coding" (HSC) and has the obvious advantage of allowing the system to continue basic operation during periods of congestion.

The concept is very simple. Imagine that a particular byte of encoded data represents the intensity level of a particular point on the screen. A simple HSC technique might be to take the high order four bits and send them in one packet (marked essential) and the low order four bits in a different packet (market non-essential). In the normal case when the packets arrive at the destination the byte is reconstructed. In the case of congestion, perhaps the packet containing the less important low order bits has been discarded. The receiver would then assume the low order four bits have been lost an treat them as zeros. The result would be to give 16 levels of intensity for the particular point rather than the 256 levels that would have been available had the less important packet not been discarded. In practice, HSC techniques need to be designed in conjunction with the encoding (and compression) methods. These can be very complex indeed.

In principle, this is not too different from what we do in the analogue broadcasting environment.

Most colour TV sets contain a circuit called a "colour killer". When the received analogue TV signal is too weak or contains too much interference the circuit "kills" the colour and displays the picture in black and white. This enables a viewer to see a picture (albeit a B+W one), which, if displayed in colour, would not be recognisable.

In radio broadcasting of FM stereo an ingenious system is used such that two signals (left channel plus right channel and left channel minus right channel) are transmitted. The two are frequency multiplexed such that the L+R signal occupies the lower part of the frequency band and the L-R the upper part. When the signal is received strongly, the channel separation can be reconstructed by addition and subtraction of the channels. When the signal is weak, the L+R signal dominates because it occupies the lower part of the band. What you get then is only L+R (mono) reception. So when the signal is weak, you lose the stereo effect but still get a basic signal.

Hierarchical Source Coding will probably become a basic technique for processing both voice and video in packet networks.

Error Control

The worst problem in processing video is packet jitter (erratic delays in packet delivery). Recovery from link errors by retransmission of data is not useable within a packet network containing video for this reason. The best thing to do with errored packets is to discard them immediately. Mis-routing due to errors in the destination field in the header can have catastrophic effects. Packets should have a frame check sequence field which should be checked every time the packet travels over a link and the packet discarded if an error is found.

There is a question about what to do at the receiver when an expected packet doesn't arrive due to errors or congestion in the network. It has been suggested that using a very short packet (or cell) size with an error correcting code might be a useful technique. Unfortunately, while this technique would overcome random single bit errors etc. it is not a satisfactory way to overcome the loss of a many packets in a group. This is because an error correcting code capable of recovering from this kind of situation would be so large that the overhead would be unacceptable.

The best technique for handling errors in video involves using the information from the previous frame and whatever has been received of the current frame to build an approximation of the lost information. A suitable strategy might be to just continue displaying the corresponding line from the previous frame, or if only a single line is lost, extrapolating the information from the lines on either side of the lost one.

High Quality Sound

High quality sound (CD quality stereo) involves a very high bit rate. Regular CDs use a bit rate of 4 megabits per second. Encoding sound is, in principle, the same problem as for voice but with a few differences for the network:

- High quality sound (such as a film soundtrack) is continuous - unlike voice transmission where talk exists in "spurts".

- The data rate is much higher (but the same compression techniques that worked for voice also work here).
- Delay through the network doesn't matter as much - this depends on the requirements for the video signal the sound accompanies.
- The major requirement is that (like video and voice) high quality sound be delivered to the network at a constant rate and played out at the receiver at a constant rate.

Chapter 5. Principles of High Speed Networks

If the user requirements outlined in the previous chapter (integration of data, voice, video, image..) are to be satisfied by packet networks then clearly a new type of packet network will be needed. Network nodes will need to handle the full data throughput capacity of the new high speed links (one million packets per second - plus) and network architectures will need to accommodate the unique characteristics of voice and video traffic.

The requirements may be summarised as follows:

Very High Node Throughput

Nodes must be able to route (switch) data at the peak combined rate of all links connected to them. In corporate networks this might mean a maximum of perhaps 20 links at 155 megabits per second, but this seems a little high for the decade of the 1990s. More likely would be a switch with less than 20 links where perhaps four of them are 155 megabits per second and the rest might be at the "T3" rate of 45 megabits per second.

But corporate private networks are one thing. Public telecommunications networks are something else. The proposal with ATM (B_ISDN) is that packet (cell) switching should become the basis of a multi-function network, which will replace the world,s telephone network. To do this, a mainline trunk exchange (probably a cluster of switching nodes) would need to handle perhaps 100 links of 620 megabits per second today and perhaps the same 100 links would be running at 2.4 gigabits per second by the time the system was built.

Using 53-byte cells, a 2.4 Gbps link can carry just less than six million cells per second *in each direction.*

The example is a little extreme in 1992 but the principle is clear. We are going to need the ability to process cells at rates of well above one hundred million per second for Broadband ISDN to become a reality.

Minimal Network Transit Time

This is a critical requirement for voice and is discussed later in section 5.2.2.2, "The Effect of End-to-End Network Delay on Voice Traffic" on page 81.

Minimal Variation in Network Transit Time

When any traffic with a constant bit rate at origin and destination travels through a network, the variations in network delay mean that a buffer somewhat larger than the largest foreseeable variation in transit time is needed. This buffer introduces a delay and for practical purposes can be considered a net addition to network transit time. This is further discussed in section 5.2.2.2, "The Effect of End-to-End Network Delay on Voice Traffic" on page 81.

To meet the above requirements networks will need to have the following characteristics:

Totally Hardware Controlled Switching

There is no way that current software-based packet switched architectures can come to even one one hundredth of the required throughput - even assuming much faster processors.

However, there are several hardware switching designs that will meet the required speeds at (predicted) reasonable cost.

Suitable Network Architecture

The network architecture must make it possible for the data switching component in a node to decide the destination to which an incoming packet should be routed *at full operating speed*.

The network architecture must provide mechanisms for the stable operation and management of the network but the data switching element must not need to get involved in extraneous protocols.

Link Error Recovery

Recovery from transient link errors by retransmission (for voice traffic), as is usual for data traffic, can seriously conflict with the requirement for uniform delivery rates. For voice, a delayed packet is worse than a packet in error. However, by the nature of packetisation, it is necessary that packets contain a header which carries routing information (identification) so the destination switch can route it to the appropriate destination. An error in this information will cause a packet to be routed to the wrong destination AND a packet to be lost from the correct circuit.

But these very high speed networks are planned to operate solely over digital (preferably fibre optical) circuits. Error rates on these circuits are around ten thousand times better than they were for traditional analogue data links.

For the data portion of the packet or cell, error checking and recovery can be applied on an end-to-end basis especially if the error rates experienced on links is very low.

The header portion is not so fortunate. An error in the header can cause a packet to be misrouted to the wrong destination. The network must at least check the headers. (In ATM there is an elegant mechanism that checks the Header Error Check field to obtain cell synchronisation. This is described in section 7.1.4, "Physical Transport" on page 134.)

Packet Length

Short (less than 64 byte), fixed length packets or cells are an attractive option because:

1. Their fixed length nature gives a uniform transmission time (per cell) characteristic to the queueing within a node for an outbound link. This leads to a more uniform transit time characteristic for the whole network.
2. The shorter the cell the shorter the time needed to assemble it and hence the shorter the delay characteristic for voice.
3. Short, fixed length cells are easy to transfer over a fixed width processor bus, and buffering in link queues is a lot easier and requires less processor logic.

One elegant solution to both the network delay and error recovery problems would be to use very short packets (perhaps 32 bytes) of fixed length. If this is done then Error Correcting Codes (ECC) can be used as a recovery from transient link errors. Two bytes of ECC are required for every eight bytes of data. A 32-byte packet would then have a routing header (2 or 4 bytes) included within it and one or four ECC two-byte groups appended to it. (One if it is thought necessary only to check the

header, two if the data is to be error recovered also.) Therefore, a packet would be either 34 or 40 bytes. (An overhead on the transmission channel in the full ECC case of 20%.) It happens that the use of ECC in this way for a voice packet is considered wasteful and unnecessary. The loss of a packet or two (provided it is relatively infrequent) or the corruption of a few bits of data is not considered to be significant.

The international standard for cell size is now 48 bytes (for ATM). In ATM the header is checked for validity but the data within the cell is not (or, rather, that checking and error recovery on the data within a frame (group of cells) is left to the end-to-end protocol called the "adaptation layer").

However, there is another side. Video transmission is fine with packet sizes of over a thousand bytes. Data transmission can be achieved with low overhead if the packet size adopted is large enough to carry the largest natural data block as produced by the user's application.

The longer the packet the fewer packets per second must be switched for a given data throughput.

This subject is discussed in more detail in section 5.5, "Transporting Data in Packets or Cells" on page 86.

Flow Control

Control of congestion is a critical matter in any packet switching environment. Traditional techniques of flow control are not possible at very high packet rates because they require significant amounts of programmed logic to operate on every packet.

In a high speed switch, input rate regulation and capacity reservation are the appropriate techniques. These can be agreed by the control processors when a connection is started and enforced at the entry points of the network.

This subject is further discussed in section 5.1, "Control of Congestion" on page 77.

Congestion Control

Congestion occurs when a node builds up too much data for its internal buffers to process. This can happen even in data networks with very detailed explicit flow controls.

One way to handle congestion is to avoid it. Good flow controls can help in avoiding congestion. Another sure way of handling congestion is to make sure that the maximum demand that can ever be placed on the network can be met at all times. This means running links and nodes at average utilisations of around ten or twenty percent at the peak! But this forgoes the benefits of sharing the network.

If the network is to process variable rate data (say voice) from many thousands of users simultaneously and if no single user can make a peak demand sufficient to be noticed, then the statistics of the situation work for us. As described in Appendix B, "Queueing Theory" on page 283 as you add up many variable sources (that are unrelated to one another) the total becomes very stable.

Congestion becomes a problem where there are a number of sources that can individually place a significant demand on the network (such as in variable rate video). In this case a small number of users (as few as

10 perhaps) might be able to each make peak demands simultaneously and bring the whole network to a standstill. The trick here is to avoid the situation where any single user can make a significant demand on the network.

But some types of traffic change radically over time. Data traffic peaks at different times in a business day. Batch data peaks during the night.

When congestion occurs packets must be discarded. For some data types (voice, video) coding can be such that low priority packets can be discarded with the net effect of a "graceful degradation" of the service. If these packets are marked as discardable in some way (this is a feature of both ATM and Paris), then the system can alleviate congestion by discarding these.

If congestion becomes very serious, then the network will need to discard packets not marked as discardable. The network should have a way of prioritising traffic by service class so that an intelligent packet discard strategy may be adopted.

This packet discard strategy must be performed by the (hardware) data switching element. The discard strategy must be very simple.

Sequential Delivery

If packets applying to one conversation are allowed to take different routes through the network (for load balancing for example) then they must be resequenced into order before delivery to the receiver. However, this means that each would have to carry a sequence number (more overhead) and the technique would result in "bursty" uneven delivery. To overcome this, delivery would then need to be buffered sufficiently to even out the bursts. This would add cost but more importantly it would add to the transit delay and thus degrade the quality.

In a high speed network this means that each connection must be limited to a fixed path through the network.

Priorities

There is no consensus yet on whether transmission priorities are relevant in a high speed network. A transmission priority may be given to a packet and that priority enables it to "jump the queue" ahead of lower priority packets when being queued for transmission within a node.

Within a tightly controlled traditional packet networking system such as SNA, the system of priorities has worked well. It gives better response time to higher priority traffic and also enables the use of much higher resource (link and node) loadings than would be possible without them.

But at such high speed, with relatively small cells (at the speeds we are considering even a 4KB block is small - in time), many people suggest that the cost of implementing priorities may be greater than it is worth. Most studies of high speed node technology suggest that the total switching (processing, queueing and transmission) in the kind of node under discussion will be well less than one millisecond.

Other kinds of priority are, however, considered essential. In a large network there needs to be some control and prioritisation of the selection of routes through a network depending on the required service characteristics for a particular class of service.

In addition it seems generally agreed that a service class type of priority should be used to decide which packets to discard at times of network congestion.

End-to-End Protocols and "Adaptation"

The characteristics of a high speed network developed thus far are such that it gives very high throughput of very short packets, but in th case of congestion or of link errors packets are discarded.

To provide a stable service, the network needs to have processing at the entry and exit points of the network. This processing will, for example, break long frames of data up into cells and reassemble at the other end. In addition, for data traffic it should implement a Frame Check Sequence (FCS) calculation to identify frames containing errors. It may also have a retransmission protocol to recover from data errors and lost packets etc. (Or it may just signal to the user that there has been a problem and allow the user to do recovery.)

Each type of network traffic requires different adaptation layer processing.

5.1 Control of Congestion

The flow control mechanisms used in existing software-based packet networks are not adequate for the high speed environment. Some types of traffic, voice or video for example, cannot have their delivery rate slowed down. You either have to process it or clear the circuit.

Also, traditional "rotating window" flow controls such as are used in traditional packet networks require complex processing in software within the network nodes. This conflicts with the requirement for totally hardware-based data routing.

The primary method suggested for control of flows in a high speed packet network is to control the rate of entry of packets to the network.

When a circuit is set up its throughput demands are assessed by a node that allocates capacity to the individual circuit (call). These demands are things like minimum guaranteed packet throughput rate, maximum allowed peak rate, priority (if any) and loss priority (the tendency for the network to throw away the packet when congestion occurs).

The method of operation suggested is that the attaching user node should control its rate of data presentation to the network through a system called "Leaky Bucket Rate Control" and *that the network should monitor this traffic at the network entry point to make sure that the end user node does not exceed its allowance.*

5.1.1.1 Leaky Bucket Rate Control

This mechanism is a control on the rate at which data may be sent into the network rather than a control of data flow through the network. Once data has entered the network there is no proposed control of flows except the implicit throughput capability of the links and nodes involved.

In concept, leaky bucket rate control operates as follows:

- A packet entering the network must pass a "gate" called the leaky bucket. This is really just a counter, which represents the number of packets that may be sent immediately on this path.
- In order for a packet to pass and enter the network the counter must be non-zero.
- The leaky bucket counter has a defined maximum value.
- The counter is incremented (by one) n times per second.
- When a packet arrives it may pass the leaky bucket if (and only if) the counter is non zero.
- When the packet passes the barrier to enter the network, the counter is decremented.
- If the packet has been delayed it will be released immediately after the counter is incremented.

Leaky bucket rate control may be operated on individual connections or it may operate on a group of connections such as all the connections on the same link or all the connections on the same virtual path (as in ATM). In addition, there may be a number of leaky buckets implemented in series to give a closer control of rates.

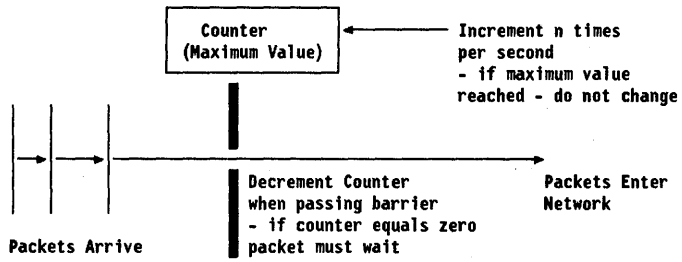


Figure 25. Leaky Bucket Rate Control

In some variations of the leaky bucket scheme there is no input queue to the leaky bucket! A packet arriving at the barrier is either allowed immediate passage or is discarded. From a network perspective it doesn't matter whether there is a queue there or not. The choice of whether or not to have queueing here depends very much on the type of traffic being carried and the design of the particular adaptation layer involved.

This scheme has the effect of limiting the packet rate to a defined average, but allowing short (definable size) bursts of packets to enter the network at maximum rate. If the node tries to send packets at a high rate for a long period of time, the rate will be equal to "n" per second. If however, there has been no traffic for a while, then the node may send at full rate until the counter reaches zero.

Paris (described in section 8.3, "Packetised Automatic Routing Integrated System (PARIS)" on page 158) uses two leaky buckets in series with the second one using a maximum bucket size of 1 but a faster clock rate. The total effect is to limit input to a defined average rate but with short bursts allowed at a defined higher rate (but still not the maximum link speed).

The scheme can be dynamic in that the maximum value of the counter and/or the rate at which the counter is incremented may be changed depending on

current conditions within the network (provided that the network has some method of signaling these conditions to the end user).

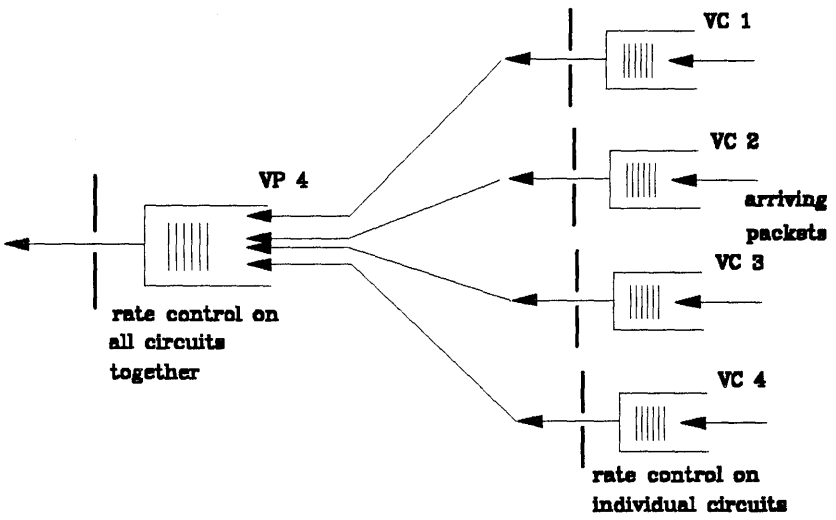


Figure 26. A Cascade of Leaky Buckets. Leaky bucket rate control is applied to individual circuits and then to the total of a logical group.

Figure 26 shows one potential configuration such that individual circuits have rate control applied to them and an aggregate rate control is applied to the total of a logical grouping of circuits. This is relatively efficient to implement in code.

5.2 Transporting Voice in a Packet Network

According to the international standard, when voice is converted to digital form, the analogue signal is sampled at the rate of 8,000 times per second (one sample every 125 µsec) and each sample is represented by 8 bits. This gives a constant bit rate of 64,000 bits per second.

The coding system is called "Pulse Code Modulation" (PCM). The basic concept of PCM is that each eight-bit sample is simply a coded measure of the amplitude of signal at the moment of sampling. But this can be improved upon by a system called "companding" (Compression/Expansion). It happens that the signal spends significantly more time in the lower part of the scale than it does at the peaks. So what we do is apply a non-linear coding so that the lower amplitude parts of the waveform are coded with more precision than the peaks. (In basic concept this is just like the "dolby" system for improving the quality of tape recordings.) In practice, PCM is always encoded this way but the standard is different in different parts of the world. One system is called "µ-law" and the other "A-law".

In order to transport this across a packet network individual samples must be assembled into packets. The principle is described below.

5.2.1 Basic Principle

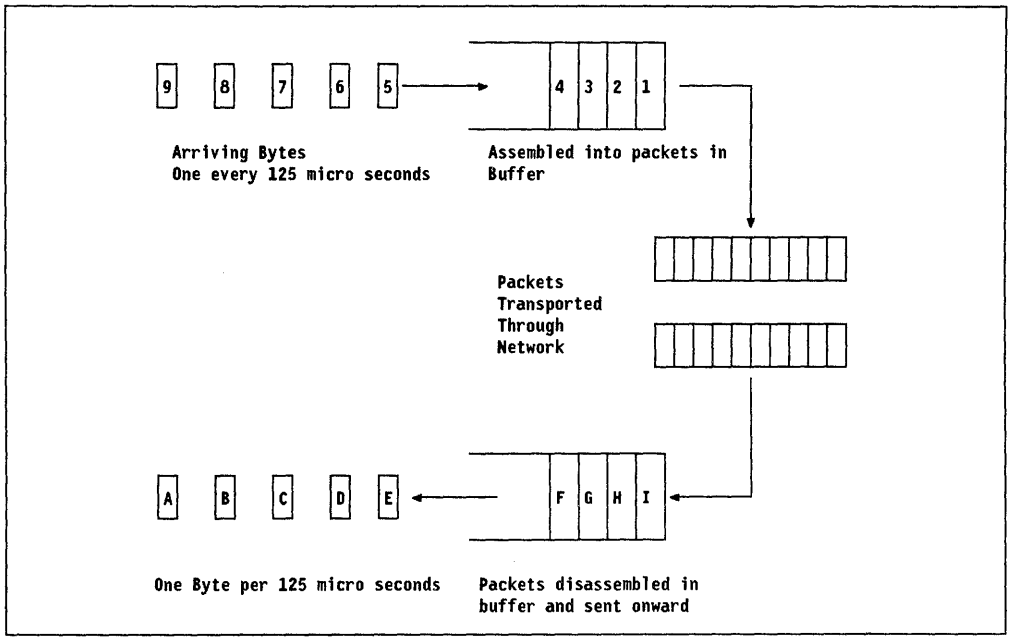


Figure 27. Transporting Voice over a Packet Network

Figure 27 above illustrates the principle of sending voice over a packet network.

1. The telephone handset generates a stream of 8-bit bytes of voice information at the rate of one every 125 µsec.
2. The digitised voice stream is received into a buffer until a block the length of a packet has been received.
3. When the packet is full it is sent into the network.
4. Once the packet is received at the other end it is disassembled and sent to the destination at the rate of one byte every 125 µsec.

A number of points should be made about this principle.

- The end-to-end delay as experienced by the end users will be the time taken to assemble a packet *plus* the transit delay through the network.
- If the network delivers packets to the destination packet disassembler at an uneven rate, then buffering will be needed at the destination to smooth out irregularities in the packet delivery rate.
- Most packet networks deliver packets at an uneven rate.
- What must happen is that when the circuit is established the receiving packet disassembler must hold the first packet for some length of time sufficient to overcome the largest possible variation in transit delay before sending anything on to the receiver.

  This increases the end-to-end delay significantly.

5.2.2 Transit Delay

The transit delay in a network is the time it takes a packet to travel through the network. This is made up of transmission time, propagation delay and node delay (processing and queueing delays within a node).

5.2.2.1 Transit Delay Variation

The problem with most packet networks is that the transit delay varies with the instantaneous load on the network.

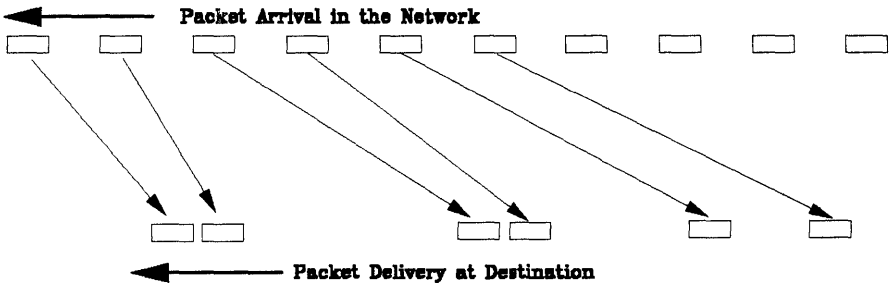


Figure 28. Irregular Delivery of Voice Packets

Figure 28 shows the regular arrival of packets into a network and the irregular rate of delivery of those packets. As mentioned earlier, to overcome these variations in network transit time we need to have a deliberate delay at the receiver and a queue of arriving packets so that the delays are not apparent to the receiver.

There is another problem here and that is that the bit timing of the "playout" operation cannot ever be quite the same as the timing of the transmitter - unless of course, there is a universal worldwide timing reference available. In continuous transmission, there will be occasional overruns or underruns at the receiver (clock slips) due to this lack of clock synchronisation.

5.2.2.2 The Effect of End-to-End Network Delay on Voice Traffic

The end-to-end delay experienced in a voice situation is made up of three components:

1. Packet Assembly Time

   The time it takes to assemble a packet or cell. Using the ATM standard 48-byte cell, we will need at least a 4-byte header (in addition to the cell header) leaving 44 bytes. At 64 Kbps (PCM code) this gives a packet assembly time of 5.5 milliseconds. For 32 Kbps the assembly time is 11 milliseconds.

   As more exotic coding schemes are employed, the packet assembly time increases as the required data rate decreases.

2. Network Transit Time

   This depends on the structure of the network but should be less than one millisecond per node traversed *plus* propagation delay at about 5.5 µsec per kilometer.

3. Delay Equalisation

This is the deliberate delay inserted immediately before the receiver in order to smooth out the effects of transit delay variation (jitter). Depending on the characteristics of the network this delay could be set somewhere between two and ten milliseconds.

The delay characteristics of the network are very important for two reasons:

1. Long transit delays cause the same effect subjectively as the well known "satellite delay". (This is 240 milliseconds one way.) Most people find holding a voice conversation over a satellite circuit difficult. The effect is one that many people never become accustomed to.

   There is some argument over what is an acceptable delay. Some academics say 100 milliseconds others 150. But all agree that a one way delay of 90 milliseconds or so causes no subjective loss of quality to most people.

2. The problem of echoes. Experience shows that when the delay is 45 milliseconds or more there is a potential problem with echoes.

   The primary source of echoes is the "hybrid" device[26] where the connection to the end user is carried over a two-wire analogue circuit. Another source is reflections on the two-wire line itself.

   In the case where the connection is fully digital from one end to the other the situation is controversial. In a recent paper on the subject (Sriram et.al. 1991. see bibliography) the authors argue that echo cancellation is not needed in situations where the circuit is fully digital from end to end. Other people say that there is mechanical feedback caused in some telephone handsets and that this is a source of echo that cannot be eliminated by the fully digital circuit. This is an important and unresolved issue.

   The importance rests in the fact that while echo cancellation technology is very good indeed, echo cancellers cost money.[27] In small countries where distances are short, network providers have only installed echo cancellers on international connections. A requirement for echo cancellation could add significant cost to their networks. In larger countries (such as the USA or Australia) propagation delays are so great that echo cancellation is a requirement anyway.

5.2.3 Voice "Compression"

There are various ways available to reduce the data rate required in a voice circuit from the 64 Kbps standard rate. A coding scheme which reduces the data rate to 32 Kbps without measurable loss of quality is called Adaptive Differential PCM (ADPCM). In concept this encodes each sample as the difference between it and the last sample, rather than as an absolute amplitude value.

There are many ways of voice compression which rely on the fact that a voice signal has considerable redundancy. (You can predict the general characteristics of the next few samples it you know the last few.)

PCM and ADPCM are very good indeed in terms of quality. It is very difficult for a listener to detect the difference between an original analogue signal and one that has gone through encoding and later decoding. And because digital

---

[26] See section 2.3.4, "Echo Cancellation" on page 37.

[27] Papers in the technical literature suggest that to build a digital echo canceller for this environment using a digital signal processor (DSP) requires a DSP of about 5 MIPS. This outweighs all the other functions performed in a digital packetiser/depacketiser (PADEP) by a ratio of five to one.

transmission is perfectly accurate, there is no loss in quality no matter how far the signal travels.

Nevertheless, even though we can't hear a quality loss, a small loss does take place. This was the reason that the 64 Kbps standard for digital voice was adopted in the first place. In large public networks, (such as in the USA) the transition between the existing analogue system and a universal digital system was (is) expected to take a very long time. During that transition, a call through the network may go through the conversion from analogue to digital and back again many times. Quality loss adds up, little by little. The standard was chosen because it was felt that there would need to be as many as *seven* conversions from analogue to digital and back again along the path of some calls.

5.2.3.1 Variable Rate Voice Coding

One of the ways of encoding voice looks to see when there is no actual speech and just stops sending data during the gaps.[28] This is not a new principle - it was used in the past over long distance analogue circuits but is much improved using digital techniques. Speech does occur in "talk spurts" and it is half-duplex (most of the time only one person is talking). This means that about 60% of any (one way) voice conversation, consists of silence. Why transmit silence?

There are many techniques available for encoding voice in this way. In the encoded form the conversation consists of short bursts of packets. A device called a "Voice Activity Detector" (VAD) is used to turn the encoding process on or off. It also should be noted that even within a period of speech the encoded information rate is variable.

One characteristic of the VAD is that it suppresses echoes. Provided the echo is at a relatively low level, the detector will stop encoding the signal. However, this is not perfect because when both parties talk simultaneously (a not unknown phenomena) each party could hear an echo of his/her own speech mixed up with the voice of the other speaker.

A reasonably good quality variable rate voice coding scheme should result in a peak data rate of around 20 Kbps or a little more during talk spurts and an average rate (in each direction) of around 10 Kbps.

Thus variable rate voice puts a statistical load onto the network, but variable rate coding does *not* remove the need for fast and uniform network transit delays.

5.2.3.2 Encoding Priority Schemes

In the previous section 5.1, "Control of Congestion" on page 77, it was seen that one method of alleviating congestion is to discard packets when there is a problem. If the packets can be coded in some way such that there are "essential" and "discardable" packets we can hope for a situation where all that happens when the network becomes congested is a graceful degradation in quality of service.

One suggested method is to code the voice packets in such a way as to put "essential" and "quality improvement" cells in different packets. This is

---

[28] An excellent discussion on this subject may be found in *A Blind Voice Packet Synchronisation Strategy*.

conceptually shown in Figure 29 on page 84. The most significant bits of each sample are placed into the same packet and the least significant bits into a different packet. The packets are marked in the header to say which packet may be discarded and which one may not.

When the packets are presented at their destination for playout (after buffering for a time) if a low priority packet is missing, the decoder can extrapolate and although the voice quality is affected the signal is still understandable.

The example shown above is intentionally very simple. In practice the coding schemes used in this way will be variable rate ones and the algorithm will be much more complex than just a selection of bits by their significance. Nevertheless, the principle is still the same.

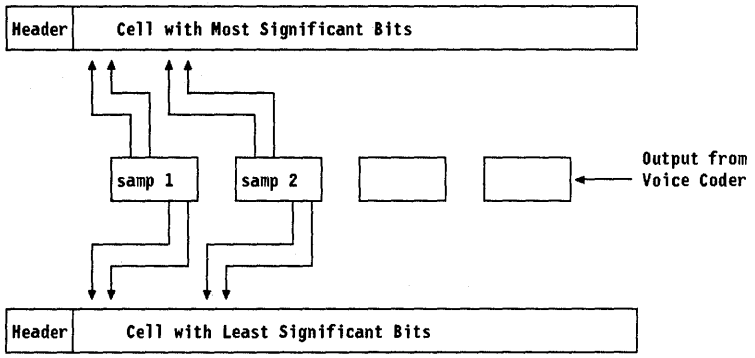


Figure 29. Assembly of Packets for Priority Discard Scheme

5.3 Transporting Video in a Packet Network

The transmission requirements of digital video were discussed in section 4.1.4, "Characteristics of Digital Video" on page 66. As far as the network is concerned video traffic is very similar to voice in the sense that both require a timing relationship to be maintained between the sender and the receiver. Packets or cells must arrive at a regular rate if the receiver is to maintain a stable picture.

There are some differences from voice however:

- The absolute amount of bandwidth required is enormous compared with telephone quality voice transmission.
- The quality required of a video transmission varies widely with the application. Broadcast quality requires a significantly greater bandwidth than does remote education or video conference applications. Video Phones require even less (as low as 128 KBps).
- While a raw video signal is a very high constant rate, the characteristics of video make variable rate coding schemes significantly more effective than they are for voice. The problem is that the amount of variation is extreme. A still picture, properly coded has an information content of 25 bits per second. A piece of very fast action may require an instantaneous rate of over 100 megabits per second.

  Voice traffic occurs in spurts but the extremes of variation in throughput requirement are not at all the same.

- Video is much more redundant than voice and a "glitch" is perhaps less significant. A missed frame here and there will hardly be noticed.
- The natural coding of video (because of the amount of information) is in large blocks (in excess of a thousand bytes).
- Most video is not interactive. Broadcast quality video is almost never so. For one-way video we don't have the strict network delay problems that exist with voice. We can afford to have a large reassembly buffer and a playout delay of perhaps several seconds to compensate for transit delay variations in the network.
- Interactive video is usually accompanied by voice and so tighter transit delay requirements are needed but video does not need to be exactly synchronised to the voice in any case. A 100 millisecond difference is quite acceptable. Thus we can afford a 100 ms playout buffer for video traffic even if it accompanies a voice signal.
- Encoding priority schemes such as described earlier for voice are also available for video traffic (see section 5.2.3.2, "Encoding Priority Schemes" on page 83). This enables packets carrying "less essential" parts of the signal to be discarded by the network during periods of congestion.

The biggest problem with video is just the enormous data rate that is required. If the peak data rate required by a single video user (or even a small number of users) is a significant percentage of the total capacity of the network then there is potentially a serious congestion problem. For example, if a broadcast quality signal fully variable rate encoded required a peak data rate of 50 megabits per second (even though the average might be say 10 megabits per second) and the base network uses 140 megabit internode links, (that is, a single user can take up 30% of one resource) then there is a potential congestion problem. The safe planned utilisation of a network (for stable operation) in this situation might be as low as 30%.

As the number of users increases and the capacity of the network increases the problem becomes less and less significant. One hundred broadcast quality video users with characteristics as described above will require perhaps 1,000 megabits per second *but the maximum total peak requirement might be no more than 1,200 megabits per second.* In the previous example, the peak requirement of a single user was four times the average requirement. In the case of a hundred users, the peak (for practical purposes) is only 20% greater than the average. This is the result described in Appendix B.1.4, "Practical Systems" on page 287.

5.4 Transporting Images

Image traffic is really not too different from traditional data traffic. Images range in size from perhaps 40 kilobytes to a few megabytes. If the user happens to be a real person at a terminal, sub-second response time is just as valuable for images as it always was for coded data transactions.

In the "paperless office" type of application, image users tend to spend more "think time" looking at the screen once it is displayed. That means that the transaction rates per terminal tend to be lower but perhaps that is because most of the experience to date is with systems that are very slow in displaying the image and the user is thereby encouraged to get all the information possible from one display before looking at the next.

In engineering graphics (CAD) applications, interaction can be as often as once a minute and the user demands sub-second response time for megabyte sized images.

Of course, images can be compressed and ratios of four to one are about average. This reduces network load and speeds up transmission time.

Image systems are only in their infancy in 1992 but many people consider that they will become as common as interactive coded data systems are today. Storing the enormous quantity of data required is a greater problem than transmitting it (actually, transmitting it is the easy part).

An open question for the future is "what will be the effect of very high quality displays". It is possible today to buy color displays with a resolution of 4,000 points by 4,000 points with 256 colors and excellent quality. (The main use of these to date has been in air traffic control, military applications and in engineering design.) The picture quality is so good that it rivals a color photograph. The point here is that images with this level of resolution are many times larger (even compressed) than the typical formats of today.

If these high resolution systems become popular then there will be significantly higher requirements for the network.

5.5 Transporting Data in Packets or Cells

The term "packetisation" refers to the process of breaking up blocks of user data into shorter blocks (called "packets") for transmission through the network.

Packets

The term "packet" has many different meanings and shades of meaning depending on the context in which it is used. In recent years the term has become linked to the CCITT recommendation X.25 which specifies a data network interface. (See Appendix D, "An Introduction to X.25 Concepts" on page 293.) In this context a packet is a fixed maximum length (default 128 bytes) and is preceded by a packet level header which determines its routing within the network.

In the late 1960s the term "packet" came into being to denote a network in which the switching nodes stored the messages being processed in main storage instead of on magnetic disk. In the early days a "message switch" stored received data on disk before sending it on towards its destination.

In a generic sense "packet" is often used to mean any short block of data which is part of a larger logical block.

The major advantages of breaking a block of data into packets for transmission are:

1. The transit delay through the network is much shorter.
2. Queues for intermediate links within the network are more easily managed and offer a more uniform delay characteristic. See Appendix B.1.4, "Practical Systems" on page 287.
3. Buffer pools and I/O buffers within intermediate nodes can be smaller and are more easily managed.

4. When an error occurs on a link (whether it is an access link or a link within the network itself) then there is less data to retransmit.

The great disadvantage is that the processing time, both in the network nodes themselves and in the attaching equipment, is greatly increased. As described elsewhere in this document, most software driven data switching equipment takes about the same amount of processor time to switch a block regardless of the length of that block. (This is not exactly true due to the effects of I/O interference but that is usually trivial.) For example if a 1 KB block is broken up into eight 128-byte packets then the load on the network switching nodes is multiplied by eight.

5.5.1.1 Transit Delay

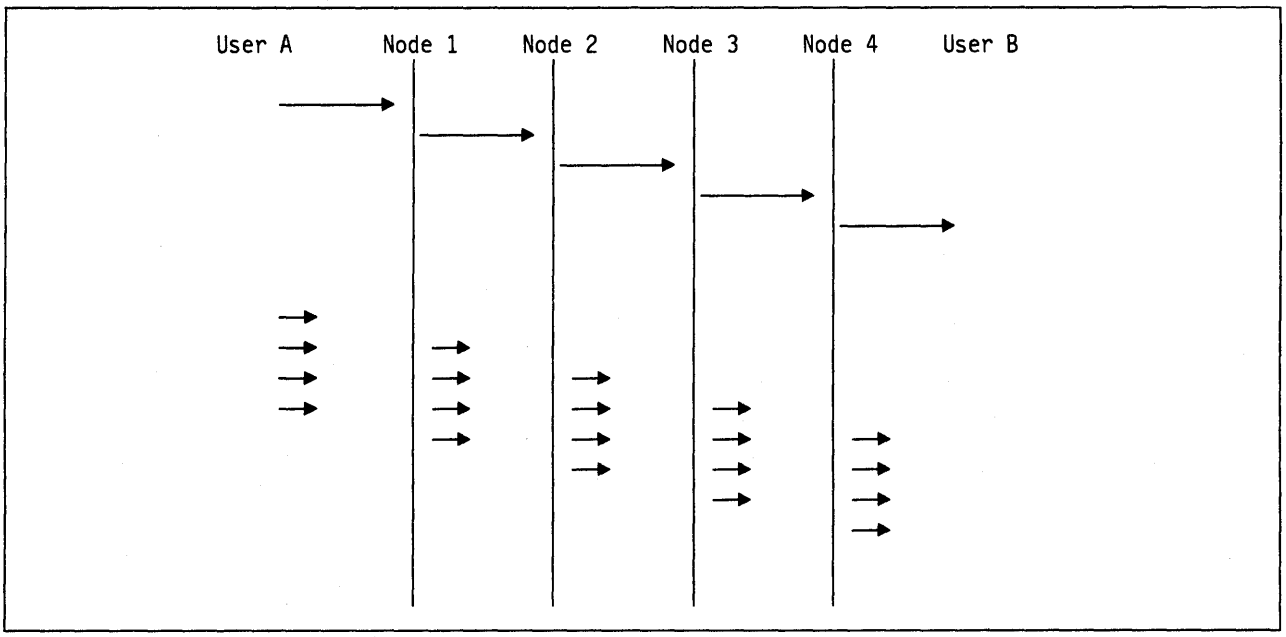


Figure 30. Effect of Packetisation on Transit Time through a 4-Node Network

Assume User A, in Figure 30 has a block of 1024 bytes to send through a 4-node network to user B. Assume also that the link speeds are the same, the nodes are infinitely fast and that there is no other traffic.

- User A sends to Node 1 and takes (for our discussion purposes) 4 units of time.
- Node 1 sends to Node 2 also taking 4 units of time.
- Node 2 sends to Node 3.
- And so on until the message arrives at User B.

The total time taken has been 5 times 4 units = 20 units of time.

Now, if the 1024-byte block is broken up into four, 256-byte packets then the following scenario will occur:

- User A sends the first packet to Node 1, taking 1 unit of time.
- Node 1 sends this packet to Node 2, but while this is happening User A is sending packet 2 to node 1.
- While User A is sending the third packet, Node 1 is sending a packet to Node 2 and Node 2 is sending a packet to Node 3.

- This happens through the network until the last packet arrives at User B.

It is obvious from the diagram that sending the message as small packets has reduced the network transit time to 7 units compared with the 20 units needed without packetisation. This is due to the effect of overlapping the sending of parts of the message through the network.

5.6 Connection Oriented versus Connectionless Networks

One distinguishing characteristic of a network (or network protocol) is the presence or absence of a "connection" between the end users. When a connection is present the network is called "Connection Oriented" and when there is no connection the network is called "Connectionless".

After 30 years of building data networks, this is still an issue on which there is considerable disagreement and which can evoke strong feelings.

5.6.1.1 Connectionless Networks

In a connectionless network, a network node does not keep any information relating to interactions currently in progress between end users of the network. Every data block transmitted must be prefixed by the full network address of both its origin and its destination.

Sometimes, (such as in the pre-1979 versions of SNA[29]), the network address takes the form of a structured binary number, which can be used relatively easily to determine the appropriate routing for the data. Sometimes, (such as in typical LAN networks), the network address has no meaningful structure that is usable for routing purposes.

Characteristics of connectionless networks are as follows:

- When a data block arrives the network node must calculate on which outbound link to send the data towards its destination. This decision may be very complex and compute intensive or very simple depending on how much information about the destination is available within the destination address field.

  In the extreme case where the destination address contains no information at all about its location, (such as is the case with LAN addresses), the node may need to keep tables relating every known destination address to its real location in the network. This is the case with "transparent bridges" between LANs.

  Usually, the destination address will be structured in some way such that it can be related to knowledge of the network's topology kept within the node. For example, the network address may contain a destination node number and the switching node may contain a network map so that it may calculate the best outbound path on which to forward the data block. This process can be very simple (such as in the first version of SNA) or very complex (such as in ARPANET or TCP/IP).

- If the network allows multiple paths to be used for individual data blocks then blocks will arrive in a different sequence from the sequence in which they were delivered to the network.

[29] In early SNA each node had a number. When a frame was routed by a node there was a single table showing which link traffic for a given node number must be sent on. The switching node knew nothing about routes or about connections.

If blocks are able to be delivered out of sequence then a much more complex end-to-end protocol will be required to resequence them before presentation to the end user.

- The header prefix required in a connectionless situation is typically much longer than for a connection oriented network. This affects the efficiency of the network as headers take up link capacity and require transmission time. This may or may not be important depending on the length of the data blocks being handled.
- There is no need for a connection establishment sequence for one end user to send data to another. To send data an end user just has to put the destination address onto the front of the message and send it. This saves time and overhead.
- Implementation of flow and congestion controls in a connectionless network is much more difficult than in a connection oriented one because individual connections (though they of course exist) are unknown by the network and thus cannot be controlled.

5.6.1.2 Connection Oriented Networks

In a connection oriented network, once a connection is established, there is no need to place a destination address in the block header every time a data block is sent. All that is needed is an identifier to specify which connection is to be used for this block.

There are many ways of constructing connection oriented networks. For a description of the method used in SNA APPN see section 5.7.3, "Logical ID Swapping" on page 93.

Characteristics of connection oriented networks are as follows:

- The connection must be established somehow. Permanent connections (such as PVCs in X.25) are typically established by a system definition procedure. Temporary connections (such as SVCs in X.25) are typically established by placing a "call".

  Setting up a call can take considerable processing overhead in network nodes and can often take a significant delay (such as five seconds).
- Congestion control is easier than for connectionless networks because network nodes can regulate the flow on individual connections.
- Data switching is usually (but not always) significantly more efficient than for connectionless networks because the onward route of the data is predetermined and therefore does not need to be calculated.
- When a link or a node becomes inoperative (goes down), connections that were passing through the affected link or node are typically lost. A new connection must be established through a different route. This takes time and usually disrupts the connection at the end user level.

  Connectionless networks typically reroute traffic automatically around link or node failures.

5.6.1.3 Connection Oriented Connectionless Networks

In a series of token-ring LANs connected by "source routing" bridges we see the case of a connection existing over a fixed route where the individual switches (the bridges) do not know about connections at all. This principle is described in section 11.4, "Source-Routing Bridges" on page 251.

The IBM experimental high speed packet switch called "Paris" uses a similar method of switching to that described above for TRN bridges. (See the description in section 8.3, "Packetised Automatic Routing Integrated System (PARIS)" on page 158.) Paris is different because the data switching part has no record of the existence of connections. The routing decision is made by the switching nodes completely on information present in the header of every data message. (The control processor in each node does know about connections passing through the node's switching part, as it allocates capacity and monitors throughput for congestion control purposes.)

Connections do exist in both these systems but in either system it is only the source node that knows about it.

5.6.1.4 Connections across Connectionless Networks

Consider the diagram in Figure 31.

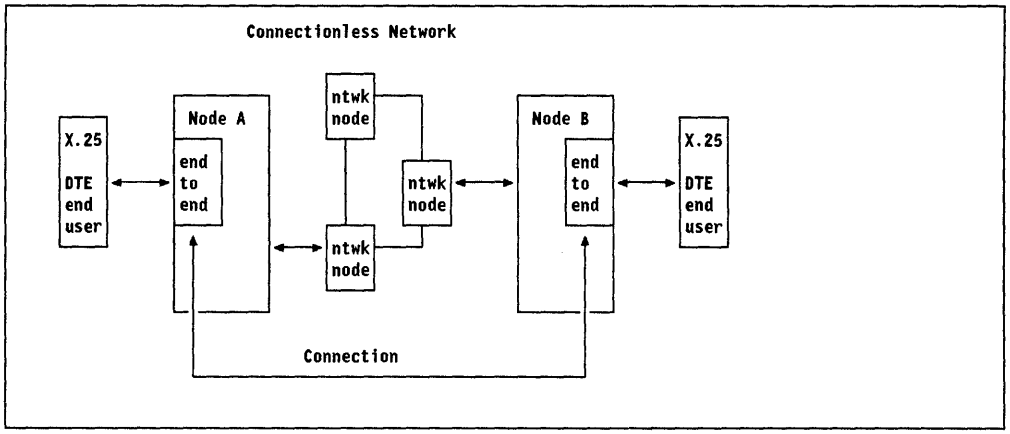


Figure 31. A Connection across a Connectionless Network

This is an important case in practical networks.

In the example there is a connection between the two end users. This connection is known about and supported by nodes A and B, but the switching nodes in the network do *not* know that a connection exists. The end-to-end function holds the network address and the status of its partner end-to-end function and looks after data integrity and secure delivery etc.

What exists here is really a connection oriented network built on top of a connectionless one.[30]

[30] The IBM X25Net product which implements an X.25 network works exactly in this way.

5.6.1.5 A Connection Is Always Present

Well, almost. In an application sense, it is very rare for communication to take place without a connection being present logically, even if the components of the network sometimes do not know about it.

5.6.1.6 Connections in SNA Networks

In SNA the only entity that can send or receive data is the Logical Unit (LU). Data is *always* sent from one LU to another on a connection called a "Session".

But inside SNA networks sessions are handled differently.

1. In the first versions of SNA, network nodes (always IBM 3705s running the Network Control Program, NCP) selected the link on which to forward a given data block by the binary "subarea number" (really destination node number) within the destination address field of the data header.

   Since the network was (and still is) constrained to deliver data blocks in sequence there could be one and only one possible path from any given origin node to any given destination.

2. Since about 1980, SNA "Subarea" networks (networks in which the real network address is structured such that the node number is present as a subfield) have been connection oriented. That is, there are predefined routes through the network which are known by each network node. Sessions are still unknown to the network nodes (or, more correctly, to the transmission network component of the network nodes) but are carried on connections called Virtual Routes (VRs). VRs map to explicit routes (ERs). The destination subarea (node) number together with the ER number in the frame header is used to determine the routing of incoming data blocks.

   The routing headers used by this form of SNA are 26 bytes long comprising origin and destination network addresses (each of 48 bits) and Explicit Route and Virtual Route numbers.

3. SNA APPN networks select a new route for each end user session through the network. The APPN network nodes keep connection tables which record a fixed (for the duration of a session) relationship between a session identifier (called an LFSID) on one link with a session identifier on another link. For a more detailed explanation of this form of routing see section 5.7.3, "Logical ID Swapping" on page 93.

   This means that the routing header for SNA APPN is only six bytes (two of which are the LFSID).

5.7 Route Determination within the Network

There are two aspects to routing within a network:

1. Determining what the route for a given connection shall be.
2. Actually switching (routing) the packet within a switching node.

There are many methods of determining a route through a network. So long as this can be performed before the connection is set up (or during the connection establishment) it doesn't matter much to the switching nodes. For very high throughput, the critical item is that the switching element must be able to decide where to route an incoming packet in a few microseconds.

5.7.1 Dynamic Node by Node Routing

In this method there is no route determined when the connection is set up. This is due to the fact that there is no connection. Each packet is sent into the network with its full destination address imbedded in the header. Each node knows the current network topology and loadings and is able to decide where and on which path the packet should be directed.

This process can be very fast (for a software technique). It is the principle behind "Arpanet" routing and that of TCP/IP. The IBM product X25Net uses this method of routing internally and achieves a throughput of up to 1,000 packets per second *on a PS/2* (depending on the model used as switching node).

Special switches exist which use this type of technique and achieve throughputs of 50,000 packets per second.

But this is a software based technique. It is very difficult to see how it could be efficiently implemented in hardware. Packet rates of millions per second are not likely to be achieved by this method any time soon.

5.7.2 Source Routing

In the source routing method the origination node (or interfacing function) is responsible for calculating the route the packet must take through the network. A routing vector is appended to every packet sent and that vector is used by intermediate nodes to direct the packet towards its destination.

This method is used in the IBM Token-Ring implementation for routing through a bridged token-ring network. (See section 11.4, "Source-Routing Bridges" on page 251.) It is also used in the IBM research project called "Paris". (See section 8.3.2, "Automatic Network Routing (ANR)" on page 160.)

In this method the sending node must either know the network topology or it must use some method (such as broadcasting) to find the optimal route. But once the route is determined, intermediate switches do not need to refer to any system tables or parameters to make the routing decision. The next stage of the route is right there in the packet header.

A drawback of this method is that the routing vector in the packet header takes some storage and is an overhead. But this is quite small and the benefits of being able to make a fast routing decision outweigh the small increase in bandwidth overhead.

5.7.3 Logical ID Swapping

Logical ID swapping is an internal network technique for routing data through a network of packet switched nodes. While the technique is widely used in existing networks and not specifically a high speed technique, it is regarded by many as an appropriate technique for supporting high speed networks supporting Frame Relay and ATM. A connection oriented technique (see section 5.6, "Connection Oriented versus Connectionless Networks" on page 88), it is used widely by many different networking systems.

Notice that logical ID swapping is an *internal* process for routing packets or cells within a network. These internal network protocols are typically *not* specified by international standards.

This technique is used in IBM APPN networks and in a number of proprietary networking protocols. It may also be used in networks supporting:

- X.25
- Frame Relay
- Asynchronous Transfer Mode (ATM)

Networks that use this technique typically multiplex many connections (or sessions) on a link using some form of logical "channelisation". That is, each block (or frame) sent on the link has a header which includes an arbitrary number identifying which logical connection that this block (or frame) belongs to. Systems vary in the rules governing how the number is allocated and how a path through the system is defined but the principle is the same.

- In APPN, the logical channel identifier is called a Local Form Session Identifier (LFSID) and is located in the format 2 (FID_2) transmission header. The connection is called a session.
- In X.25 the logical connection is called a virtual circuit. The identifier is called a logical channel and is contained within the packet header.
- In Frame Relay the identifier is called the DLCI (Data Link Connection Identifier) and this is located in the address field of the link header. The connection is called a virtual link.
- In ATM the identifier is called a Virtual Channel Identifier (VCI) and it is situated in the cell header.

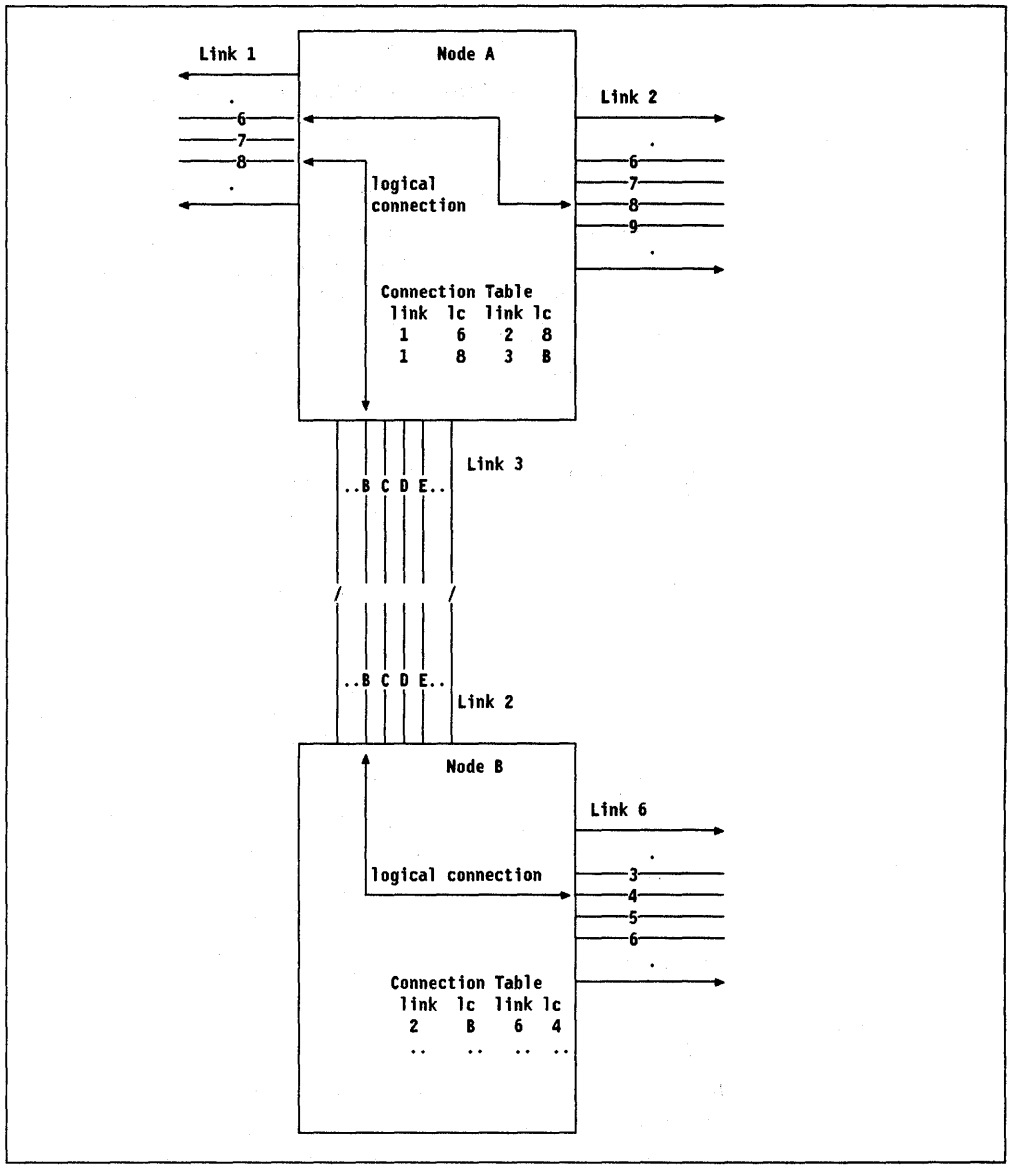


Figure 32. Data Networking by Logical ID Swapping

5.7.3.1 Data Transfer

In Figure 32 there is a logical connection between an end user system connected to link 1 (ID 8) on node A and another end user system attached to link 6 (ID 4) on node B. In operation the procedure works as follows:[31]

1. The end user system places data to be sent on a link to the network. (In the example, link 1 on node A.)
2. The user data has a header appended to it which contains a logical channel identifier. (In the example, lc 6.)
3. Node A receives the block and looks at the header. It finds that the received block has lc 6.

[31] Another example in this document is ID swapping used with frame relay. See Figure 54 on page 146.

4. The node then looks at the connection table for lc 6 on link 1 (the node knows which link the data was received on).
5. The node finds that lc 6 on link 1 is connected to lc B on link 3.
6. The node then changes the lc ID within the data header from 6 to B.
7. The node then queues the data for transmission on link 3.
8. When node B receives the data it sees that it belongs to lc B. (Notice here that both nodes know the same lc number at each end of the link because it is the same link. The lc number only has meaning in the context of the single link between node A and node B.
9. Node B then repeats the process changing the ID to "4" and sending it out on link 6.

5.7.3.2 Determining the Route

There are many ways of determining the route and setting up the connection tables.

- It could be done by a central node and updates sent to each node along the path when the connection is set up.
- It could be determined by the originating node (or a node providing route calculation support to the originating node). If this is done, the route can be sent out in a special message which travels along the specified route and signals the control processor in each node to set up the connection tables.

  This is exactly what happens in APPN. In APPN, when a session is set up, a routing vector is included in the connection (session) setup packet (the BIND). As the BIND progresses through the system, each node in the path builds a connection table entry for the new session.
- It can be done in a distributed way by allowing each node that receives the setup message to calculate the next hop along the path.
- It may be predefined through a system definition process.

The point about this is that connection setup is relatively infrequent (compared to the routing of data blocks) and is not too time critical. A connection setup time of 200 milliseconds (or even a second or two) is quite tolerable in even a very high speed network.

Of course, in most systems the tables do not exist in the form suggested in the example - they will be set up in whichever way is most efficient within the using system.

5.7.3.3 Characteristics

The system has the following characteristics.

Minimal Bandwidth Overhead

The ID part of the header is the only essential field for this method of routing. The systems described above use between 10 and 20 bits for this field.

Fixed Route(s)

Frames (packets, cells...) flow on the fixed predetermined route. This means that frames will (in most systems) arrive in the same sequence in which they were sent.

Efficient Switching

Relatively few instructions are required to perform the switching function. In order to route a packet towards its destination reference must be made to the connection tables. So whatever process performs the switching must have very fast access to the tables. In addition, when a new connection is set up or an old one is terminated the tables must be updated. (The database of network topology and loadings can, of course, be maintained quite separately.)

In a software based system (traditional packet switch) this is no problem at all as the function that maintains the tables shares the same storage as the code that does the switching.

In a system using a hardware based routing mechanism, this mechanism must have very fast (sub-microsecond) access to the tables. The updating function must also have access though its demands for access are not as critical (it can wait a bit).

This means that in a hardware implementation you need to have a shared set of tables that is instantly accessible to the hardware switch and also easily accessible from the control processor.

This can increase the cost of the implementation above that of the ANR technique. (See section 8.3.2, "Automatic Network Routing (ANR)" on page 160).

Requires Connection Setup

The techniques require that the connection tables be set up and maintained dynamically. This is a processing overhead in each node and makes "datagram" transport quite inefficient.

5.8 End-to-End Network Protocols

When you increase the speed of the links in a network or even the throughput rate of the network nodes one major factor does not change - propagation delay. (Sadly, we can't increase the speed of light - at the present level of technology anyway.)

Network layer protocols perform the following functions:

- Data error detection (if the underlying network does not detect errors in the data).
- Data error recovery by retransmission (if the underlying network does not guarantee the integrity of the data).
- Recovery from lost frames or packets.
- Packet or frame resequencing (if the underlying network does not guarantee the delivery of packets in the sequence in which they were delivered to the network).

Existing network protocols that perform the above functions are generally viewed to have too many "turnarounds" in them. This applies to OSI (layer 4), TCP (especially) and most other common end-to-end protocols. An excellent description of this problem may be found in *A Survey of Light-Weight Transport Protocols for High-Speed Networks.*

The result of this is that as we speed up the underlying packet network, application throughput (especially for interactive applications) does not improve very much in many cases. When we have a high speed network we would like it to give better performance than traditional packet networks have delivered in the past.

Considerable work is proceeding in the research community aimed at producing an efficient end-to-end (layer four) protocol for use with high speed networks. One such proposal is called XTP. (A good description of XTP may be found in: *The Xpress Transfer Protocol (XTP) - A Tutorial*.)

The primary object of a good network layer protocol for the high speed environment is to minimise the number of "turnarounds" (when a sender must wait for an acknowledgement from the receiver) in the protocol. To do this the following principles may be employed:

- Optimism. Assume that the underlying node hardware and links are very reliable. Errors will occur and they must be recovered, but they will be very infrequent.
- Assume that the user (program) at the other end of the connection is there and ready to receive all that we send. Don't use a "handshaking protocol" to find out first.
- Try to put as many small packets into one block as reasonably possible (if your network can process large frames).
- Design the network protocol for fast connection setup and termination. Perhaps "piggyback" the first data packet onto the control packet that sets up the connection.
- Clearly define the interface to higher layer (session and presentation layer) functions and minimise the number of interface crossings necessary to transfer a block of data.

5.8.1 SNA in a High Speed Network

SNA is unique in that it uses a minimal network layer protocol, preferring to put the error recovery functions into the network. At the network layer SNA assumes that the network it is running over is very stable and reliable.

When SNA networks are run "over the top" of other networks (such as X.25 or Frame Relay or a LAN) then we must do something to make sure that the underlying network is very reliable. (This was a design decision in the early days of SNA, to put the reliability (and its attendant cost) into the backbone part of the network and avoid the cost of providing extensive network layer function. At the time, this was optimal from a cost point of view.)

In the case of X.25, an X.25 virtual circuit is regarded as a link in SNA. But it is a special link. No end-to-end link protocol is run over it because of the cost and performance implications. This means that to run SNA over a packet network (X.25), that network must be reliable. If the network loses data on a virtual circuit and signals this to SNA, SNA will see this as the unrecoverable loss of a link and all sessions traveling on that link are immediately terminated. Application level error recovery is then necessary in order to resume operation.

In some SNA systems there is an optional network layer protocol (called "ELLC") which does give full error recovery over an unreliable network connection. This

was developed in the early days of packet networks and today these networks are generally very reliable and ELLC is considered unnecessary.

When SNA is run over a LAN network or a Frame Relay network where the loss of a frame or two occasionally is a normal event an end-to-end protocol is used to stabilise the connection. This is called IEEE 802.2 and is a "link layer" protocol as far as the ISO model is concerned. Nevertheless, when an end-to-end protocol is run across a network in order to recover from network errors, it is just an academic point to discuss what OSI layer is being obeyed. The function here is network layer class four regardless of how it is performed.

The big problem with running SNA over a disjoint packet network (even a LAN) is that the network management and directory functions are not integrated with those of SNA. This means that you run two networks, SNA and something else underneath where neither network knows about or can coordinate properly with, the other.

If a fast packet switching architecture was to be fully integrated into SNA then a different structure would be possible.[32] The "transmission network" part of SNA could be replaced with a fast packet switching architecture. In addition to the minimal adaptation function, a full network layer would need to be interposed between the SNA end users and the network to maintain the stability of the service to the end user. This is exactly what is done with LAN and Frame Relay support now but done somewhere higher up in the protocol stack. A system like this offers the potential benefit that both the network management and directory systems of SNA, and the underlying fast packet network, could be integrated and work together providing a single unified network interface to the end user.

[32] IBM has not announced any plan for doing what is suggested here. The discussion is included in order to bring into perspective the interaction of SNA with high speed networks. IBM cannot comment on future plans.

5.9 A Theoretical View

Packet switching architectures may be better understood by comparing the functions performed by the protocols against one another. ATM requires that nodes implement only the functions of the three sublayers of the physical layer. Other packet switching techniques require more complexity, as shown in the architectural model given in Figure 33 on page 100. In general the higher you go up the stack the more complex processing in a network transit node becomes.

The degree of complexity can be best understood if the layer one and two functions needed in non-ATM packet service are broken down into sublayers.[33] If the functions performed by HDLC/SDLC link control are broken up in this way we get the following structure:

Sublayer 1A

This layer adapts to the medium characteristics, transmission coding and things like bit timing and jitter tolerance.

Sublayer 1B

This is the MAC function when many devices are connected to a shared medium. In SDLC link control this is the control of "mulitdropping"; in a LAN architecture it is the physical (but not logical) control of transmission of data onto the shared medium. In Sonet this is the TDM function.

Sublayer 1C

This is the ATM cell transfer function. It doesn't exist in traditional DLC designs.

Sublayer 2A

This layer builds the frame for passing to the physical layer. Addition of the framing flags (B'01111110') and the provision of transparency (by bit "stuffing") is done here.

Sublayer 2B

This function is responsible for routing the data to different destinations (that is, the address part of the header).

Sublayer 2C

This function provides the Frame Check Sequence (FCS) bytes at the end of the frame and is responsible for error detection.

Sublayer 2D

This layer handles error recovery by retransmission and link level flow control (by window rotation).

[33] The sublayering concept described here is due to Vorstermans et. al. 1988. See bibliography.

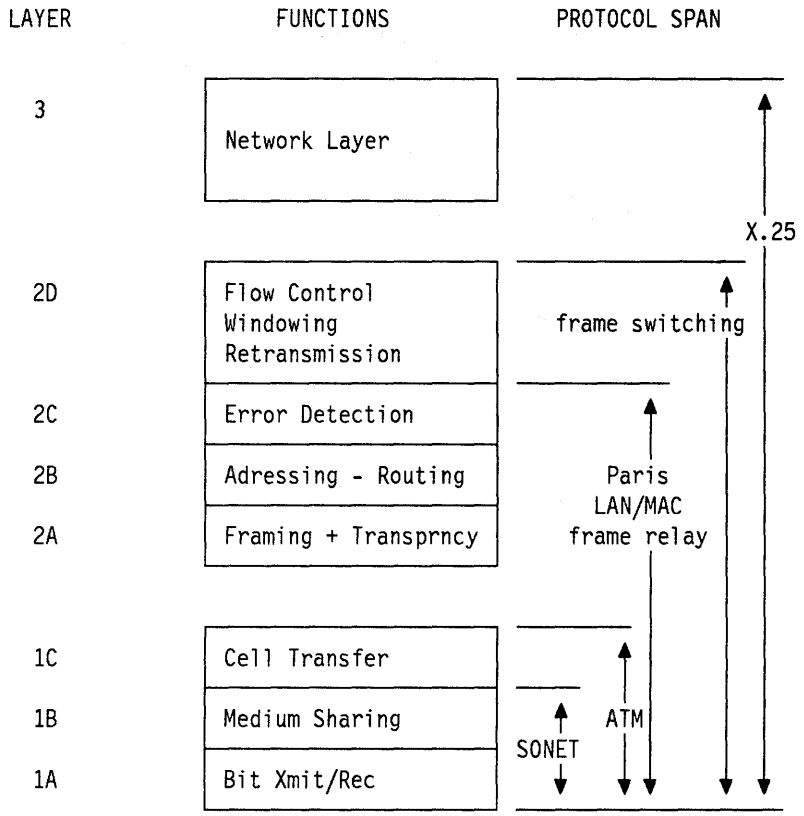


Figure 33. Protocol Span of Packet Networking Techniques

As mentioned elsewhere in this document, ATM requires that only the physical layer be implemented in network nodes.

Different networking techniques use different sublayer functions:

X.25 requires the full implementation of all three layers. Layer 1 (without the ATM sublayer), layer 2 (LAPB) and layer 3 (Packet Protocol).

Frame Switching requires the full implementation of layer 2 (that is, sublayers 2A, 2B, 2C, and 2D), thus including the complexity of retransmission, windowing, and flow control.

Frame Relay, LAN/MAC and Paris all provide much the same level of function. This requires sublayers 1A (optionally 1B) along with sublayers 2A, 2B and 2C.

ATM (without considering the AAL) uses all three sublayers of layer 1. With the AAL it is equivalent to Frame Relay.

Sonet is a TDM system and uses only the function of the first two sublayers of layer 1.

5.10 Summary of Packet Network Characteristics

The following table summarises the characteristics of the packet networking techniques described in this document and compares them with two traditional packet networking technologies (APPN and X.25).

Table 1. Packet Switching Technology Comparison

Function	APPN	X.25 Packet Switching	Frame Relay	Cell Relay (ATM)	Paris
Throughput (typical)	500 to 30,000 packets per second	500 to 30,000 packets per second	1,000 to 100,000 frames per second	10 to 100 million cells per second	100,000 to 500,000 frames per second
Access Speed	up to 2 Mbps	up to 64 Kbps	up to 2 Mbps	E3 (35 Mbps) +	E3 (35 Mbps) +
Packet Size	Variable	128 bytes	Variable	48 + 5bytes	Variable
Standards body	n/a	CCITT	CCITT and ANSI	CCITT	n/a
Standards Approval	n/a	1980	1990	1992 (frame), 1996 (service)	n/a
Bandwidth Management	Flow Controls	Network Dependent	Network Dependent	Input Rate Control	Input Rate Control
Routing	Label Swapping	Not Specified	Not Specified	Label Swapping	Source
Error Detection	Full	Full	Full	Header Only	Full
Error Action	Recovery by Retrans-mission	Network Dependant	Discard Frame	Discard Cell if Bad Header	Discard Frame
Real Time Voice	No	No	No	Yes	Yes
Full Motion Video	No	No	No	Yes	Yes
Switched Calls	Yes	Yes	Not Yet	Yes	Yes

Chapter 6. High Speed Time Division Multiplexing Systems

6.1 Integrated Services Digital Network (ISDN)

Some people believe that ISDN is probably the most significant development in communications since the invention of the automatic telephone exchange.

ISDN describes and specifies a (digital) user interface to a public (digital) communications network. It *does not* describe the internal operations of the communications network - just the interfaces to it and the services that it must provide.

Much of the impetus for ISDN comes from the desire to use the installed copper wire "subscriber loop" at higher speeds, with more reliability and new services. In most countries the total value of installed copper wire subscriber loop connections represents the largest single capital asset in that country. There is an enormous potential benefit in getting better use from it.

Existing copper subscriber loops vary widely in quality and characteristics and making use of it for digital connections represents a significant technical challenge. See section 2.3.3, "The Subscriber Loop" on page 35.

This document is not concerned with the details of narrowband ISDN. The subject is discussed because it illustrates the use of modern digital signaling techniques and TDM operation.

6.1.1 Types of ISDN

There are three generic types of ISDN.

Narrowband ISDN is the form of ISDN that is becoming widely available today. There are two forms of access (Basic Rate and Primary Rate) and the service offered is the connection of 64 Kbps channels primarily on a switched service basis. There is also a low rate "connectionless" packet switching ability available through the "D" channel.

Thus narrowband ISDN offers TDM connection of 64 Kbps channels through a switched network. Channels normally **cannot** be faster than 64 Kbps - if you use two channels this does not give a single 128 Kbps channel, it gives two, unrelated 64 Kbps channels.

Special equipment may of course be designed to synchronise multiple B channels and provide a wider single channel; indeed, there is equipmen available already which will do this. However, this equipment is not a part of the network.

Wideband ISDN is a form of ISDN where a user *is* able to access a wider synchronous data channel through using a group of adjacent slots on an ISDN primary rate interface. Thus if 6 slots are used, they may form a single, 384 Kbps data channel (this is called an H0 channel).

Some ISDN public network equipment on the market today allows this option but most PTTs do not yet offer this as part of their ISDN service even if their equipment allows it.

Broadband ISDN does **not** provide high speed synchronous channels. It is a cell based packet switching system.

Since both narrowband and wideband ISDN offer synchronous TDM derived "clear" channels many people assume that broadband ISDN will be similar. In fact broadband ISDN is based on "Asynchronous Transfer Mode" (ATM) cell switching[34] and works in a very different way. This is because of the problem of allocating variable bandwidths over a Time Division Multiplexed (TDM) trunk circuit as described below in section 6.2.7, "The Bandwidth Fragmentation Problem" on page 123.

6.1.2 The ISDN Reference Model

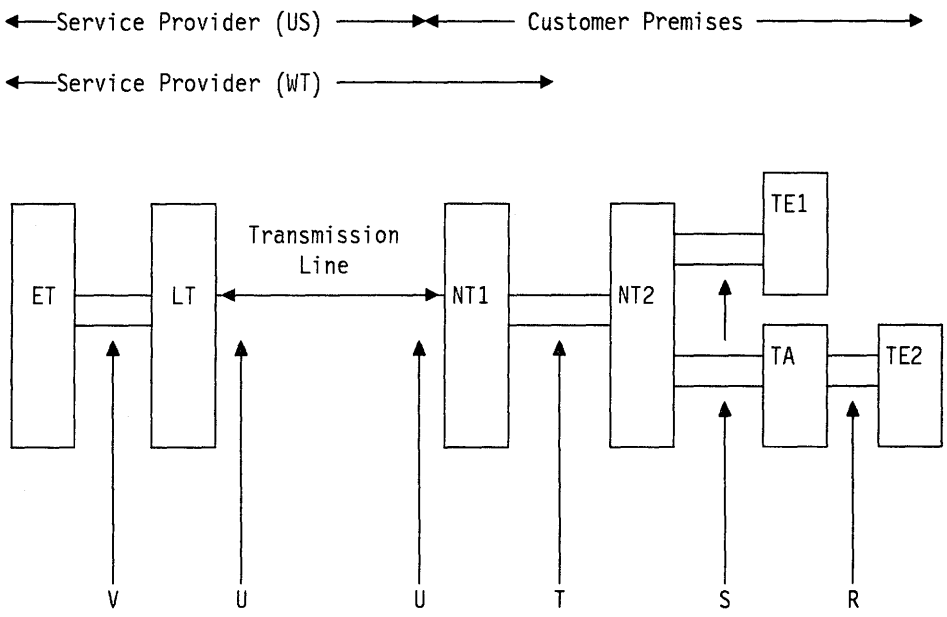


Figure 34. ISDN Reference Configuration

The CCITT has defined ISDN in terms of a set of functions with fixed relationships to one another and with defined interfaces between them. These are illustrated in Figure 34. The function being performed is digital transmission between a customer (end user) and a public telephone exchange. The functions defined are as follows:

ET Exchange Termination.

LT Line Transmission Termination. This is termination of the transmission line at the exchange. (LT and ET can be in the same physical piece of equipment.)

NT1 Network Termination 1. This function terminates the subscriber loop transmission line and provides an interface for customer equipment. In Basic Rate ISDN, NT1 changes the transmission protocol and reformats the frame.

NT1 has a different status in the US from its status in other countries. In the US, NT1 is legally considered "customer premises equipment" (it is

[34] See section 7.1, "Asynchronous Transfer Mode (ATM)" on page 129.

"like" a modem[35]). In the rest of the world, NT1 is considered service provider equipment. Thus in the US it is possible for the NT1 function to be integrated within end user equipment, while in other countries it will be a separat physical box.

NT2 Network Termination 2 is a PBX or communication controller function.

TA The Terminal Adapter function connects non-ISDN terminals to the ISDN network.

TE2 Terminal Equipment 2 is just the new name for all "old style" terminals with RS-232/422/449, V.24/35 or X.21 (generically called "RVX") interfaces.

TE1 Terminal Equipment 1 is terminal equipment designed to interface directly to the ISDN system.

R, S, T, U, V designate reference points where protocols are defined.

6.1.3 ISDN Basic Rate

For people hitherto unfamiliar with digital communication techniques, perhaps the most surprising thing about ISDN is its connection to the small end user (such as a small business or a private home). This "ISDN Basic Rate Interface " (BRI) uses the same twisted pair of copper wires (subscriber loop) as is currently used for a home telephone. This single pair of wires carries **two** (64 Kbps) B channels and a (16 Kbps) D channel.

Thus, on the same physical wire as exists today a user has two independent "telephone lines" instead of one *plus* a limited ability to send data messages to other users without the need to use either of the two B channels.

"B" (Bearer) Channel

A "B channel" is 64 thousand bits per second in both directions simultaneously (64 Kbps full-duplex). When a user places a "call" (for voice or for data) a continuous path is allocated to that call until either party "hangs up" (clears the call).

This is the principal service of (regular) ISDN. The B channel is an end-to-end "clear channel" connection (derived by TDM techniques) which may be used for voice or for data. It can be thought of in the same way as an analogue telephone connection but, of course, it is digital.

"D" Channel

The D channel is not an "end-to-end clear channel" like the B channels. This D channel carries data in short packets and is primarily used for signaling between the user and the network (when the user "dials" a number, the number requested is carried as a packet on the D channel). The D channel can also be used for sending limited amounts of data from one end user to another through the network without using the high capacity B channels[36] (this can be useful in a number of applications).

In Basic Rate the D channel operates at 16 Kbps.

---

[35] It is nothing like a modem.

[36] This is a facility that is available in some networks and not in others.

The ISDN BRI allows for up to eight physical devices to be attached simultaneously to the network by means of a "passive bus". Obviously, since there are only two B channels a maximum of two devices can use a B channel simultaneously. However, all eight devices can simultaneously share access to the D channel for low speed data applications.

6.1.3.1 The ISDN Basic Rate "U" Interface

The "U" interface is the interface between the Network Terminating Unit (NT1) on the customer's premises and the local telephone exchange.

This interface must transfer user data at a total rate of 144 Kbps full-duplex over an existing *two-wire* subscriber loop. This is a complex problem, because signals must travel in both directions over the same physical wires; and, therefore, one device receiving a signal from another device will also receive an echo of its own transmissions in the form of interference with the received signal.

As discussed in section 2.3.3, "The Subscriber Loop" on page 35, this connection is variable in quality and has all kinds of impairments. Nevertheless, it is considered very important that the protocol should operate over *any* subscriber loop connection *without* the need to select especially good ones or to perform any "conditioning".

The U interface is *not* internationally standardised. The CCITT considered that technology was changing so rapidly in this area that the presence of a standard would be likely to inhibit future development. In addition, since the installed subscriber loop varies markedly in characteristics from one country to another (different average distance, different wire gauges) this was felt to be a necessary freedom. Thus in Europe, the PTT supplies the Network Terminating Unit (NT1) and is free to use any protocol on the U interface that it deems appropriate.

In the US, the situation is different. For legal reasons, all equipment on customer premises must be open to competitive supply (especially the NT1). To enable this the American National Standards Institute (ANSI) has published a standard for the BRI U interface. In the US therefore, a supplier of EDP equipment may decide to integrate the NT1 function within a terminal or a personal computer and connect to the U interface.[37] In the rest of the world suppliers of EDP equipment *must* connect to the S (or T) interface.

In practice there are three U interface standards in general use:

US

This country uses true full-duplex transmission over the two-wire subscriber loop. To achieve this the interface circuits must use very sophisticated adaptive echo cancellation techniques (see section 2.3.4, "Echo Cancellation" on page 37).

The line code used is the 2B1Q code described in section 2.2.12.1, "2-Binary 1-Quaternary (2B1Q) Code" on page 29. The data rate is 160 Kbps (144 Kbps of user data plus framing and a maintenance channel) but the signaling rate is only 80 kbaud. The line code was chosen because in the environment of an impaired transmission channel

[37] As of January 1992 all ISDN Basic Rate interface adapters announced by IBM use the S interface only.

a lower signaling rate helps with echo cancellatio n and makes the signal easier to receive.

The maximum practical distance over which this technique will work is about 12 kilometers.

Germany

In Germany, true full-duplex transmission is also used. The line code is 4B3T code at a data rate of 160 Kbps and a signaling rate of 120 Kbps (see section 2.2.10, "4-Binary 3-Ternary (4B3T) Code" on page 26).

As in the US, the receiver must use advanced adaptive echo cancellation techniques.

The maximum distance for transmission (without special repeaters) is 4.2 kilometers using .4 mm wire and 8.0 kilometers using .6 mm wire. These maxima match the characteristics of subscriber loops in Germany.

Other Countries (Time Compression Mode)

As stated above, country telecommunication carriers are able to decide which protocols to use on the U interface (and perhaps use many different ones).

A common protocol in use in many countries does *not* use simultaneous full-duplex bidirectional transmission. Short bursts of data are sent at a higher data rate, but in one direction at a time. This gives the effect of real full-duplex transmission, but without the complexity and cost of using adaptive echo cancellation. This technique is sometimes called "Time Compression Mode" or "Burst Mode".

The method of operation is as follows:

- A frame of 38 bits (illustrated in Figure 35 on page 108) is transmitted from the network exactly once every 250 microseconds.
- The transmission data rate is 384 Kbps and "regular" half-bauded AMI transmission code is used (see section 2.2.5.1, "Alternate Mark Inversion" on page 19). The signaling rate is the same as the bit rate - 384 kbaud. This means that the transmission time for a frame is 99 µsec.
- The frame carries two bytes from each B channel and four bits from the D channel. This gives each B channel an average rate of one byte every 125 µsec, which equals 8000 bytes per second or 64 Kbps. The D channel rate is 16 Kbps.
- When the NT1 receives a complete frame it waits 5.2 µsec "guard time" (to allow reflections to dissipate a bit) and sends its frame up to the network.
- This means that the maximum time available for signal propagation is:

  $250 - 2 \times (99 + 5.2) = 42.6$ µsec

  Since propagation must take place in both directions the maximum propagation delay in one direction is half of this - that is, 21.8 µsec.

  Since propagation speed on most subscriber loops is between 5 µsec and 5.5 µsec per kilometer, the maximum available distance using this technique is about 4 kilometers.

The advantage of the technique is that it is simple, easy and relatively low in cost (saves the cost of the special adaptive echo canceller chip required by the other techniques). The disadvantage is the rather short maximum distance available, though of course, you could speed up the transmission rate and get greater distances if needed.

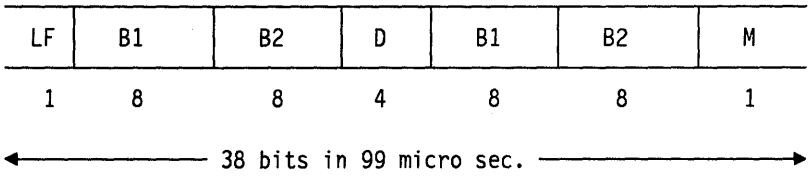


Figure 35. "U" Interface Frame Structure. The M channel is used for service and maintenance functions between the NT and the exchange.

6.1.3.2 Multiframing

The M channel used in Time Compression Mode (TCM) is a simple example of a "superframe" or "multiframe". Within each 38-bit TCM frame a single bit is used to provide two independent clear channels. Every second bit (the M bit in every second frame) is for a "service channel". Since the frame rate is one every 250 µsec this gives an aggregate rate for the channel of 2048 bits per second.

One bit in every four frames belongs to the "service channel". Since the frame rate is 4096 frames per second (one every 250 µsec) the service channel operates at 1024 bits per second.

Of course there is a need to identify which bit is which. To the "higher level" 38-bit frame, the M bit forms a single clear channel, but within itself it has a structure. A multiframe begins with a code violation (that is, the M channel bit violates the code of the 38-bit frame). This delimits the beginning of the multiframe. After the code violation (CV) the next M channel bit belongs to the transparent channel, the next to the service channel, the next to the transparent channel and then the next multiframe starts with another CV.

The concept of multiframes (frames within frames) is very important as it is used extensively within ISDN Primary Rate and Sonet/SDH.

6.1.3.3 The ISDN Basic Rate Passive Bus ("S" Interface)

Simple, elegant and efficient, the passive bus is an excellent example of what can be achieved with modern digital signaling technology.

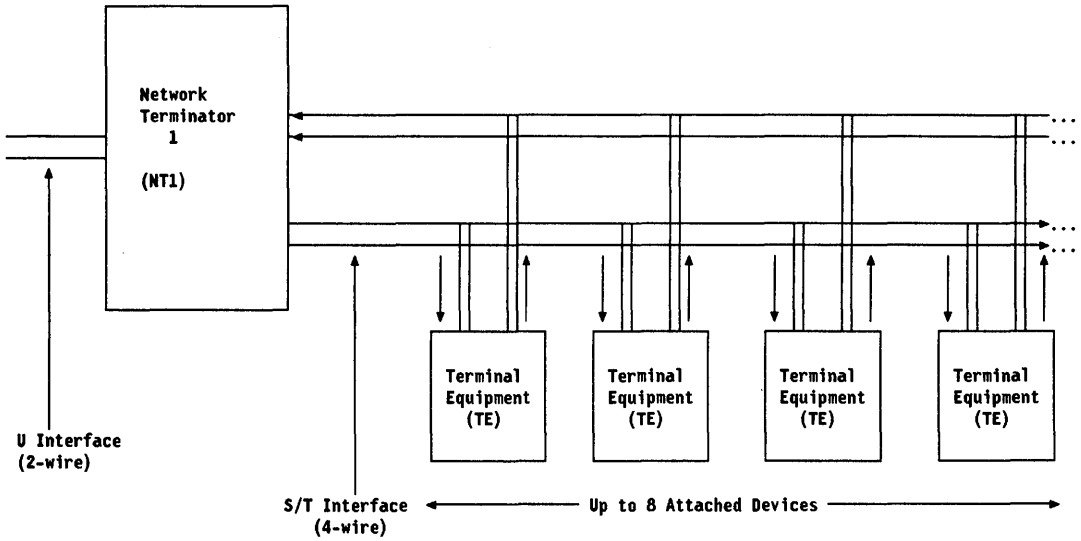


Figure 36. ISDN Basic Rate Passive Bus

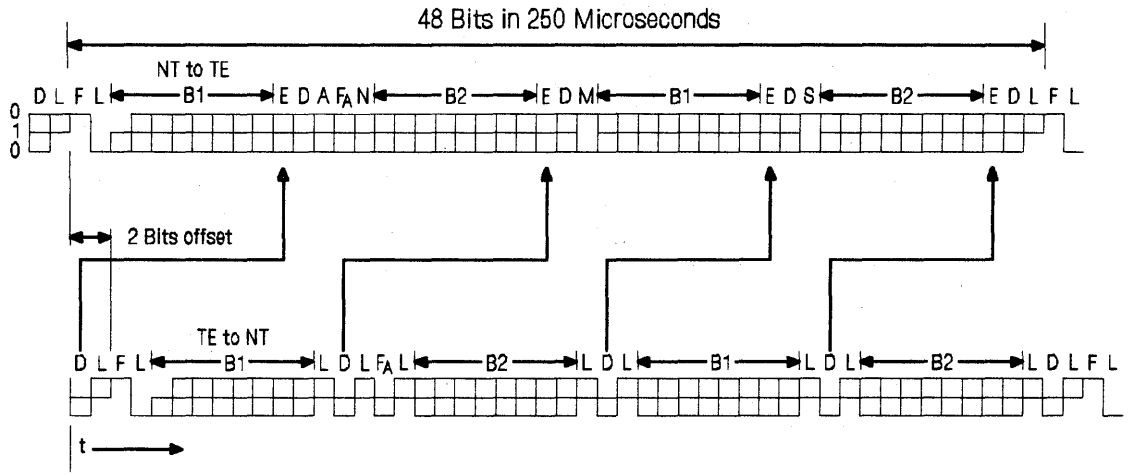
Consider Figure 36. The Network Termination Function (NT1) is here a separate physical device. NT1 operates on the U interface on its upstream side (to the exchange) using two wires. On the downstream side it must communicate with up to eight separate "terminal equipment" devices connected to the passive bus. To do this it uses the "S" interface protocol.[38]

The passive bus consists of just four wires connected (through a transformer) to the NT1. Two wires are used for transmission and two for reception. Up to eight TE devices are able to "plug in" to this bus and to operate independently of each other. The protocol is performed by all connecting TEs and by the NT1.

The objective is for each TE to be able to communicate with the network using either (or both) of the B channels and the D channel. Since a B channel is a clear channel (that is, the device may send any sequence of bits without restriction), only one TE may use a B channel at any one time. The D channel is a packet channel and TEs are constrained to use a rigorous protocol (called "LAPD" for Link Access Protocol D channel) for operation.

The frame formats and method of operation of the S and U interfaces are very different and this means that NT1 has quite a lot of function to perform. Nevertheless, it is important to realise that NT1 passes both the B and D channel data through to the exchange *without any form of modification or processing of the data*. NT1 does frame reformatting and electrical signal conversion only - it does not change or look at the data.

[38] This same protocol is used for the "T" interface also, so it is often referred to as the S/T interface protocol.



F = Framing bit
L = DC balancing bit
D = D-channel bit
E = D-echo-channel bit
F_A = Auxiliary framing bit or Q-bit

N = Bit set to a binary value N = F_A
B1 = Bit within B channel 1
B2 = Bit within B channel 2
A = Bit used for activation
S = Reserved for future standardization
M = Multiframing bit

Figure 37. ISDN Basic Rate S/T Interface Frame Structure. The frame format used in the TE to NT direction is different from the one used in the NT to TE direction.

Frame Formats

Figure 37 shows the frame formats used on the S/T interface. The first thing to notice about these is that the formats are quite *different* depending on the direction of transmission (TE-to-NT, or NT-to-TE).

Frame Timing

Frames are sent in each direction, as nearly as possible, exactly one every 250 µsec. Since a frame is 48 bits long this means that the data rate is 192 Kbps.

Each frame contains two bytes from each B channel and 4 bits from the D channel. Note that these are spaced at regular intervals throughout the frame.

Frame Generation

In the NT to TE direction (of course) the NT generates the frame and puts data in each field. TEs can access (read) any or all fields in the frame as appropriate.

In the TE to NT direction things are different. There is no single device available to generate the frame so **all TEs generate a frame and transmit it onto the bus simultaneously**.

This would normally be a formula for disaster... multiple devices writing independent information onto the same wires at the same time. However, this is not the case. The line code and the frame structure are organised so that the framing bits will be the same polarity for all TEs. Thus, provided the timing is accurate, all TEs may write the same frame to the bus at the same time.

The line code used is Pseudoternary and is described in section 2.2.5, "Pseudoternary Coding" on page 18.

Frame Synchronisation

Each TE derives the timing for its frame transmission from the timing of the frame it receives from the NT. The TE starts its frame transmission precisely 3 bits after it receives the start of a frame from the NT and subsequent bits are transmitted using the clock derived from the received bit stream.

There are problems here with jitter and with propagation delay. Each TE will detect the frame from the NT at a (very slightly) different time (jitter). Therefore the TEs will not transmit exactly simultaneously. This effect is minimised by requiring a very high quality receiver.

Also, it takes time for electricity to propagate down the bus (about 5 µsec per kilometer). At 192 Kbps a bit is just over 5 µsec long. We can't do much about this! The result of the propagation delay problem is that the passive bus must limit the distance between the first TE on the bus and the last one. That is, TEs must cluster together on the bus.

For a different reason (operation of the D channel) there is a maximum distance (again due to propagation delay) between the NT and the first TE on the bus.

TE to NT Frame

Fields within the TE-to-NT frame may be put there by different TEs. For example, one TE may be using one B channel and another TE the second B channel. Of course, only one TE may use a B channel at any one time.

Because no TE can know what another TE is writing into the frame, each field that can be separately written has an additional DC balance bit (designated "L") to ensure the overall DC balance of the frame.

Each D channel bit has a separate DC balance bit and may be written by any TE (or all of them together).

6.1.3.4 Operation of the D Channel

The D channel operates quite differently from the B channels. Where the B channels are "transparent" (the TE may send or receive any arbitrary bit stream), the D channel is rigorously structured.

A TE on the D channel sends and receives short packets of data. One primary requirement is that TEs must share the D channel with some kind of fairness. Consecutive packets may belong to different TEs. All communication is from the network to/from individual TEs. There is no TE-to-TE communication possible (except through the exchange).

There is no problem with transmission from the exchange to the TE. Since there is only one device sending, it is able to intermix packets for different TEs at will. But in the other direction, multiple TEs cannot transmit data on the same channel at the same time. This is the same situation that existed in the past with multiple terminal devices sharing a multidrop analogue data link. As with analogue data links in the past, the problem is how to control which device (TE) is allowed to send at a particular time. In the past "polling" was used for this purpose but with digital signaling techniques available a much more efficient technique is used.

As far as the TDM structure is concerned (that is, at the "physical layer") the D channel is a clear channel of 16 Kbps full-duplex. As frames are transmitted or received by each TE, consecutive D channel bits are treated as though there was

no other information in between. A link control protocol is used between each TE (through the NT) to the exchange.

Operation of the D channel depends on the interaction of four things:

LAPD Link Control

LAPD is a version of the international standard HDLC. It is very similar to IBM SDLC, LAPB (the link control in X.25) and IEEE 802.2 LLC. As far as the S interface TDM structure is concerned, the important aspects are the frame structure and the address bytes.

A LAPD frame is started and ended by the binary sequence B'01111110'. This is a unique sequence since whenever a string of five one bits appears in the data, the transmitter must unconditionally send an additional single zero bit (later removed by the receiver). So the only time a string of zero followed by six ones occurs is when there is a frame boundary or an abort. The frame delimiter (B'01111110') is also sometimes referred to by the letter "F".

Immediately following the frame delimiter there is a two-byte address field. This will be discussed later but from the S bus perspective the important thing to note is that this contains a TE identifier (number) and is unique. It is the uniqueness that is important here.

Pseudoternary Line Code

This was discussed in section 2.2.5, "Pseudoternary Coding" on page 18. It is important to note that to send a "1" (one) bit the transmitter *does nothing*. That is, a one bit is represented by *no voltage* on the line. Zero bits are represented by alternating + or - voltages.

The D Channel Echo

In the NT-to-TE direction there is a channel called "D channel echo". Notice that this is in addition to the regular D channel.

The D channel echo is generated in the NT1 and has no function upstream towards the exchange. When a D channel bit is received from the TEs, it is immediately reflected back to the TEs in the echo channel. Every TE can see the status of the last D channel bit received from the TEs by the NT1.

Timing

Two timing aspects are critical.

1. As discussed above, each TE must generate and send a frame as nearly as possible at *exactly* the same time as each other TE. This is done by synchronising each TE's transmitter timing with its receiver and by limiting the distance between the first and the last TE on the bus.
2. A D channel bit sent in the TE-to-NT direction *must be received by its sender (on the D channel echo) before the TE sends the next D channel bit*. This characteristic is critical to the whole operation.

To receive a frame on the D channel a TE simply monitors the received D channel for a frame character followed by an address field containing its (the TE's) address. When this is detected, the TE receives the frame.

Sending a frame from the TE to the network is much more complex:

Monitor for Idle Channel

When the TE wants to send a frame to the network (through the NT1) it first monitors the D Channel Echo for a sequence of eight one bits. Continuous one bits denote an idle channel.

Send Flag and Address

Once the TE sees an idle D channel, it will begin transmitting its frame upstream, one bit at a time into the D channel bit positions of the TE-to-NT frame.

Collision Avoidance

It may have happened that another TE was also waiting to send. If it was, it may also have started transmission into the D channel at the time of exactly the same bit as the first TE did. (This is quite a likely event.)

After each TE has sent a bit on the D channel it must wait and listen for the echo. The echo must be received before the TE is allowed to send the next bit. The Pseudoternary line code makes the bus act as a logical OR. That is, within some limitations it is *legal* for multiple TEs to send data simultaneously.

If a TE receives a zero bit on the echo channel when the last bit it sent was a one, then it must immediately terminate its transmission (before sending the next bit). The TE that sent the zero will continue to send without knowing that another TE ever tried. Since address fields are unique (they always contain a unique TE number among other things), the TE with the highest numbered address will win the contention.

Send Data and FCS

By the time the address field is sent the contention operation will be over and the winning TE may go on to send a single frame of data. The TE appends a Frame Check Sequence (FCS) number to the end of the data to provide an error detection mechanism but this is treated as data as far as the D channel itself is concerned (NT1 doesn't know or care).

Monitor for Ones

When data transmission is complete the TE will monitor the D channel echo for a sequence of nine one bits instead of eight before it is allowed to attempt to send another frame. If it sees nine one bits, but has nothing to send, it will change its monitoring counter such that for subsequent blocks it will monitor for strings of eight ones again.

Another TE waiting for access to the D channel is allowed to start transmitting after only eight one bits and will thus obtain access before the first TE has another chance.

This procedure ensures that every TE is allowed the opportunity to send at least one frame before the first TE is allowed to send the next frame.

Priorities

It is possible to have low priority TEs (or low priority functions within a TE) that are allocated a lower priority. In this case the TE will wait to see ten one bits before sending and wait to see eleven one bits after sending but before sending again.

This ensures that TEs with an allocated count of eight always have sending priority over those with a count of ten.

Passive Bus - Theme and Variations: Within the above mode of operation, several configurations of passive bus are possible.

The Short Passive Bus is between 100 and 200 meters long (depending on the cable) and allows TEs to be placed anywhere on the cable (attached via 10 meter stubs).

The Extended Passive Bus insists that the TEs be grouped on the cable over a distance of between 25 to 50 meters (again depending on the cable characteristics). The maximum overall length allowed from the NT is 500 meters.

Point-to-Point operation is possible if only one TE is connected and in this case the length is limited by propagation delay to a maximum round trip delay of 42 microseconds. 42 µsec is the maximum round trip delay allowed such that the D channel echo can be received before the next D bit is sent.

Maximum length due to cable impedance is expected to be about one kilometer.

For operation on busses longer than about 200 meters, a special NT is needed which adapts for the time delay due to propagation.

LAPD Link Control: As mentioned above, LAPD is a member of the HDLC series of link control protocols.[39]

Each TE has at least one connection with the ISDN network. This connection is used for passing call requests and call progress information. The TE may also have other connections with services within the network. Thus, running on the same link, there are multiple TEs each (perhaps) with multiple simultaneous connections to different points within the network. This means that there is a need to address each endpoint uniquely.

In SDLC a single "control point" (which does not have an explicit link address) identifies up to 255 secondary devices using a single-byte link address. In LAPB (X.25), communication is always between peer entities and so the link address may take only two values (X'01' or X'03'). LAPD uses a two-byte address which contains two fields:

1. A SAPI (Service Access Point Identifier) which represents the service or endpoint within the ISDN network.
2. A TEI (Terminal Endpoint Identifier) which identifies the TE.

Link control operation is conventional.

ISDN Frame Relay: Frame Relay is a fast packet switching service which was designed to be used with ISDN. In fact its definition is part of the ISDN definition - albeit, it is an additional "service" which may be provided by network providers. The Frame Relay service may be accessed through a B channel, an H channel (ISDN wideband access) or through the D channel.

The basic service of delivering low rate packet data from one end user to another using D channel access is *not* Frame Relay but rather is a basic service of the ISDN network. Frame Relay is more fully described in section 8.2, "Frame Relay" on page 146.

---

[39] LAPD is described in detail in *IBM ISDN Data Link Control - Architecture Reference.*

6.1.4 ISDN Primary Rate Interface

Primary Rate (PRI) transmission uses two pairs of wires (subscriber loops) - one pair in each direction. Because of the use of unidirectional transmission the problem of interference due to echoes is much less. In addition, the service does not use unselected subscriber loops. Wire pairs are carefully selected for quality *and have repeaters inserted aproximately every 1.6 kilometers (mile).* This allows a much higher transmission rate.

Communication is always point-to-point since there may be only one device (typically a combined NT1/NT2 such as a PBX) connected to the primary rate interface.

In Europe, the transmission rate used is two megabits per second. In the US, the speed used is 1.544 megabits per second (the same as "T1"). This results in the availability of 30 B channels with one (64 Kbps) D channel in Europe with 23 B channels plus one D channel in the US. The systems are very similar and so only the European system is described here.

Wideband Channels: On the primary rate interface it is possible to take a group of B channels and use them as a single wideband data channel. This a feature of the ISDN definition but is implemented in only a very few real ISDN networks. An H0 channel concatenates six slots giving a single channel with an available bandwidth of 384 Kbps. There are also H11 channels (24 slots - 1536 Kbps) and H12 channels (30 slots - 1920 Kbps). H channels are end-to-end wideband transparent connections on which the user is entitled to put any data desired.

Coding and Frame Structure: Communication is point-to-point on four wires and so the electrical level coding and the frame structure is conventional.

In Europe the coding used is HDB3 which is described in section 2.2.8, "High Density Bipolar Three Zeros (HDB3) Coding" on page 24.

The frame structure is shown in Figure 38 on page 116. Slot 0 is used for framing and maintenance information. Slot 16 is a 64 Kbps D channel. All other slots may be used for B channels or as part of an H channel.

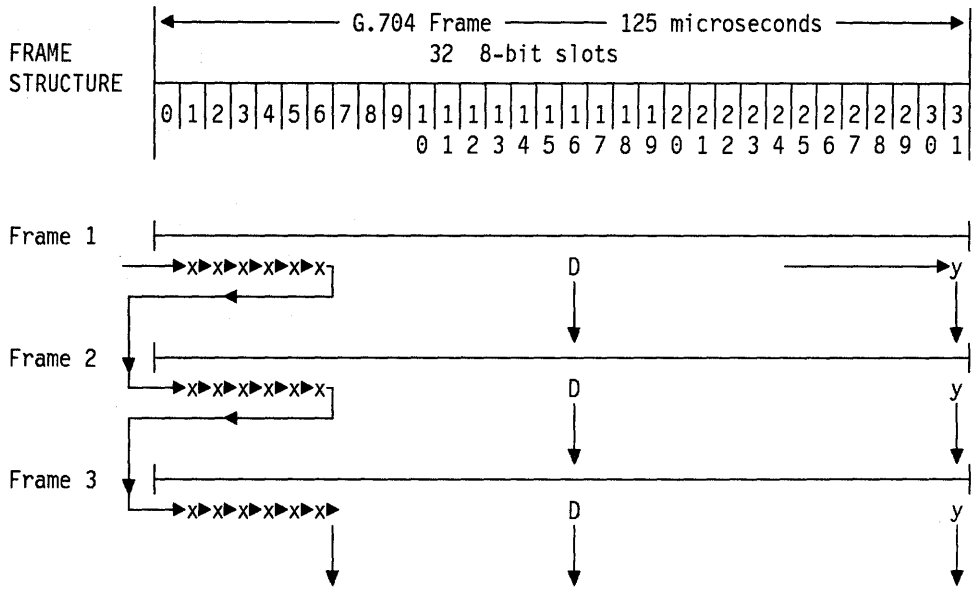


Figure 38. ISDN Primary Rate Frame Structure (Europe). Arrows show the sequence of slot concatenation for an H0 (wideband) channel, the D channel (slot 16) and a B channel (slot 31).

6.2 SDH and Sonet

Sonet (Synchronous Optical Network) is a US standard for the internal operation of telephone company optical networks. It is closely related to a system called SDH (Synchronous Digital Hierarchy) adopted by the CCITT as a recommendation for the internal operation of carrier (PTT) optical networks worldwide.

Sonet and SDH are of immense importance because of the vast cost savings that they promise for public communications networks.

Traditionally, public telephone company networks have been built by using a cascade of multiplexors at each end of a high speed connection. This principle is discussed in Appendix A.1.4, "Sub-Multiplexing" on page 278. In physical reality this resulted in the configuration illustrated in Figure 39.

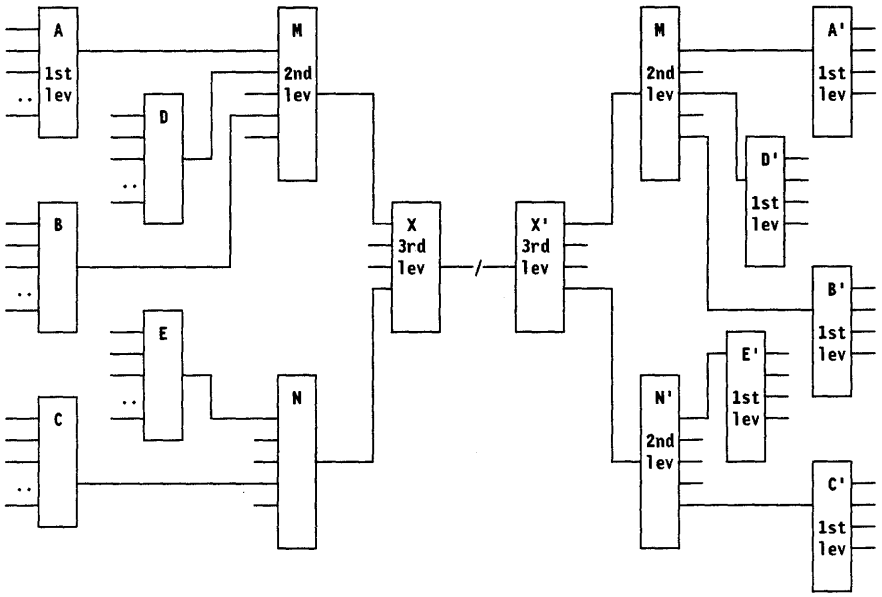


Figure 39. The Multiplexor Mountain. Each multiplexor illustrated exists as a separate physical device although many multiplexors may be mounted together on a single rack.

In order to use a high speed interexchange link it was necessary to multiplex a very large number of slower speed circuits onto it in stages. The faster the link, the more stages required.

There are a number of important points to remember here:

1. The internals of this structure are proprietary. Each pair of multiplexors in the system has to be manufactured by the same supplier. In the figure, the concept of multiplexor pairs is illustrated. A pair of multiplexors "see" a clear channel connection between them even though the connection might really go through several higher layer multiplexors. (In the figure multiplexors A and A′, B and B′ are multiplexor pairs.)
2. The multiplexing structure in the US is different from the structure used in Europe, and both are different from the structure used in Japan. This leads to compatibility problems when interconnecting systems between countries and also means that equipment designed and built in one country often cannot be used in another.

3. There is an enormous cost benefit to be gained by integrating the multiplexing function with the internal functioning of the telephone exchange and hence removing the multiplexors entirely. Modern telephone exchanges are digital time division multiplexors in themselves.
4. If access is needed to a single tributary circuit (or small group of circuits) then it is necessary to demultiplex the whole structure and then remultiplex it.

Sonet and SDH eliminate these problems. A single multiplexing scheme is specified that allows:

1. A standardised method of internal operation and management so that equipment from many different manufacturers may be used productively together.
2. Multiple speeds of operation such that as higher and higher optical speeds are introduced the system can expand gracefully to operate at the higher speeds.
3. Worldwide compatibility. A single optical multiplexing hierarchy which applies throughout the world and accommodates the existing speeds used in both Europe and the US.
4. Many levels of multiplexing and demultiplexing to be accomplished in a single step. (You don't have to demultiplex the higher levels to gain access to the lower levels.)
5. Many different payloads (different speed channels) to be carried through the system.
6. Access to low bandwidth (T1, E1 style) tributaries without the need to demultiplex the whole stream.
7. Considerably better efficiency than before. For example, the floating payload feature of Sonet eliminates the need for the customary 125 µsec buffers required at crosspoints in the existing (plesiochronous[40]) multiplexing schemes.

6.2.1 Sonet Structure

The basic structure in Sonet is a frame of 810 bytes which is sent every 125 µsec. This allows a single byte within a frame to be part of a 64 Kbps digital voice channel. Since the minimum frame size is 810 bytes then the minimum speed at which Sonet will operate is 51.84 megabits per second.

```
810 bytes × 8000 frames/sec × 8 (bits) = 51.84 megabits/sec
```

[40] See Appendix C, "Getting the Language into Synch" on page 291

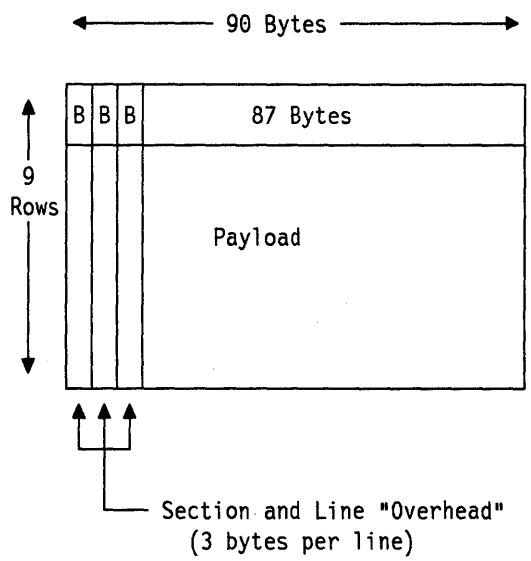


Figure 40. Sonet STS-1 Frame Structure. The diagramatic representation of the frame as a square is done for ease of understanding. The 810 bytes are transmitted row by row starting from the top left of the diagram. One frame is transmitted every 125 microseconds.

This basic frame is called the Synchronous Transport Signal level 1 (STS-1). It is conceptualised as containing 9 rows of 90 columns each as shown in Figure 40.

- The first three columns of every row are used for administration and control of the multiplexing system. They are called "overhead" in the standard but are very necessary for the system's operation.
- The frame is transmitted row by row, from the top left of the frame to the bottom right.
- Of course it is necessary to remember that the representation of the structure as a two-dimensional frame is just a conceptual way of representing a repeating structure. In reality it is just a string of bits with a defined repeating pattern.

The physical frame structure above is similar to every other TDM structure used in the telecommunications industry. The big difference is in how the "payload" is carried. The payload is a frame that "floats" within the physical frame structure. The payload envelope is illustrated in Figure 41 on page 120.

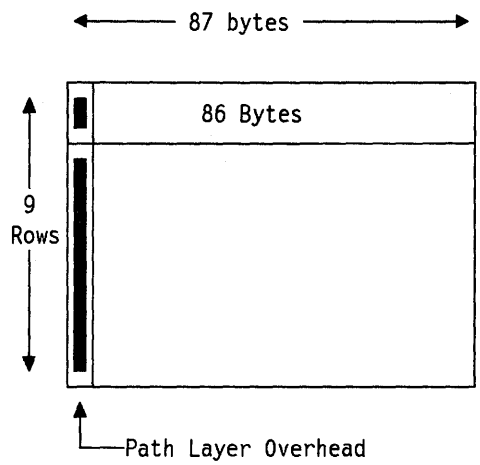


Figure 41. Sonet Synchronous Payload Envelope

Notice that the payload envelope fits exactly within a single Sonet frame.

The payload envelope is allowed to start anywhere within the physical Sonet frame and in that case will span two consecutive physical frames. The start of the payload is pointed to by the H1 and H2 bytes within the line overhead sections.

Very small differences in the clock rates of the frame and the payload can be accommodated by temporarily incrementing or decrementing the pointer (an extra byte if needed is found by using one byte (H3) in the section header). Nevertheless, big differences in clock frequencies cannot be accommodated by this method.

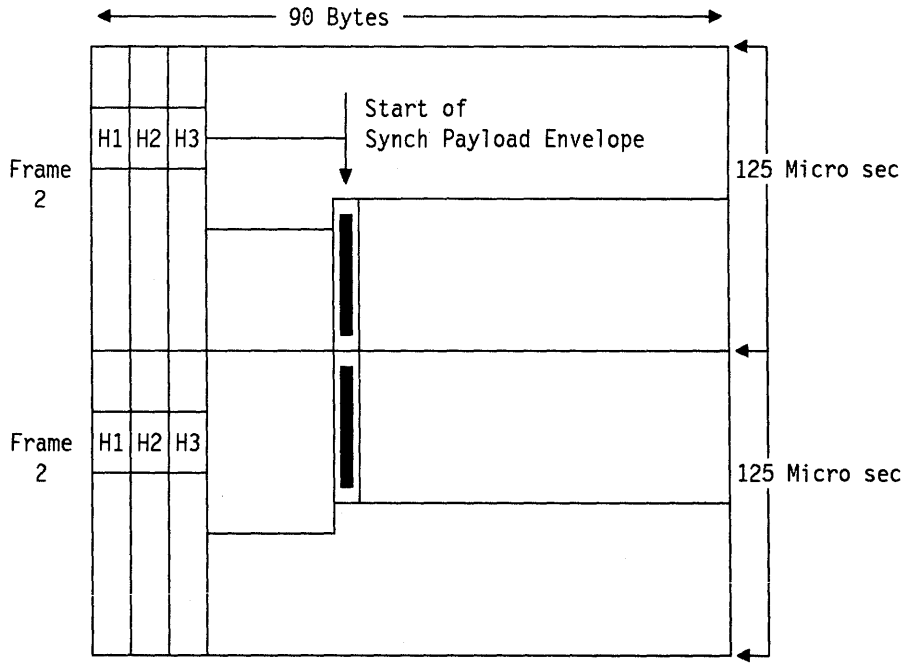


Figure 42. Synchronous Payload Envelope Floating in STS-1 Frame. The SPE is pointed to by the H1 and H2 bytes.

Multiple STS-1 frames can be byte multiplexed together to form higher speed signals. When this is done they are called STS-2, STS-3 etc. where the numeral suffix indicates the number of STS-1 frames that are present (and therefore the line speed). For example STS-3 is 3 times an STS-1 or 155.52 megabits per second. This multiplexing uses the method illustrated in Figure 43.

An alternative method is to phase align the multiple STS frames and their payloads. This means that a larger payload envelope has been created. This is called "concatenation" and is indicated in the name of the signal. For example, when three STS-1s are concatenated such that the frames are phase aligned and there is a single large payload envelope it is called an STS-3c.

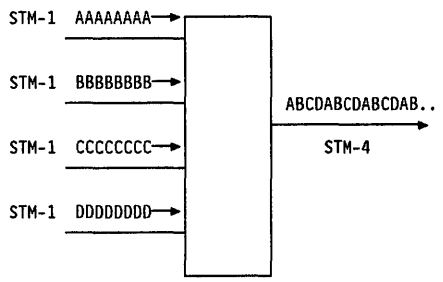


Figure 43. STM-1 to STM-4 Synchronous Multiplexing

6.2.2 SDH

In the rest of the world, Sonet is not immediately useful because the "E3" rate of 35 megabits does not efficiently fit into the 50 megabit Sonet signal. (The comparable US signal, the T3 is roughly 45 megabits and fits nicely.)

The CCITT has defined a worldwide standard called the Synchronous Digital Hierarchy, which accommodates both Sonet and the European line speeds.

This was done by defining a basic frame that is exactly equivalent to (Sonet) STS-3c. This has a new name. It is Synchronous Transport Module level one or STM-1 and has a basic rate (minimum speed) of 155.52 megabits per second. This is shown in Figure 44.

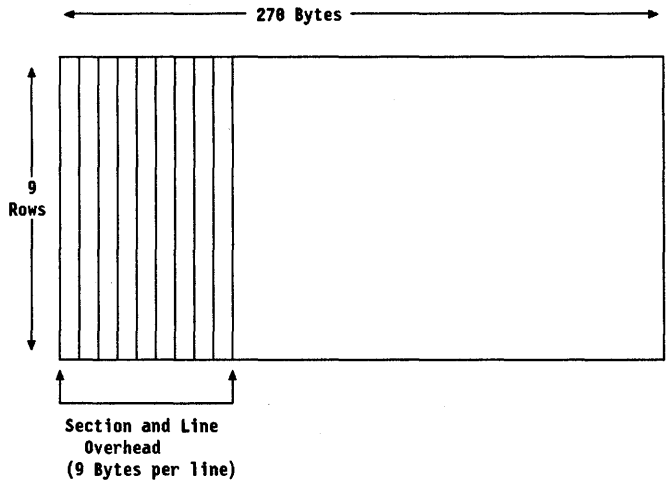


Figure 44. SDH Basic Frame Format

Faster line speeds are obtained in the same way as in Sonet - by byte interleaving of multiple STM-1 frames. For this to take place (as in Sonet) the STM-1 frames must be 125 µsec frame aligned. Four STM-1 frames may be multiplexed to form an STM-4 at 622.08 megabits per second. This (again like Sonet) may carry four separate payloads byte multiplexed together (see Figure 43 on page 121). Alternatively, the payloads may be concatenated (rather than interleaved) and the signal is then called STM-4c.

6.2.3 Tributaries

Within each payload, slower speed channels (called tributaries) may be carried. Tributaries normally occupy a number of consecutive columns within a payload.

A US T1 payload (1.544 megabits/sec) occupies three columns, a European E1 payload (2.048 megabits/sec) occupies four columns. Notice that there is some wasted bandwidth here. A T1 really only requires 24 slots and three columns gives it 27. An E1 requires 32 slots and is given 36. This "wastage" is a very small price to pay for the enormous benefit to be achieved by being able to demultiplex a single tributary stream from within the multiplexed structure without having to demultiplex the whole stream.

The tributaries may be fixed within their virtual containers or they may float, similar to the way a virtual container floats within the physical frame. Pointers within the overhead are used to locate each virtual tributary stream.

6.2.4 Sonet/SDH Line Speeds and Signals

Table 2. Sonet Speed Hierarchy

Signal Level	Bit Rate	DS0s	DS1s	DS3s
STS-1 and OC-1	51.84 Mbps	672	28	1
STS-3 and OC-3	155.52 Mbps	2,016	84	3
STS-12 and OC-12	622.08 Mbps	8,064	336	12
STS-48 and OC-48	2488.32 Mbps	32,256	1344	48

6.2.5 Status

Sonet/DSH standards are now (1991) firm and equipment implementing them is beginning to become available. However, there are many desirable extensions that have not yet been standardised. For example, there is no standard for interfacing customer premises equipment to STS-3c (STM) available as yet. However, it is likely that this will happen in the future since the FDDI standard contains an interface to STS-3c for use as wide area operation of FDDI rings.

6.2.6 Conclusion

Successful specification of a system which integrates and accommodates all of the different line speeds and characteristics of US and European multiplexing hierarchies was a formidable challenge. Sonet/SDH is a complex system but it is also a very significant achievement. It is expected that equipment using SDH will become the dominant form of network multiplexing equipment within a very short time.

6.2.7 The Bandwidth Fragmentation Problem

Existing telecommunication backbone systems are almost exclusively based on TDM structures. The system is cost effective and efficient in an environment where bandwidth allocation is done in a small number of fixed amounts.

In the future, high speed backbone wide area networks owned by the PTTs will need to become a lot more flexible. It is predicted that there will be significant demand for arbitrary amounts of bandwidth and for "variable bandwidth" such as is needed for a video signal. Many planners in the PTTs believe that TDM technology is not sufficiently flexible to satisfy this requirement. This is partly because of the perceived "waste" in using fixed rate services for variable traffic (such as interactive image, variable rate voice or variable rate encoded video) and partly because arbitrary variable amounts of bandwidth are very difficult to allocate in a TDM system. This latter problem is called "bandwidth fragmentation".

The problem of bandwidth fragmentation on a TDM link is exactly the same problem as storage fragmentation in main storage buffer pools within a computer system.

For this example assume we have a 4 Mbps link which consists of 64, 64 Kbps slots represented by the dotted line in Figure 45 on page 124. Each dot represents one 64 Kbps slot.

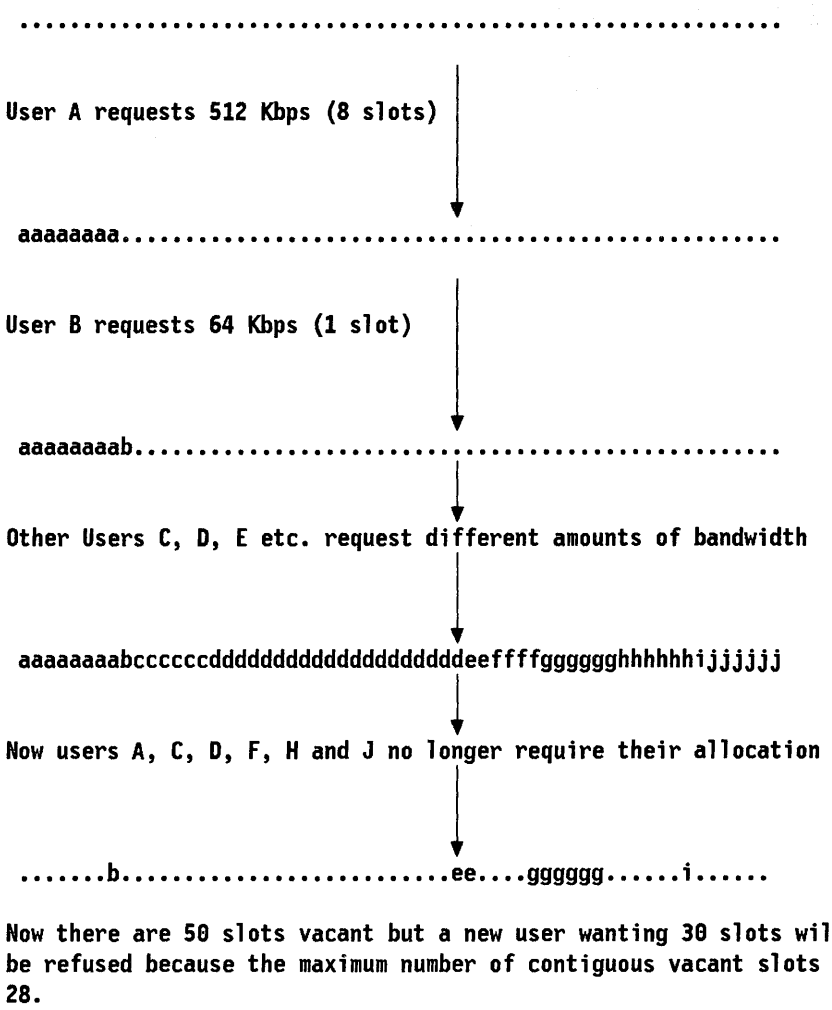


Figure 45. Bandwidth Fragmentation

The above example is trivial but illustrates the point. In computer memory variable length buffer pools (before virtual storage fixed the problem) it was found that after operation for a long period of time perhaps only 20% of the memory would be in use and the other 80% would be broken up into fragments that were too small to use. This is a significant waste of resource. In addition, the control mechanisms needed to operate a scheme such as this would be complex and more expensive than alternative schemes based on cell multiplexing.

A mechanism like the computer's virtual storage could be used to concatenate multiple 64 Kbps channels into wider logical channels, but that requires significant hardware and will only work when communication on the link is peer-to-peer. In the new SDH/Sonet system it is possible for multiple devices to access the same link and extract groups of channels (tributaries) without the need to demultiplex the whole channel. A virtual channel allocation system would take away this ability.

This problem is the primary reason that broadband ISDN will use a cell-based switching system.

6.2.8 Synchronous Transfer Mode (STM)

Synchronous Transfer Mode was a proposed TDM system for implementing broadband ISDN. It was developed for some time by standards committees. It uses the same principles as SDH (and can be thought of as an extension of SDH).

STM was abandoned in favor of the Asynchronous Transfer Mode (ATM) cell switching approach.

Chapter 7. Cell-Based Networking Systems

The concept of cell switching can be thought of as either a high performance form of packet switching or as a form of statistical multiplexing performed on fixed length blocks of data.

A cell is really not too different from a packet. A block of user data is broken up into packets or cells for transmission through the network. But there are significant differences between cell-based networks and packet networks.

1. A cell is fixed in length. In packet networks the packet size is a fixed maximum (for a given connection) but individual packets may always be shorter than the maximum. In a cell-based network cells are a fixed length, no more and no less.
2. Cells tend to be a lot shorter than packets. This is really a compromise over requirements. In the early days of X.25 many of the designers wanted a packet size of 32 bytes so that voice could be handled properly. However, the shorter the packet size the more network overhead there is in sending a given quantity of data over a wide area network. To efficiently handle data, packets should be longer (in X.25 the default packet size supported by all networks is 128 bytes).
3. Cell-based networks do *not* use link level error recoveries. In some networks there is an error checking mechanism that allows the network to throw away cells in error. In others, such as in ATM (described below) only the header field is checked for errors and it is left to a "higher layer" protocol to provide a checking mechanism for the data portion of the cell if needed by the application.

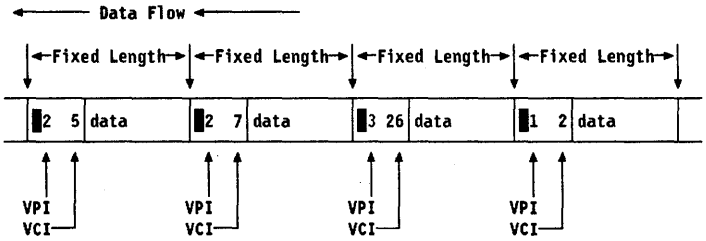


Figure 46. Cell Multiplexing on a Link. Cells belonging to different logical connections (identified by the VPI and VCI) are transmitted one after the other on the link. This is not a new concept in the data switching world but it is quite different to the fixed multiplexing techniques used in the TDM approach.

Figure 46 shows a sequence of cells from different connections being transmitted on a link. This should be contrasted with the TDM (Time Division Multiplexing) technique where capacity is allocated on a time slot basis regardless of whether there is data to send for that connection. Cell-based networks are envisaged as ones that use extremely fast and efficient hardware based switching nodes to give very high throughput, millions of cells per second. These networks are designed to operate over very low error rate very high speed digital (preferably optical) links.

The reasons for using this architecture are:

- If we use very short fixed-length cells then it simplifies (and therefore speeds up) the switching hardware needed in nodal switches.

- The smaller the cells can be made the shorter the transit delay through a network consisting of multiple nodes. This principle is described in section 5.5, "Transporting Data in Packets or Cells" on page 86.
- The statistical principle of large numbers (see Appendix B.1.4, "Practical Systems" on page 287) means that a very uniform network transit delay with low variance can be anticipated with the cell approach.
- Intermediate queues within switching nodes contain only cells of the same length. This reduces the variation in network transit delays due to irregular length data blocks (which take irregular lengths of time to transmit) in the queues.

The reasons that cell switching has not been popular in the past are:

- High error rate analogue links potentially cause too high an error rate to allow end-to-end recovery. In most cases, link level error recovery is needed.
- With the software-based switching technology of the 1970s and 1980s a cell (or packet) takes the same amount of processing time in a switching node *regardless of its length*. Thus the shorter the packet or cell, the greater the cost of the processing in the intermediate switches.
- Hardware technology had not progressed to the point where total hardware switching was economically feasible (it has been technically possible for some years).
- In the older technology end-to-end error recovery processing added a very significant cost to the attaching equipment. (Significantly more storage and instruction cycles required.) This is needed with cell networks today but hardware cost has become much less and this is no longer a problem.
- The presence of link headers at the front of each cell caused a measurable overhead in link capacity. In the days when a 2,400 bps link was considered "fast" this was a significant overhead. In 1992 the cost of link capacity (or bandwidth) is reducing daily and this overhead is no longer considered significant.

The cell technique is intended to provide the efficiencies inherent in packet switching without the drawbacks that this technique has had in the past. Because cells are small and uniform in size, it is thought that a uniform transit delay can be provided through quite a large network and that this can be short enough for high quality voice operation.

7.1 Asynchronous Transfer Mode (ATM)

ATM is a protocol for user access to and (to some extent) the internal operation of a public high speed cell switching system. It is the technical protocol which has been accepted as the basis for the broadband ISDN service.

Broadband ISDN will use ATM cell-based multiplexing rather than time division multiplexing for a number of reasons:

Bandwidth Efficiency

Although bandwidth is becoming very low in cost, some services are bursty in nature and use a very large bandwidth (for example digital High Definition TV). These services, because of their bursty nature, require a peak bit rate many times higher than the average bit rate. Packet or cell switching (multiplexing) potentially provides a large saving (at least 4 to 1 improvement) in bandwidth.

For a large number of voice channels taken together just using packetisation techniques without compression, the bandwidth used by packet voice is about 40% of what would be required for TDM multiplexing systems.

Equipment Cost

Many people believe that a large "trunk" telephone exchange using high speed packet technology is likely to cost some 30% less than a comparable exchange using TDM techniques.

The argument goes that TDM exchanges (PBXes etc.) require control wiring and data path wiring to be separate. A packet (cell) switch sends its control packets through the same paths used for data.

Service Suitability

ATM technology is suitable for all types of traffic. These are:

- Voice (both variable and constant rate)
- Traditional data
- Image
- Video (variable rate)

Dynamic Allocation of TDM Capacity

The B_ISDN system needs to provide arbitrary amounts of bandwidth allocated on demand. There is a technical problem in doing this in a TDM system. This problem is described in section 6.2.7, "The Bandwidth Fragmentation Problem" on page 123.

7.1.1 ATM Concept

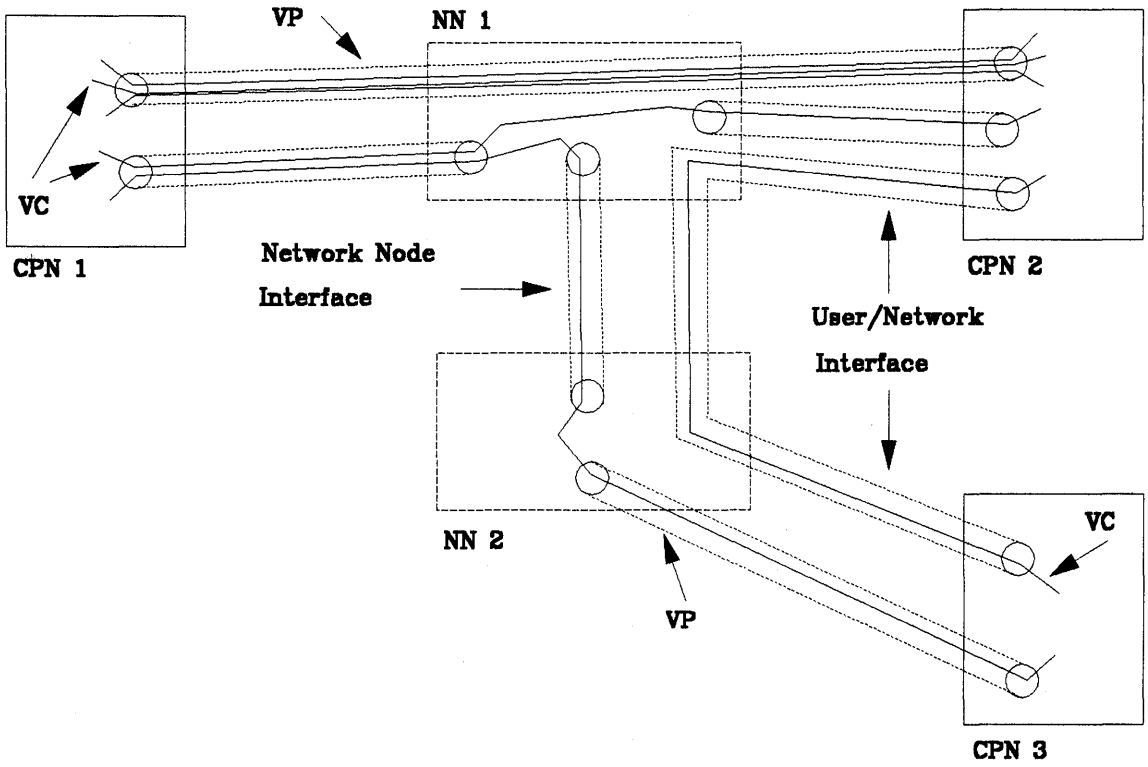


Figure 47. Routing Concept in an ATM Network. Physical links between nodes are not shown.

The conceptual structure of an ATM network is shown in Figure 47, above.

Network Nodes (NNs)

Two network nodes (called NN1 and NN2) are shown in the figure. These perform the backbone data transport within the ATM network.

Customer Premises Nodes (CPNs)

This is the ATM name for end user equipment. For some services in some countries this equipment may belong to the PTT. In the US and in most countries this equipment will be purchased by the end user from suppliers unrelated to the PTT.

The User Network Interface (UNI)

The UNI is specified exactly by the applicable standards. It is structured according to the reference model for ISDN illustrated in Figure 34 on page 104.

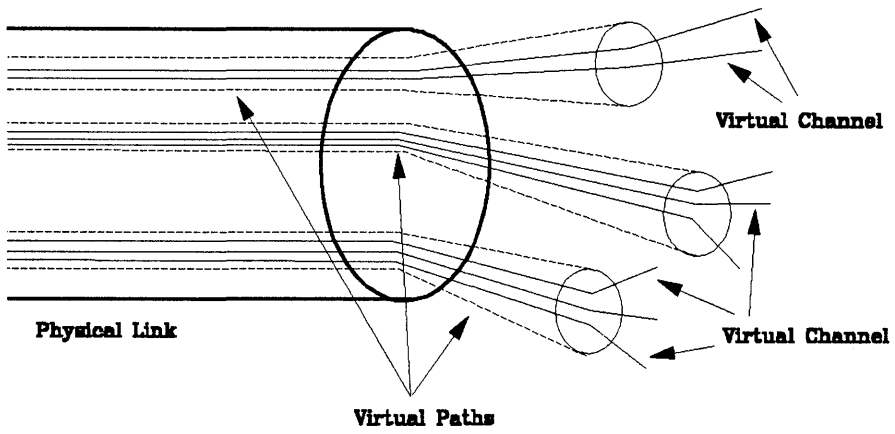


Figure 48. Link, Virtual Path and Virtual Circuit Relationship

Links

There may be one or many physical link connections between the nodes but these are not shown in Figure 47 on page 130 for clarity. The multiplexing of virtual paths and virtual circuits over a physical link connection is shown in Figure 48.

Links between nodes may be carried as "clear channels" such as over a direct point-to-point connection but may also be carried over an SDH/Sonet connection.

Virtual Path (VP)

A VP is a route through the network. VPs may exist:

1. Between CPNs (as between CPN1 and CPN2 and between CPN2 and CPN3 in the figure)
2. Between NNs and CPNs (as between CPN1 and NN1, CPN2 and NN1 and CPN3 and NN2 in the figure) and
3. Between NNs (as between NN1 and NN2 in the figure)

A VP may be routed through an NN by reference only to the VP number or it may terminate in an NN. A VP entering a CPN always terminates in that CPN. (A CPN, by definition, cannot perform routing on VPs or VCs - or rather, it can but whatever it does in this regard is not defined by ATM.)

Virtual Connection (VC)

A virtual connection is the end-to-end connection along which a user sends data. The concept is very close to that of a virtual circuit in X.25. (See Appendix D, "An Introduction to X.25 Concepts" on page 293.) The major difference is that a virtual connection carries data in one direction only, whereas a virtual circuit is bidirectional.

Virtual Channel

A virtual channel is a separately identified data flow on a link. A virtual connection through the network is a sequence of interconnected virtual channels.

7.1.2 Cell Format

The cell format is illustrated in Figure 49.

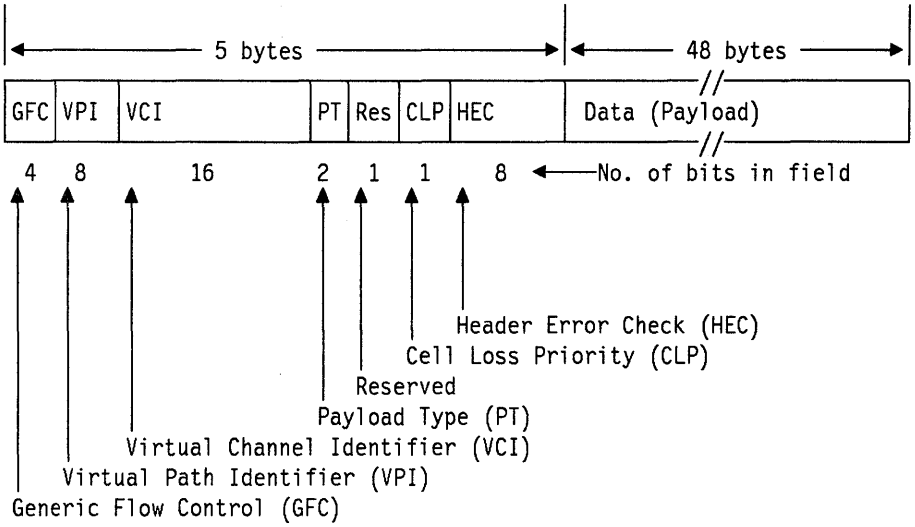


Figure 49. ATM Cell Format at User Network Interface (UNI)

The most important fields in the cell header are the Virtual Path Identifier (VPI) and the Virtual Channel Identifier (VCI). Together these identify the connection (called a virtual connection) that this cell belongs to. There is no "destination network address" because this would be too much of an overhead for a 48-byte cell to carry.

The VPI/VCI together are similar in function to the Logical Channel Identifier in X.25 or the DLCI of Frame Relay in that they do not identify the destination address of the cell explicitly but rather they identify a connection which leads to the desired destination. Thus:

- An ATM system is basically a connection oriented system (see section 5.6, "Connection Oriented versus Connectionless Networks" on page 88). The system uses either call-by-call (switched circuit) set-up or semi-permanent connections.

  There is a connectionless mode of operation defined. In this mode of operation the first cell of a group carries the full network address of the destination within its data (payload) field. Subsequent cells belonging to the same user data block do not carry a full network address but rather are related to the first cell by having the same VPI/VCI as it had. One way of looking at this is as a very short term connection. DQDB works in a similar way in that when a frame is sent as a group of related cells, only the first cell carries the full network header and subsequent cells carry an identifier. This identifier is used by the receiving node to identify cells that are part of the same frame of data. (See section 9.5.7, "Data Segmentation" on page 209).

- Internal network operation uses the principle of logical ID swapping described in section 5.7.3, "Logical ID Swapping" on page 93.

- The cell size of 48 bytes was determined by the CCITT as a compromise between voice and data requirements. See the discussion in section 5.2.2, "Transit Delay" on page 81.

7.1.3 Virtual Connection (Virtual Channel Connection)

The virtual channel is defined as "a logical association between the endpoints of a link that enables the unidirectional transfer of cells over that link". It is important to note that this definition only has scope within an individual link.

The end-to-end connection over which a user sends data in ATM is called a virtual connection (or sometimes a virtual channel connection). This is defined as "A concatenation of virtual channels to provide transfer of cells between one or more points in a network and one or more other points in that network".

Virtual connections may be "semi-permanent" (set up by administrative procedure with the network administration) or they may be set up on demand from the CPN. Virtual connections that are set up on demand are called "Switched Virtual Connections".

The VC is carried within a VP. When an NN routes a VC towards its destination this may be done for the individual VC (if the VP terminates in this NN). Alternatively, the VP itself may be routed in which case the NN will not know about the VCs within it.

In Figure 47 on page 130 there is a VP between CPN1 and CPN2. In this case NN1 switches cells based on the VP identity but neither knows nor cares about the VCs within that VP.

In the same figure there is a VC between CPN1 and CPN3 which transits both NN1 and NN2. This VC uses three different VPs along its route. In this case each NN must know about both the VPs involved and the VCs.

7.1.3.1 Guaranteed Sequential Delivery

Cells delivered to the network by a CPN over a virtual connection are transferred to the partner CPN in the same sequence as they were presented to the network. This is very important as it means that the end user (or the adaptation layer function) does not have to resequence cells that arrive out of order. But it also restricts the network to using a single path for any given virtual connection.

The payload (data) part of a cell may contain errors. Transmission errors within the data portion of the cell are *not* detected by the network (this is up to either the end user equipment or the adaptation layer).

The network does, however, provide some protection against the misrouting of cells. The cell header contains the HEC (Header Error Check) field. This field allows the *correction* of all single bit errors in the header part of the cell *or* for the *detection* of most single and multi-bit errors. The "or" in the previous sentence is critical. When the algorithm determines that there is an error it has no way of determining whether that error is a (correctable) single bit error or an (uncorrectable) multi-bit error.

What to do in this situation requires further study. If the overwhelming majority of errors are single bit then it seems the best thing to do is to correct them and to tolerate the fact that some mis-routing will occur on cells that really had multi-bit errors. Or you could play it safe and discard all cells where the algorithm says there is an error.

7.1.3.2 Quality of Service Characteristics (QOS)

Each virtual channel connection has a given quality of service characteristic associated with it. This QOS specifies a guaranteed minimum available bandwidth as well as a maximum peak (instantaneous) allowed bandwidth. In serious overload situations, when the network cannot recover from overload by discarding only cells marked as low priority, the network will select which cells to discard depending on the QOS characteristic on the VC.

A VP also has a QOS associated with it. VCs within a VP may have a lower QOS than the VP but they cannot have a higher one.

7.1.3.3 Cell Discard and Loss

Cells may be lost or discarded by the network. The network does *not* detect the loss of cells and does *not* signal the end user when it has discarded cells from a particular connection.

Some variable bit rate encoding schemes for voice and for video are structured in such a way that two kinds of cells are produced:

- Essential cells which contain basic information to enable the continued function of the service and
- Optional cells which contain information that improves the quality of the service (voice or picture quality)

If the end user equipment marks a cell as low priority that cell will be discarded first if cells need to be discarded due to network congestion.

If discarding marked cells is insufficient to relieve the network congestion, then the network may use the QOS to decide which cells to discard next.

Some services (such as bank cash dispenser operation) are very important to the end user and must continue to operate at all costs. These services would have a high quality of service, and cells belonging to these virtual connections would not be discarded. On the other hand many voice conversations (or batch data transfer) can be interrupted and restarted later with very little impact on anyone. In this case a lower quality of service may be specified which would allow the network to discard cells belonging to these virtual connections in situations of extreme congestion.

7.1.4 Physical Transport

Although ATM is designed to be independent of the physical transport medium, it has been specified to operate over either SDH/Sonet or a "clear channel". Operation over Sonet connections is likely to predominate in the US where operation over clear channels will be the European approach.

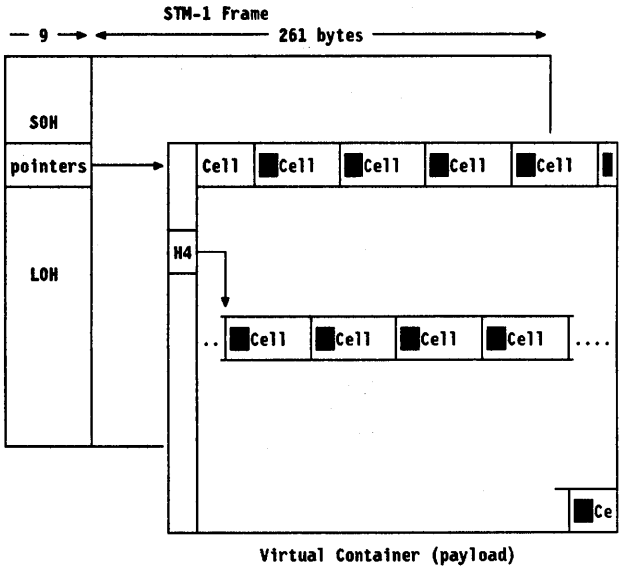


Figure 50. ATM Cells Carried within an STH Frame

Figure 50 shows how ATM cells are transported within a Sonet/SDH frame. Notice that within the virtual container, cells are concatenated row by row without regard to row or cell boundaries.

Operation is defined at:

- 155.52 megabits/sec using STS-3c (STM-1)
- 622.08 megabits/sec using STS-12c (STM-4)
- Either of the above speeds over a "clear channel"

The figure shows the frame structure for operation at 155.52 megabits/sec in an STS-3c frame. In the overhead section of the virtual container there is a pointer (H4) to a cell boundary so that the system can decide where the stream of cells begins. Cells span frames without any gaps.

Over a clear channel (and over STM-4) synchronisation is different. When the receiver is unsynchronised, it looks constantly at the stream of incoming bytes to see if a cell header can be found. It does this by assuming that every string of five bytes (overlapping byte by byte) is a valid cell header and checking the HCS field. If the HCS field checks correctly then this *may* be a cell boundary. It will then assume that the next cell boundary is 53 bytes from the tentative one that it just found and check that cell header. The process continues until the receiver has found a number (to be defined) of valid consecutive cells. When this happens the receiver has achieved cell synchronisation.

The important point to note here is that cell synchronisation is obtained without any additional markers in the data. There is no code violation or synch character to mark the beginning of cells. Thus the process is very efficient.

7.1.5 User-Network Interface (UNI)

The User Network Interface is a set of definitions that specify the interface between customer equipment and the equipment provided by the PTT. These definitions are structured according to the ISDN reference model (see Figure 34 on page 104).

Physically there is a link to the network. The link capacity is shared by the use of the VPI and VCI fields of the cell header.

7.1.5.1 The D Channel

There is always at least one VP and one VC from the CPN to the NN to which it is directly connected. This forms a virtual connection between the CPN and the NN. This VC is used to pass control and maintenance information such as requests to set up a new connection etc. This is really just another ISDN D channel but it will be structured rather differently to the D channel of regular narrowband ISDN because of the cell-based environment.

7.1.6 Network Node Interface (NNI)

The NNI is a superset of the UNI. In addition to having all of the data transfer functions of the UNI the NNI contains network signaling procedures for operation and control of the total network. It however, does not contain the end user signaling protocols as these terminate in the NN nearest to the CPN.

This is a very new concept in the standards world. Today, there exists *no* standard from any standards organisation that specifies the *internal operation* of a packet network. X.25 specifies only the end user interface and the services to be provided. B_ISDN will specify the network internal protocol so as to allow worldwide compatibility and interworking.

7.1.7 Internal Network Operation

7.1.7.1 Logical ID Swapping

ATM networks will operate internally by logical ID swapping. This was described in concept in section 5.7.3, "Logical ID Swapping" on page 93. Logical ID swapping is relatively fast in operation and it provides a single route through the network for each virtual connection. This is important because the network is constrained to deliver cells on a virtual connection in the same order in which they were presented to the network. (Of course different virtual connections between the same pair of CPNs may go by different routes.)

In ATM there are two IDs for each virtual connection: The VPI and the VCI. Some ATM network nodes (NNs) may only know about and switch VPs. Other nodes will know about and switch both VPs and VCs.

An NN must keep a table of VPIs relating to each physical link that it has attached. This table contains a pointer to the outbound link where arriving data must be routed. If the VP terminates in the particular NN, then the NN must keep a table of VCs for each terminating VP. This table contains pointers for the further routing of the data. The VC may be a D channel and terminate in this particular NN. Alternatively, the VC may be logically connected to another VC through this NN in which case the NN must route the data to the appropriate outbound connection (identified by the link, VPI and VCI).

When the NN routes the cell onward using only the VP then the VPI number is changed. (VPIs only have meaning within the context of an individual link.) When the NN routes the cell by using both the VPI and the VCI then the outgoing cell will have a different VPI and VCI.

The switching part of this process can be relatively efficient in that it is possible to build a total hardware implementation that will route the data correctly. The problem is that if the switching tables require frequent updating (such as the set up of new connections and the termination of old ones) then frequent access to the tables may be required from a programmed processor. The need for shared memory access can impair the throughput of the hardware switching function.

7.1.7.2 Flow Control

There is no flow control in an ATM network of the kind that is standard in traditional packet switching networks (such as SNA). When a connection is requested, the parameters of that connection (service class and requested throughput rate) are examined and the connection is allowed only if the network has sufficient capacity to support the new connection.

This is a lot easier said than done. Any system that allocates capacity in excess or real capacity on the basis of statistical parameters allows the possibility (however remote) that by chance a situation will arise when demands on the network exceed the network resources. In this case, the network will discard cells. The first to be discarded will be cells marked as lower priority (CLP bit). After that discarding will take place by service class.

At the entry point to the network, the NN monitors the rate of data arrival for a VP or a VC according to the declared service class and will take action to prevent a CPN from exceeding the allowed limits of the service class in use. The CPN is expected to operate a "leaky bucket" rate control scheme (see section 5.1.1.1, "Leaky Bucket Rate Control" on page 77) to prevent it from making demands on the network in excess of its allowed capacity.

7.1.7.3 End-to-End Data Integrity

There is no end-to-end data integrity provided by the ATM network itself. This is the responsibility of the end user equipment. The "adaptation layer" function (implemented in the CPN) provides this function.

7.1.8 ATM Adaptation (Interfacing) Layer (AAL)

In order to make an ATM network practical it is necessary to adapt the internal network characteristics to those of the various traffic types that will use the network. This is the purpose of the adaptation layer. It would have been possible to leave this function to end user equipment suppliers but that could mean that many different systems of voice or video coding (all incompatible with one another) would come into use. The function of the adaptation layer is to provide generalised interworking across the ATM network.

One way of looking at the adaptation layer is to regard it as a special form of Terminal Adaptor (TA) within the ISDN model.

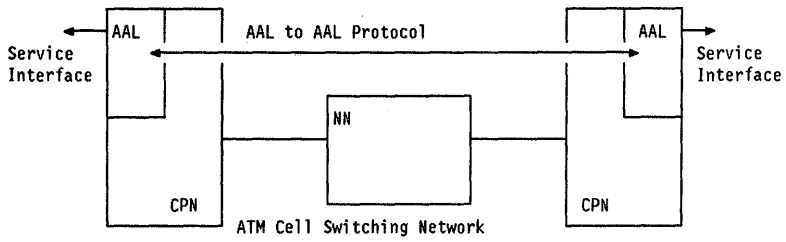


Figure 51. The ATM Adaptation Layer

7.1.8.1 AAL Service Classes

The ATM Adaptation Layer (AAL) has four different classes of service which correspond to the four generic classes of network traffic.

	Class 1	Class 2	Class 3	Class 4
Timing Between Source and Destination	Related		Not Related	
Bit Rate	Constant	Variable		
Connection Mode	Connection Oriented			Connection-less

Figure 52. Characteristics of Service Classes in the ATM Adaptation Layer

The AAL function operates an end-to-end protocol across the ATM network to provide support for end users of the four identified service classes:

Class One (Circuit Emulation)

This service emulates a leased line. It is intended for constant rate voice and video applications etc.

This service requires the following characteristics:

- There is a constant bit rate at source and destination.
- There is a timing relationship between source and destination.
- There is a connection between end users of the service.

The adaptation layer performs the following functions in order to support this service:

- Segmentation and reassembly of data frames into cells.
- Handling (buffering) of cell delay variations.
- Detection and handling of lost (or discarded) cells.
- Recovery of the source clock frequency (in a plesiochronous[41] way).
- Detection of bit errors in the user information field.

[41] See Appendix C, "Getting the Language into Synch" on page 291

Class Two (Variable Bit Rate Services)

This is intended for voice and video traffic that is basically isochronous at the level of end user presentation, but which may be coded as variable rate information.

These services have a variable flow of information, need some timing relationship between the ends of the connection and are connection oriented.

The services provided by the AAL for class 2 are:

- Transfer of variable bit rate information between end users.
- Transfer of timing between source and destination.
- Indication of lost or corrupted information not recovered by the AAL itself.

Class Three (Connection Oriented Data)

This is traditional data traffic as known in an SNA or X.25 network.

These services are connection oriented and have a variable flow of information.

Two services are provided called "Message Mode" and "Streaming Mode". Message mode provides for the transfer of single frames of user information. Streaming mode provides transport for multiple fixed length frames of user data.

The AAL for class three provides:

- Segmentation and reassembly of frames into cells.
- Detection and signaling of errors in the data.
- The multiplexing and demultiplexing of multiple end user connections onto a single ATM network connection.

Class Four (Connectionless Data)

This service has several uses in sending ad-hoc data but could be used for example, to carry TCP/IP or LAN interconnection traffic where the protocol in use is inherently connectionless. (See section 5.6, "Connection Oriented versus Connectionless Networks" on page 88.)

These services are connectionless and have a variable flow of information. It is intended to support connectionless networking protocols such as TCP/IP and services that transfer data character by character (such as is the case with an "ASCII TWX" terminal).

Like the other three types of service, class four requires:

- Segmentation and reassembly of frames into cells.
- Detection of errors in the data (but not retransmission).

In addition to these basic services class four gives:

- Multiplexing and demultiplexing of multiple end user data flows onto a single cross-network data flow (for single character transmissions).
- Network layer addressing and routing.

In order to do this the AAL has a number of end-to-end protocols across the ATM network. In addition the AAL builds its own headers on the data to carry

information (such as sequence numbers) essential to the performance of the AAL function.

It can be seen that the functions provided by the AAL are very basic ones. They are similar to those provided by a LAN at the MAC layer. Figure 58 on page 155 shows the interface between the LAN MAC layer and the logical link control layer. This is the level of function provided by the AAL (albeit that the AAL handles many types of service in addition to data). For data applications a logical link layer (such as IEEE 802.2) will still be needed for successful operation.

7.1.9 Status

ATM is a fast developing standards proposal. It is defined by the CCITT Study Group XVIII. In 1992 there is no working commercial ATM system available anywhere in the world. Prototype systems are beginning to appear in early tests.

Nevertheless, work on the standards is progressing very rapidly. The basic set of standards are scheduled for final vote by the CCITT technical committee (SG XVIII) in mid 1992. Nevertheless, as of January 1992 none of the standards are firm yet and many things could change before they are accepted. This would see early ATM systems going into commercial service as early as 1996.

The major obstacle to be overcome is that as yet, there is no experience in operating cell-based switching systems for voice or video traffic. Over the years network engineers have developed an excellent body of knowledge about the behavior of circuit switched systems carrying voice and the design parameters are very well known. (Of course there is a lot of knowledge available about packet switching systems for data traffic and some of this knowledge has application.)

The (billion dollar) question is "what is the statistical behavior of variable rate packetised voice and video systems when operated on a large scale". Many people (perhaps the majority) believe that it will all add up statistically very nicely and stable operation of these systems will be possible with link and node utilisations as high as 80% or more. Some other people say no. They believe that the variation in the aggregate traffic load due to statistical variance could be so great as to prevent network operation at resource utilisations much above 20%. (Some speakers expressed this view at the CCITT Study Group XVIII meeting in Melbourne in December 1991.)

If network utilisation can approach 80% then the economic viability of ATM is difficult to question. At loadings of 20% the economics are problematic. Time will tell.

7.2 Broadband ISDN

All of the discussion in the previous section was about broadband ISDN as much as it was about ATM. ATM is the technical structure and protocols for the service called broadband ISDN. The B_ISDN service is designed as far as possible to be independent of the implementing technology.

It is hard to overestimate the importance of broadband ISDN. Broadband ISDN is proposed to be the *replacement* for all of the telephone exchanges in the world!

Of course, such replacement takes a long time and enormous levels of investment. The concept will need to be proven in the field, the standards will need to be stable and properly debugged and the benefits thoroughly proven before the required level of investment will be forthcoming.

Nevertheless, it is very widely believed in the telecommunications industry that broadband ISDN will become the worldwide telecommunications network of the next century.

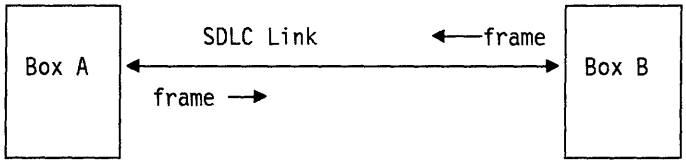
Chapter 8. High Speed Packet Networking

8.1 Frame Switching

Frame switching describes a very common generic network interfacing technique. Many "X.25 networks" on the market offer an "SDLC PAD" function which is capable of switching data from an SDLC link in one part of the network to another SDLC link somewhere else in the network.[42] The term "Frame Switching" does not describe any standardised protocol or interface at all.

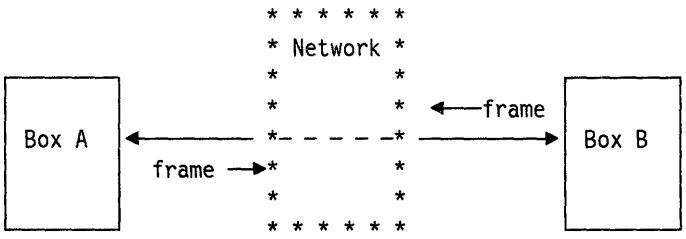
The frame switching technique is not new. Nor is it a "high speed" technology. Understanding frame switching is, however, important to understanding the genesis of Frame Relay.

Consider two machines communicating over a point-to-point SDLC[43] link.



In the diagram when Box A needs to send something to Box B header and trailer information specific to the link control protocol is appended to the ends of the block to be sent. This block of data is called a "frame".

In a network situation, inside the frame there will typically be a network header on the front of the user data.



If we want to place a network between Boxes A and B as illustrated above, then there are many alternatives. Typically in the front of the user data message there will be a set of network protocol headers. We could:

1. Terminate the link control protocol (from Box A's perspective make the network "look like" box B) and then use the network protocol headers to

[42] The IBM X.25Net product can perform this function.

[43] In principle, the link protocol used doesn't matter at all. Frame switching interfaces exist for many different types of link control such as BSC and LAPB as well as SDLC.

route the data to its destination. The problem with this is that the network needs to understand the network headers in the data. This is often a complex problem since there are many different network protocols in existence.

2. Take whatever is sent by Box A, put our own network headers onto the front, route it through the intermediate network and then send it on to Box B.

   This is very simple to do and satisfies many requirements but:

   - Link control protocols have very short time-outs, which can be lengthened but are a problem if the network delay is irregular.
   - Many link controls (especially SDLC) rely for their operation on regular polling. Sending polls across the network (and receiving the responses) can cause significant additional network overhead.
   - The network addresses of the communicating boxes must be prespecified to the network and must be fixed. The address in the link control header[44] can be used to identify different destinations, but this ability is very limited.

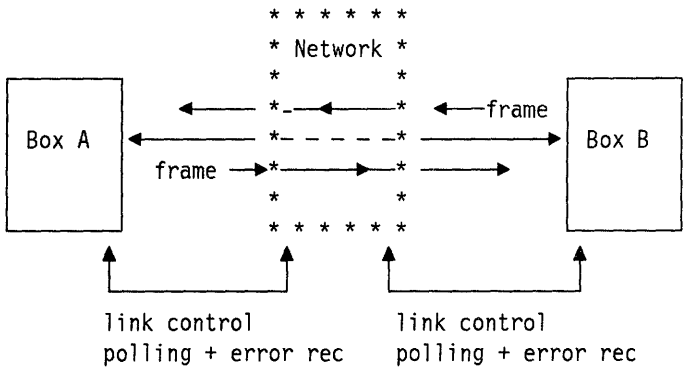


Figure 53. Frame Switching. The network viewed from Box A "looks like" Box B but polling and error recovery are handled locally and do not pass through the network.

Frame switching is like alternative 2 above, but the polling function is handled locally at each end of the network. Only data and control messages are forwarded through the network. The details of how this works vary from one implementation to another. A number of characteristics should be noted:

- The link control from the user to the network is terminated in the nearest network node. This means that link time-outs are no longer a problem. Since SNA has no time-outs in its network protocols (outside the physica and link layers), SNA systems will not have any problem with time-outs. Some other protocols do rely on time-outs and can experience problems in this situation.
- Many implementations of this kind allow the end user boxes to initialise their communications at different times. This precludes those boxes from exchanging their characteristics using XID (eXchange IDentification) protocols. It is this characteristic which prevents many networks from correctly handling SNA PU 2.1 nodes. These nodes rely on XID type 3 to tell each other their characteristics and will not function correctly unless the

[44] in SDLC one can address up to 255 secondary "boxes" but the control end of the link has no address - it is specified by context.

frame switching network understands the XID protocols and properly synchronises the initialisation.

- Since the line control is terminated by the nearest network node when a frame is sent to the network it is acknowledged immediately on receipt. If that frame is subsequently lost by the network (through a node failure, for example) the end "boxes" will "think" that the frame has been received by the destination when it really has not.[45] This is no problem for SNA (or OSI) since link control is not relied upon for network level acknowledgements (that's what layered architecture is all about). However, it is disruptive since if a frame is lost there is no recovery and connections (sessions) running above the link layer must be terminated and restarted.

  Some older devices (predominantly BSC) actually use line control level acknowledgements to signal correct receipt of the data to the user application. (There are BSC versions of frame switching available in the marketplace.) These older devices will not function correctly over a frame switching system because data can be lost without being detected.

- There is another problem with link level switching systems (this also applies to LAN bridges and routers). The network loses the ability to distinguish between different types of traffic based on priority.

  For example, an SDLC link may carry many SNA sessions all at different priorities. At any point in time the link may be carrying highly critical interactive traffic or batch traffic (or a mixture). The network has no way of knowing so it must treat all traffic as a single priority.

  The consequence of this is that the network is unable to load its intermediate links or nodes to utilisations much above 60% at the peak (with average utilisations of perhaps 20% to 30%). In contrast, if the network is able to distinguish between data flows based on priority, then resources can be utilised up to 60% (peak) for interactive traffic only and the remaining capacity may be used for non-critical batch traffic. (SNA networks operate this way.)

  The "net" of the above is that networks that switch frames are much less efficient in their use of link bandwidth (and node cycles) than networks (like SNA) that are able to recognise priorities.

---

[45] Contrast this with the later description of Frame Relay. In Frame Relay the link control operates across the network and thus can be used to recover from network losses of data.

8.2 Frame Relay

Frame Relay is a standardised technique for **interfacing** to a packet network.

Frame Relay originated as an optional service of ISDN. Users send frames to a network node over an ISDN B, H, or D channel and these frames are passed to another user in some other part of the network (also through a B, D, or H channel). Frame Relay has, however, been implemented in few (if any) public ISDN networks. Rather, Frame Relay has become very important as a network interfacing technique quite independently of ISDN.

Frame Relay is an interface definition. While it is possible for networks to use Frame Relay techniques internally for transmission between network nodes there will be many networks that do not work this way. A Frame Relay network is a network that allows a user to attach through a Frame Relay interface and which provides the services and facilities necessary to support communication between FR interfaces.

An interim technology?

While Frame Relay should not be regarded as a true high speed technology it is extremely important because:

1. It can be implemented fairly easily on existing packet switching equipment.
2. It can provide an immediate throughput improvement of between ten and thirty to one over previous technologies *using existing equipment*.

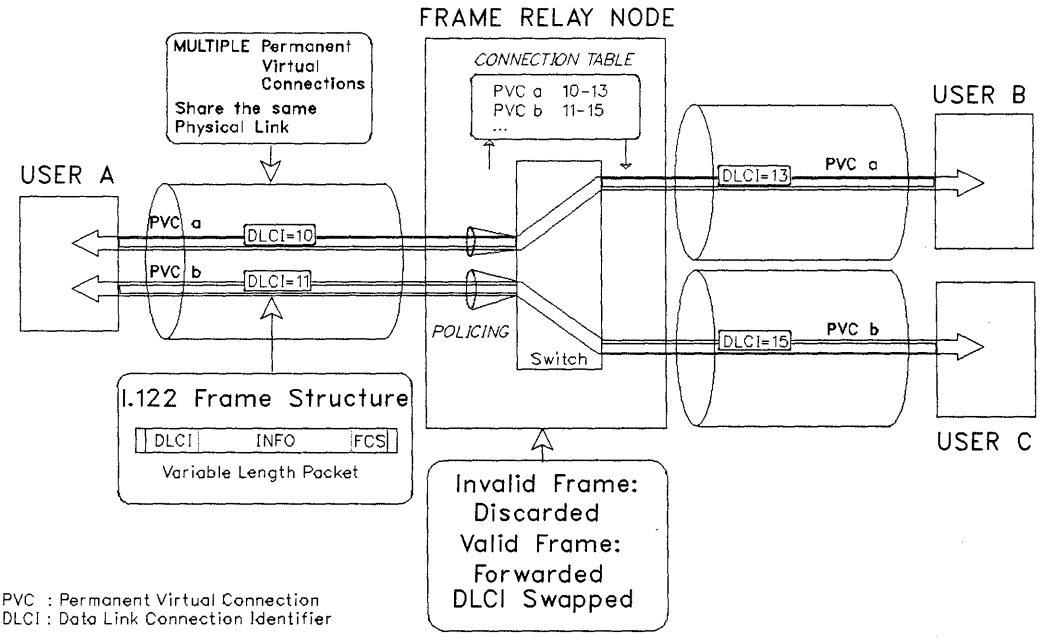


Figure 54. Frame Relay Principle

8.2.1 Concept of Frame Relay

Frame Relay is an exceedingly simple concept. Communicating devices interact with one another across the network transparently. That is, a network is interposed between devices communicating on a link but the devices are not aware that this has happened. In practice, it is not quite as simple as the above suggests because a special link control protocol is used but the principle still holds.

8.2.2 Basic Principles

The basic principles of Frame Relay are as follows:

A virtual link (connection) exists across the network.

This connection consists of pairing a local address (called a Data Link Connection Identifier, DLCI) on one port (link) with a local address on another port somewhere else in the network. This is *exactly* the same as the virtual circuit concept in X.25 where a VC is a pairing of logical channels on different ports across the network. See Figure 126 on page 295.

The Frame Relay local address is just the address field in the data link frame. In Frame Relay this is called the Data Link Connection Identifier (DLCI).

Figure 54 on page 146 illustrates an FR connection within a single node. User A has connections with both User B and User C. The connection between User A and User B is called a Permanent Virtual Connection (PVC).

When User A communicates with User B the link address field (DLCI) as seen by User A is "10". When User B communicates with User A it sees the connection as having DLCI 13. The network, in this case the node, has a connection table which relates DLCIs on one interface with DLCIs on another and so constructs a table of virtual connections.

> *In reality, each table entry must consist of <node id, link id, DLCI> because the DLCI field has meaning only in the context of a particular link. For example in Figure 56 on page 151 DLCI 50 is used by both Terminal Equipment (TE) A and TE C but there is no relationship between the two DLCIs.*

Data delivered to the network with a given DLCI on a particular port is transported across the network and delivered to the port appropriate to this virtual connection. When the data is forwarded by the network, the DLCI is swapped. The new DLCI represents the connection on the destination port.

Many virtual connections may share a single physical port.

This is the central characteristic of a separate network. It is true of X.25 style networks as well. The end user "saves" the cost of all the ports that would be needed if point-to-point lines were installed.

Every non-error frame is relayed.

This means POLLs and acknowledgements (if any) are relayed. The network does not know the difference between a POLL and a data frame. It is all just data to the FR network.

Error frames are discarded.

The FR network makes no attempt at error recovery of incorrect frames. Error recovery is the responsibility of the link control protocol.

Frames may be of any length.

Up to a network imposed maximum, frames may be of any length. (A minimum network capability of 262 bytes is mentioned in the standard.) Most of the announced FR networks (in January 1992) have a maximum frame size of 2 KB. The standard allows frame lengths of up to 8 KB and it is expected that networks handling frames of this size will become available in the future.

There is no "packetisation" performed on transported frames.

Some networks may packetise internally (though these will probably be a minority).

FR networks avoid the irregular time delays and consequent time-out problems caused by the presence of relatively long frames by using fast link connections between nodes. (See the discussion in appendix B.1.4, "Practical Systems" on page 287.)

Link control operates from device to device across the FR network.

Although the FR specification was originally related to the ISDN LAPD specification, there is no constraint on the logic of whatever link control is used. The link control must, however, use the HDLC frame format and allow for the two address bytes of FR at the start of the frame.

- The frame format is identical to that of SDLC/HDLC *except* that two address bytes are used. See Figure 55 on page 150.
- This frame format is the format used by the link control called "LAPD" on the D channel of narrowband ISDN.
- The generic frame format of the HDLC series of link controls consists of FAC (Flag followed by Address followed by Control) then data, then a FCS (Frame Check Sequence) field and another Flag.

  In Frame Relay there are no control bytes. That is, the network does not look at the link control protocol at all.

  The beginning flag signals the start of a frame and also the start of accumulation of the Frame Check Sequence.
- Some product implementations will undoubtedly use the LAPD protocol because it is a point-to-point protocol and does not involve polling. Normal SDLC and LAPB are more difficult because they only use a single address byte and therefore a code change is required to use FR. Also, SDLC uses polling and this adds unnecessary network traffic.

  IBM SNA products will use IEEE 802.2 protocol instead of LAPD (see section 8.2.7, "SNA Connections Using Frame Relay" on page 154). IEEE 802.2 is a link control protocol designed to operate in a LAN environment in many ways similar to the FR environment.

A Local Management Interface (LMI) to access the network's management functions.

There is a reserved link address (DLCI) which allows for communication between the attaching device and the network. This provides a

mechanism for communicating the status of connections (PVCs). Initially there are three functions available:

- A query command that allows the DTE to ask the network if it is still active. This is called a "heartbeat" or a "keep alive" message.
- A query to ask the network for a list of valid DLCIs defined for this interface.
- A query to determine the status (congested or not) of each DLCI.

In the future this may be used for more extensive network management information and for setting up virtual link connections dynamically (switched virtual circuits). Attached devices are not required to use the control channel. Of course, if the LMI is not used the attached device cannot establish switched virtual circuits.

The LMI is used to set up Switched Virtual Circuits.

There is a method of setting up switched virtual circuits in the FR standard. This involves sending a call request to the network using the Local Management Interface (LMI). Notice that this is an "out of band signaling" situation. The request to set up a connection is sent on a different DLCI (the LMI) from the one over which the connection will be made. (In X.25 a call request is placed on the logical channel which will latter carry the data.)

In January 1992 there are several public frame relay offerings announced in the United States. None of these offer switched virtual circuits.

Signaling about congestion provides a Flow Control function.

The network is able to signal the attached device about congestion in the network. This is done with two bits in the address section of the frame header called the "FECN" (Forward Explicit Congestion Notification) and the "BECN" (Backward Explicit Congestion Notification) bits.

When the FECN bit is set it indicates that there was congestion along its path in the direction of flow. The BECN bit indicates that there was congestion in the network for frames flowing in the opposite direction to the frame containing the notification.

Many (if not most) products that currently use FR do not make use of congestion notification. The IBM 3745 FR attachment feature does use congestion notification to control the window size used.

8.2.3 Frame Format

The format of a frame is shown below. It is the same as a normal SDLC/HDLC frame *except* that it uses two address bytes and the control fields are absent.

- Looked at from a link control perspective, the DLCI field performs the same function as the SDLC address - it identifies the connection.
- The EA bits indicate whether the extended (3 or 4 byte) address options are in use.
- The DE (Discard Eligibility) bit tells the network that this frame can be discarded if network congestion becomes a problem. In transporting both video and voice traffic some coding schemes produce "essential" and "non-essential" frames. When a non-essential frame is discarded there is a degradation in quality but the connection still functions.

  In networks available in 1992 this bit is not used.
- The FECN and BECN bits are notifications from the network to the user equipment (DTE) that congestion exists along the path.

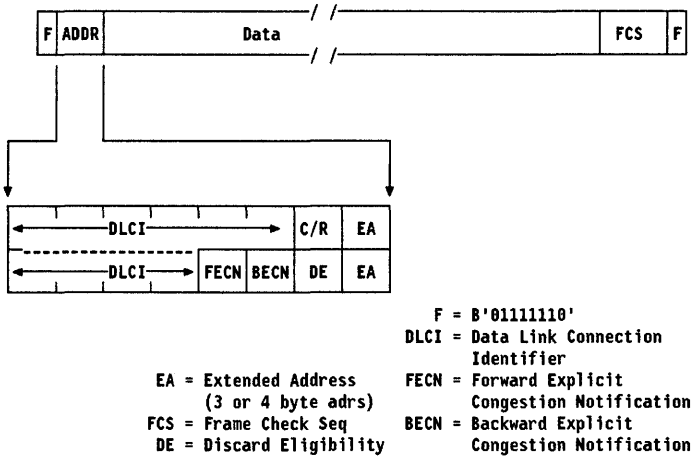


Figure 55. Frame Format. Link control fields are absent since they are ignored by the network.

8.2.4 Operation

Frame Relay is the simplest networking protocol imaginable.

1. An SDLC/HDLC type frame is sent by the end user device to the network.
2. The network recieves the frame and checks the FCS fields to determine if the frame was received correctly. If the frame was bad then it is discarded.
3. The network uses the address field to determine the destination of the data block.
4. The network uses its own internal protocols (which could be "like" Frame Relay or totally unlike) to route the frame to its intended destination. The destination is always another link within the FR network.
5. The network changes the address field in the frame header to the correct identifier for this circuit on the destination link.
6. The network sends the frame on the destination link.
7. The frame is received by the destination end user device.

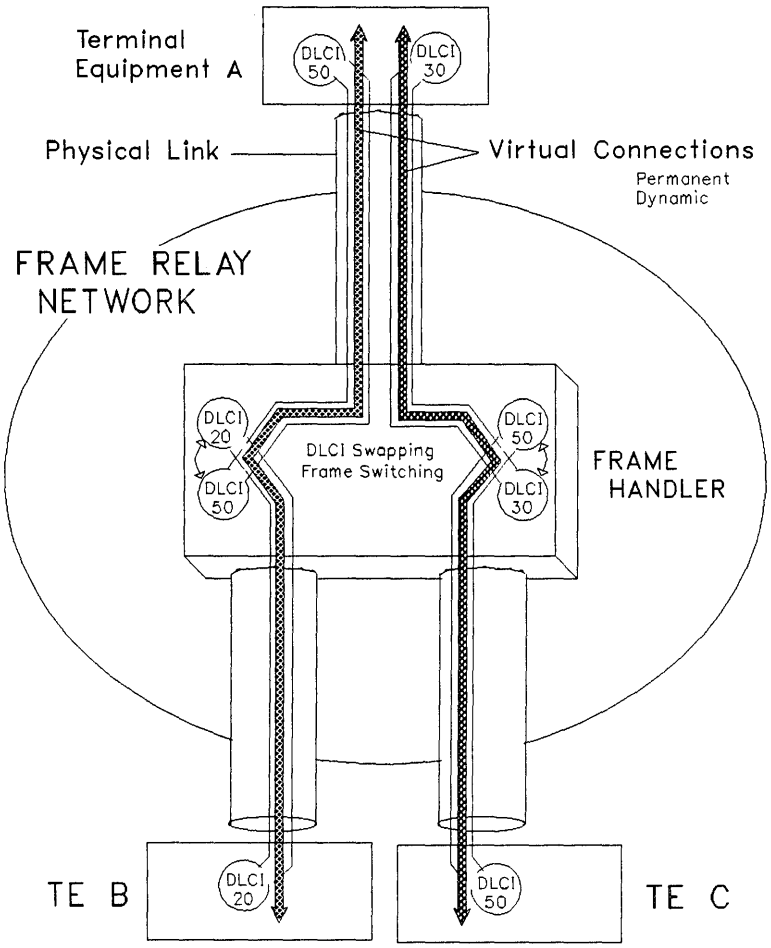


Figure 56. Frame Relay Routing Scheme

8.2.5 Characteristics of a Frame Relay Network

Lower safe link utilisations because priorities are impossible.

While the Frame Relay interface realises important performance gains because it does not know about the contents of the data, it looses the ability to prioritise *within* a single link. In SNA many sessions can be operated over a single link. These sessions may operate at different priorities and because the network can detect the priority of each session, it is able to operate at quite high resource utilisations (nodes, links..).

If there are several Virtual Links sharing a real link within the network, there is no way of giving the interactive traffic priority. The "net" of this is that node and link utilisations within the Frame Relay network cannot be as high as they could be in an SNA network for example. This means that a faster link may be needed.

Mixture of long and short blocks causes erratic response times.

The presence of a long frame and short frames mixed up within a network produces highly erratic response times. This is because the

queing delays have a big variation. (See the discussion in Appendix B.1.4, "Practical Systems" on page 287.)

The solution to this is to shorten the maximum frame length. If the length of the frame in bytes is shortened then this defeats the purpose of Frame Relay. What can be done is to shorten the frame length *in time*. You do this by using a higher speed link.

Flow control is achieved by using the link control protocol across the network.

The only available flow control is by using the link control protocol from end-to-end across the network. This means that we have lost a lot of control. To get acceptable batch throughput a large link window size will be necessary. But when the network starts to experience congestion the only control is to cut back on the window size.

The same conclusion is reached as above. The network must be fast and have a uniform delay to get acceptable throughput for batch traffic.

IBM products that use the FR interface will dynamically change the window size according to the congestion signals from the network.

In summary a Frame Relay network gains efficiency by trading network complexity for link capacity. Faster links are required in order to make the efficient protocols operate stably.

8.2.6 Comparison with X.25

At a functional level Frame Relay and X.25 are the same thing. The user has a "virtual circuit" through the network. Data is presented to the network with a header identifying the particular virtual circuit and is routed to its destination along that virtual circuit.

At a detail level they are quite different. In X.25 the basic service is a virtual call (switched virtual circuit) although permanent virtual circuits are available in some networks. Frame Relay is currently (1992) defined for permanent virtual circuits only.

In X.25 there is a rigorous separation between the "Network Layer" (networking function) and the "Link Layer" (function of delivering packets to the network). In Frame Relay both functions are integrated into the same operation. Another way of saying this is to say the in X.25 networking is done at "layer three" and that in Frame Relay networking is done at "layer two".

All of the above said, Frame Relay as an interface is much simpler than X.25 and therefore can offer higher throughput rates on the same equipment.

Some people view Frame Relay as the natural successor to X.25. Indeed many suppliers of X.25 network equipment are adding FR interfaces to their nodes. In the US, a number of public (X.25) network providers have announced that they will provide a Frame Relay service as well.

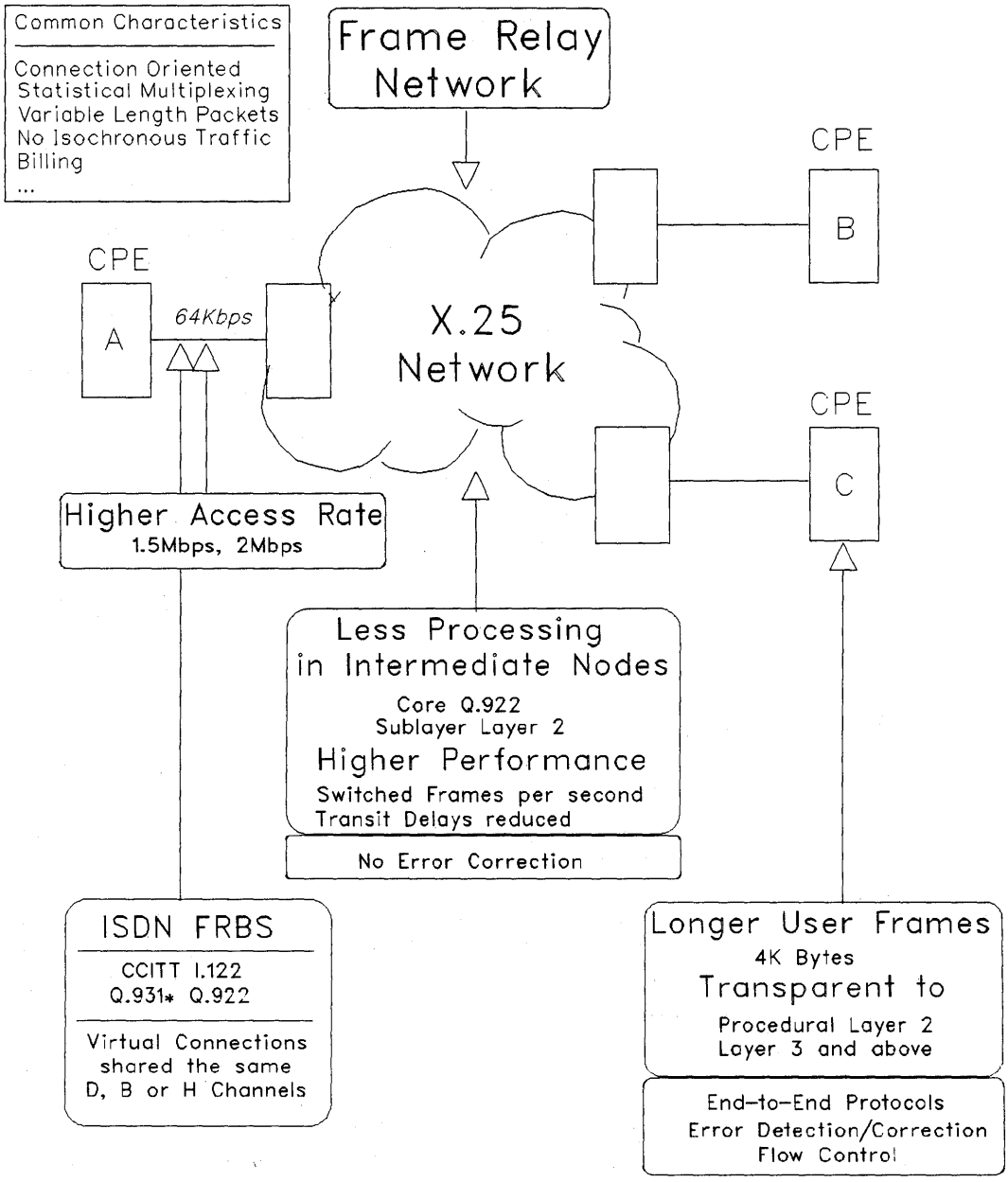


Figure 57. Frame Relay Compared with X.25

8.2.7 SNA Connections Using Frame Relay

The first consideration in running SNA over a Frame Relay network[46] is that FR is the interface to a *separate* network. At the interface between a piece of SNA equipment and a Frame Relay network, many virtual links are multiplexed over a single physical connection. The SNA equipment must be able to multiplex many virtual links over a single real link. SNA equipment already does this for X.25 and for LAN network connections, but the FR connection is little different.

SNA connections over FR networks[47] promise to be much more stable and more efficient than similar connections over X.25 networks.

Error recovery

Because there is a link control running end-to-end across the FR network, network errors will be recovered by the link control.

When SNA is used over X.25 this does not happen. An error in the network causes the catastrophic loss of all sessions using that virtual circuit.[48]

Interface Efficiency

Because there is no packetisation or packet level protocol to perform, the FR network interface is likely to use significantly less resource within the attaching SNA product than is required to interface to X.25.

Network Management

FR has a signaling channel which allows the exchange of some network management information between the device and the network. X.25 has no such ability.

Nevertheless, at the current state (1991) of the definition it is not possible to provide full seamless integration of SNA network management with the management of the FR network over which it will run. To some extent the FR network (like X.25 networks) will form a "black hole" in the SNA network management system.

Multidrop

Although it is not in the FR standard, there is a potential ability to use FR for limited "multidrop" connection for devices located in close physical proximity to one another. This is a real problem in X.25 since the X.25 interface is strictly point-to-point, so that if there are many devices in the same location many links to the X.25 network are necessary.

Using FR, a relatively simple "interface splitter" device could be constructed which would allow the connection of multiple FR devices to the same network link attachment.

---

[46] **Information in this section is derived from an early SNA prototype implementation. The description is conceptual and may not accurately reflect the operational detail of any product.**

[47] There is no such thing as a Frame Relay network. Frame Relay is a network interface protocol not an internal network protocol. The same is true of the phrase "X.25 network". X.25 is an interface not a network protocol. However, it is convenient to refer to "FR networks" and to "X.25 networks" to mean "networks that support FR interfaces" and "networks that support X.25 interfaces". In the case of Frame Relay, standards work is under way to specify the interconnections between nodes within a network using Frame Relay protocols but the work is not yet complete. Nevertheless, the vast majority of FR networks proposed in 1992 do not use FR protocols internally.

[48] Except in the case of the IBM System/36 and IBM AS/400 which are able to use an end-to-end link protocol (called ELLC) across the X.25 network. ELLC uses the elements of procedure of LAPB as an end-to-end network protocol.

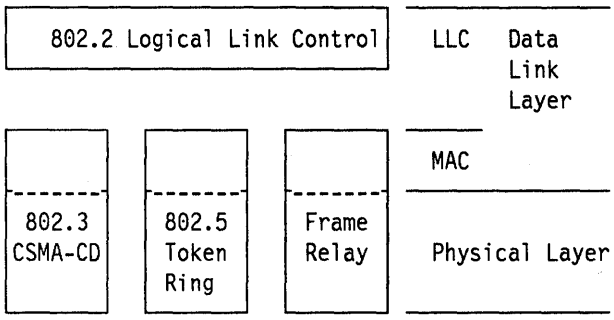


Figure 58. Frame Relay in Relation to IEEE Standards. Frame Relay functions as an alternative MAC layer which is used by the logical link control and medium access control functions.

Frame Relay support in SNA is an extension of the LAN support. Figure 58 shows how Frame Relay may be situated logically in relation to the two most important LAN architectures. There are differences of course.

1. The LAN environment at the Media Access Control level (MAC layer) is a connectionless environment. You can send data to any device at any time provided you know the correct address to send to.

   Frame Relay provides connections across the network. In the early networks these connections must be predefined although later there will be an ability to set up connections on demand (switched virtual circuits).

2. The LAN environment provides a broadcast function but Frame Relay does not.
3. A Frame Relay network can provide congestion information to attaching devices where a LAN typically does not.
4. The network management environment is quite different.
5. An end station in a LAN environment must support source routing protocols in order to allow communication across "source routing" bridges. In Frame Relay this is not needed.

These differences are either irrelevant or can be fairly easily accomodated.

Most important are the similarities:

1. Both SNA and Frame Relay are connection oriented systems. When SNA uses a LAN connection it usually builds a switched connection across the LAN.
2. Both LAN and Frame Relay take a whole data frame and transport it to another user. The functions of framing, addressing, error detection (but not recovery) and transparency are handled by the network attachment protocol (MAC layer).
3. In both LAN and Frame Relay a single physical attachment to a device may contain many virtual links to other devices.

The link control protocol used across the LAN is called IEEE 802.2. This is just another link control protocol like SDLC, LAPB and LAPD (all forms of HDLC). The difference is that SDLC, LAPB and LAPD perform the functions of framing, transparancy (via bit-stuffing), error detection and addressing. In the LAN environment these functions are provided by the MAC protocol and in Frame Relay they are provided by the Frame Relay link control. IEEE 802.2 is simply a

link control protocol that leaves the responsibility for framing, addressing etc. to the MAC function. Thus 802.2 provides exactly the function that is needed for Frame Relay. In addition 802.2 uses an addressing structure that allows multiple Service Access Points (SAPs), which provide a way of addressing multiple independent functions within a single device.

Thus the link control used by SNA across a Frame Relay connection is IEEE 802.2 (the LAN link control). In addition SNA devices use the congestion information provided by the Frame Relay network to control the rate of data presented to the network (flow control).

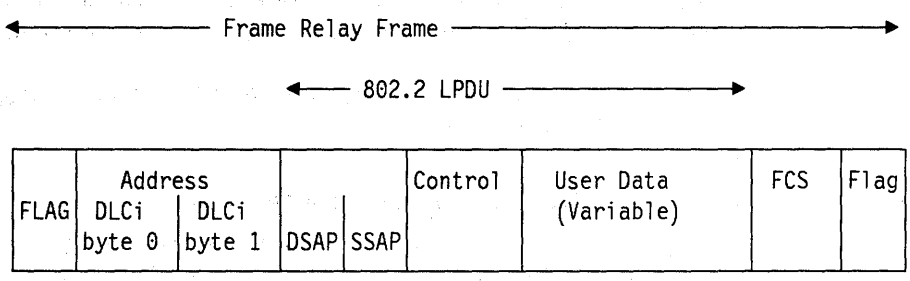


Figure 59. Frame Relay Format as Used in SNA. The LPDU format is exactly the same as that used in the LAN environment.

The HDLC family of link controls uses a "rotating window" scheme to provide delivery confirmation. Multiple blocks may be sent before the sender must wait for an acknowledgement from the receiver. This means that several blocks may be in transit at any time and compensates for propagation delays.

In IEEE 802.2 this rotating window mechanism is used for flow control across a LAN. Since most LANs contain bridges there is a need to control the flow of data in the case where the bridge becomes congested. The mechanism is exactly suited to use across a Frame Relay network.

The flow control mechanism operates at the sender end only and the receiver is not involved in its operation. The transmitter is allocated a number (n) of blocks it may send before it must wait for an acknowledgement. The receiver acknowledges blocks by number and an acknowledgement of block number 3 (for example) implies correct receipt of blocks numbered 1 and 2 (but in practice most receivers acknowledge every block).

When the transmitter is notified by the FR network of congestion along its forward path (by receiving the BECN bit in the header of a received frame)[49] it reduces its send window size to 1. In the uncongested state there may be n frames in the network between transmitter and receiver. When congestion is detected this is immediately reduced to 1.

As operation continues if the transmitter has had m (an arbitrary number) responses without seeing another congestion notification then the send window n is increased by one. This will continue to happen until the maximum transmit window size n is reached.

[49] The transmitter's "forward" path is the "backward" path of blocks being received. Hence it is the BECN bit that a transmitter must examine.

The described mechanism is about as much as any end user device can do. It is really a delivery rate control. However, it must be noted that if FR networks are to give stable operation then they will need flow and congestion controls internally.

8.2.8 Disadvantages

As discussed above (section 8.2.5, "Characteristics of a Frame Relay Network" on page 151), faster links are required than would be needed in a better controlled network (such as an SNA network) to handle the same amount of traffic. This is partly because the network cannot know about priorities within the link and partly because of the variability in frame lengths allowed.

In the environment where the cost of wide area links is dropping very quickly, this may not be important.

8.2.9 Frame Relay as an Internal Network Protocol

Many people assume that FR networks will use the logical ID swapping technique (described in section 5.7.3, "Logical ID Swapping" on page 93) for internal network operation. Indeed this would be a good technique for FR since it is easy to implement on existing networking equipment.

The advantage of FR internally within the network then lies in the ability to "throw away" error frames and not handle error recoveries. This means that far fewer instructions are needed for the sending and receiving of data on intermediate links. Buffer storage requirements within intermediate nodes are reduced because there is no longer any need to hold "unacknowledged" data frames after they have been sent. Standards work is under way to specify this method of operation and the interconnections between FR nodes (using FR) within a network.

However, it seems unlikely that FR will be widely used in this way. Many people see FR as an interim technique which will give a substantial improvement in data throughput on existing equipment. In the real world, networks offering FR interfaces will also offer X.25 and perhaps LAN routing interfaces as well. These networks will continue to use their existing internal network protocols (many use TCP/IP internally).

8.3 Packetised Automatic Routing Integrated System (PARIS)

Paris is an experimental very high speed packet switching system developed by the IBM T.J. Watson Research Center at Yorktown Heights, New York. It was built to develop a better understanding of the principles and problems involved in high speed packet switched communication and has been used in a number of field trials.[50]

The objectives of the Paris development were to build a system appropriate to a private backbone network supporting voice, video, data and image. Such a private network might have a relatively small number (less than 20) of 100 megabit links around the US. (In fact the principles and technology involved here can be extended quite easily to large networks involving perhaps thousands of links and interconnections.)

The underlying principles are those outlined in Chapter 5, "Principles of High Speed Networks" on page 73. To minimise the delay in each switching node:

- Throw away error data without retry (as in Frame Relay). This prevents delays on links caused by retries and can be accommodated quite well by appropriate end-to-end protocols at the entry points to the network. This also minimises buffer use in switching nodes since data doesn't have to be stored after transmission in case a retry is needed.
- Use a protocol such that the data switching function can be performed in hardware without software involvement. This means there needs to be a new and innovative approach to flow controls.
- Use variable length packets, because this lessens the load on the switching nodes. Statistics dictate that the maximum block size be such that its transmit time cannot have too drastic an effect on link queueing delays. See Appendix B.1.4, "Practical Systems" on page 287. The maximum packet size allowed in Paris is 4 KB but that is quite arbitrary and is dictated mainly by the link speed and the amount of buffer storage allocated in the switching nodes.
- Adopt a simplified priority scheme. Priorities cause extra complexity (= extra delay) in switching nodes. The system aims to "guarantee" a very short nodal delay which, if achieved, makes priorities (especially for voice traffic) unnecessary. There is a very simple priority mechanism in Paris which is discussed later.

[50] Paris is a research project *not* a product. IBM cannot make any comment on what uses, if any, may be made of the Paris technology in future IBM products.

8.3.1 Node Structure

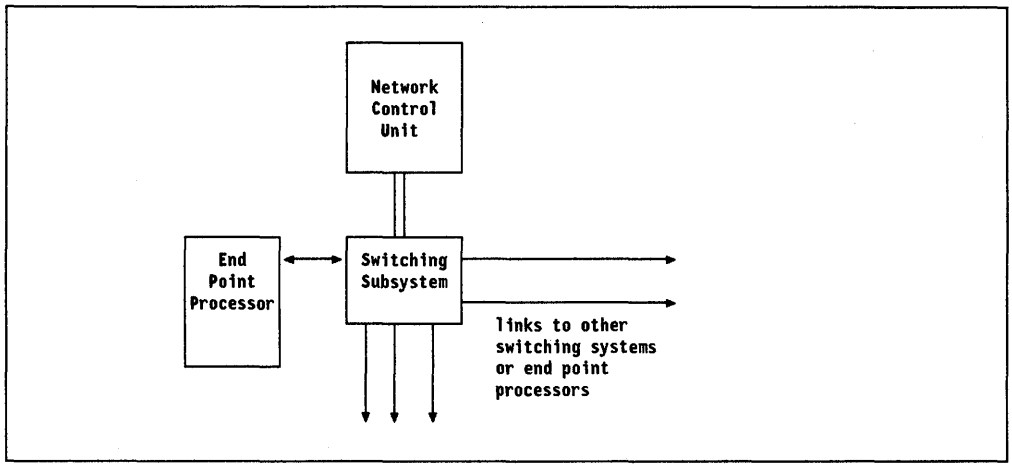


Figure 60. Paris Node Structure

Figure 60 is a schematic representation of the elements of a Paris network.

Switching Node

An intermediate switching node is comprised of a Network Control Unit closely linked to a Switching Subsystem.

Network Control Unit

This is a general purpose processor. It contains:

- A network map (topology database)
- Route selection function
- Network directory
- Capacity allocation function

Switching Subsystem

This is the business end of the system - the frame switch. It handles the routing and buffering of data and the priority mechanism.

End Point Processor

This processor links the Paris backbone network to the outside world. It must:

- Contain an interface to the switching subsystem which is able to send data blocks in the required format.
- Because every data packet sent must contain routing information, the EPP must contain either a network control function (with topology database etc.) or a function that can obtain network topology from the nearest network control function.

Some endpoint functions are:

- Voice packetisation and playout
- Flow control (admission rate control)
- Error recovery
- External interfaces and protocol adaptation etc.

8.3.2 Automatic Network Routing (ANR)

The heart of the Paris system is the system of switching data in intermediate nodes.

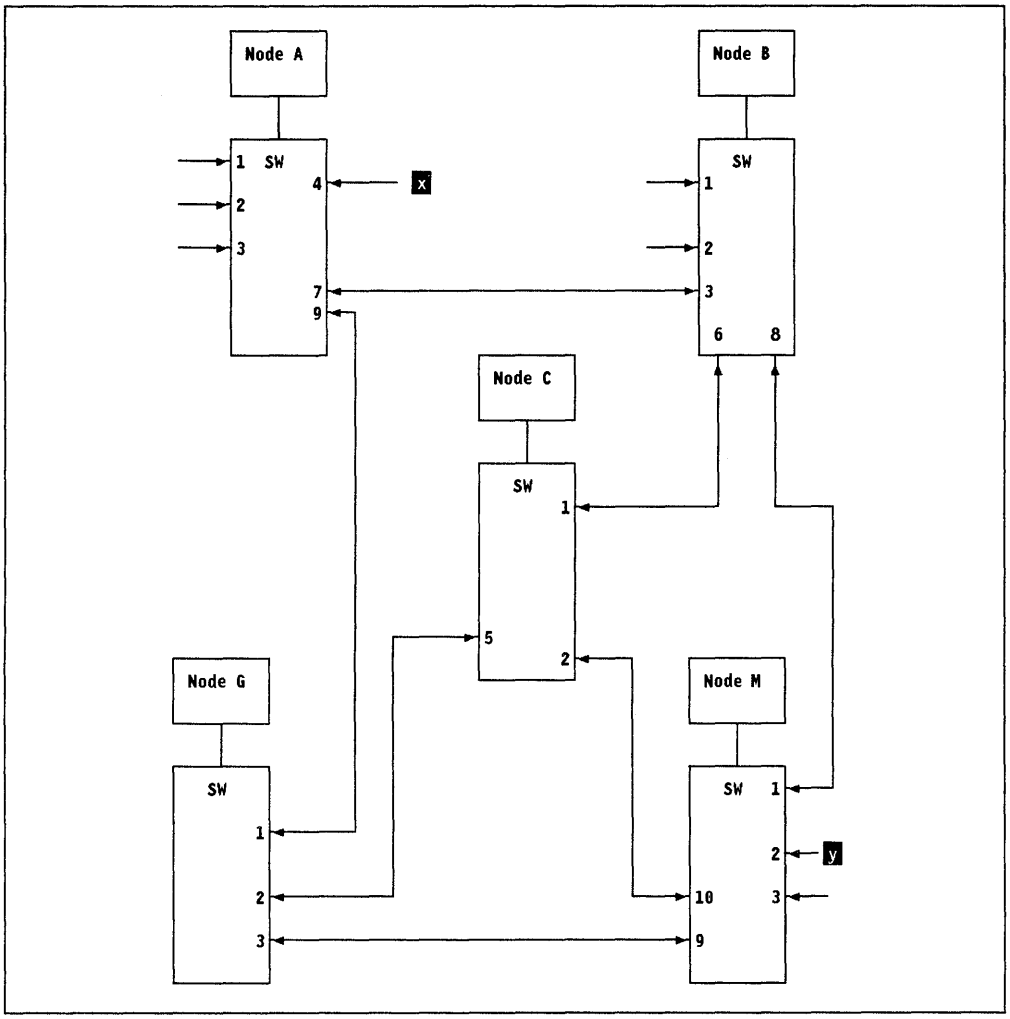


Figure 61. Paris Routing Network Structure

Switching is "connectionless". The switching subsystems process data packets without ever knowing or caring about connections. No connection information is kept in the switching subsystem

Instead, the full route that the packet must take is included in the beginning of the packet itself.

> *This is very similar in principle to the "source routing" method of interconnecting LANs. The method is also used in the connection setup process in APPN. In APPN, the session setup control message (the BIND) carries its route as a control vector within itself. Data switching in APPN is done via connection tables in each node but setting up connections is done by the routing vector method.*

Figure 62 on page 161 shows the format of a data packet. The "routing information" field shown contains the ordered set of links specifying the route a packet must take. Refer to Figure 61. The numbers located within each node, adjacent to the link attachment represent link numbers *within that node*. These are used in the following discussion.

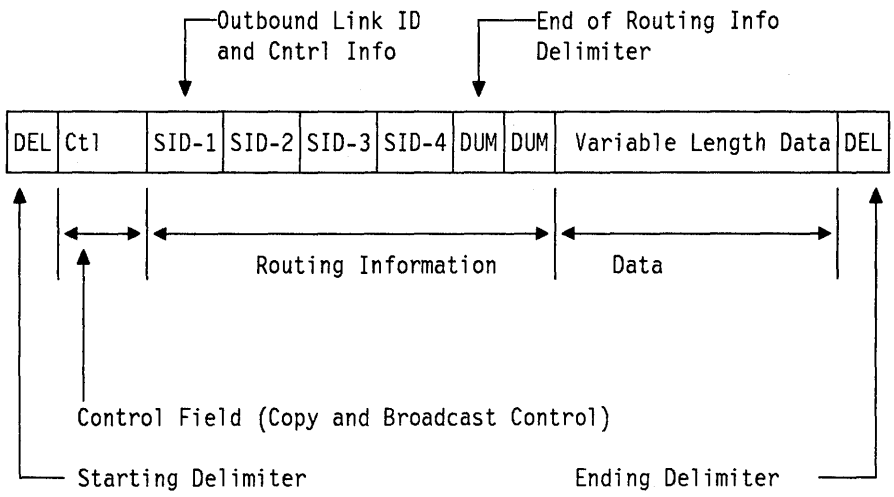


Figure 62. Data Packet Format

In Figure 61, imagine that we wish to send a packet into the network at point **x** and have the network deliver it to point **y**. The packet could go by several routes but one possible route is through node B. The routing vector would contain:

7,8,2,,,

- Since the packet arrives at node A there is no entry in the vector to denote node A.
- When the packet arrives at node A, node A looks at the first entry in the routing vector. That is the number 7.
- The entry "7" is removed from the routing vector.

  (This enables the next switching function to look at a constant place in the received packet for its routing information. This helps reduce the amount of logic needed in the switching hardware.)
- The packet is queued for link 7.

Notice here that there is nothing in the routing vector to specify the node name or number. The routing vector now looks like:

8,2,,,

The data arrives at node B and the process is repeated:

- The number 8 is removed from the routing vector and the packet enqueued for link 8.

When the packet arrives at node M, by the same process it will be sent on link 2. Now all that is left of the routing vector are two null entries and so link 2 must be connected to an endpoint processor.

There are several possible routes for our packet:

7,8,2,,,

9,3,2,,,

9,2,2,2,,,

9,2,1,8,2,,,

Any or all of these could be used to send packets from point x to point y. The switching subsystems keep no record of connections and route data only based on the routing header.

In a practical system, there will be a Frame Check Sequence field either in the header or at the end of the data or in both. A receiving node should check the FCS field before routing the data, as an error in the header could cause misrouting of the packet.

This system places the responsibility for determining the route onto the sending system. This means that the endpoint processor must have a means of calculating an appropriate route. Hence the need for the endpoint processor to have either a continuously updated topology database or access to a network control function that has this information.

Sending data in the opposite direction is exactly the same process except that y is now the origin. This leads to the fact that a real connection between two end users can take two completely different paths. In a practical system, for management and control reasons, it would probably be necessary to have data belonging to a single "call" (connection through the network) travel on the same path. There are a number of potential ways of achieving this.

8.3.3 Copy and Broadcast Functions

A number of copy and broadcast functions are possible within the system.

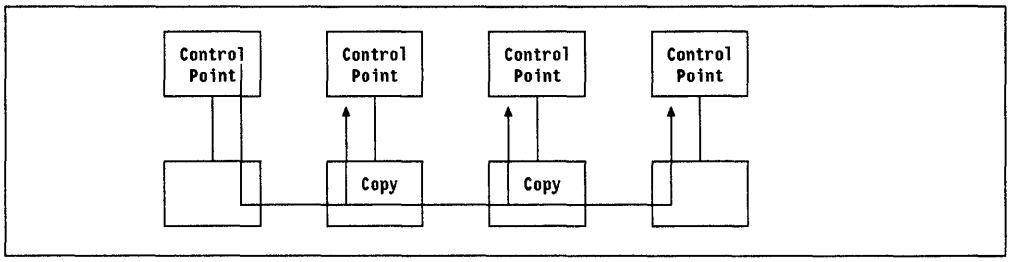


Figure 63. Connection Setup Using Linear Broadcast with Copy

Broadcast

In the Control field of the packet header there is a single bit which causes the switching node to send a copy of the packet onto all its outbound links (after removing the routing vector).

Selective Broadcast

The SID contains more information than just the identifier of the next link. It contains 4 bits of control information. One of these is for selective broadcast. When a SID is removed from the front of a packet, it is examined and if the broadcast bit is "on" the packet is routed to all outbound links on this node (except the link that the packet arrived on). It is possible to broadcast to only a subset of nodes - hence the use of the word "selective".

Copy

A copy bit exists in the Control field that directs every switching node along the route to send a copy of this packet to its network control processor.

Selective Copy

A bit to control this function is also in the SID. Selective copy copies the packet to the control processor only for selected nodes along the path.

These broadcast and copy functions are important for network control and connection setup procedures.

8.3.4 Connection Setup

Although the network data switching function itself does not know about connections, these are present between the endpoints of the network.

- When an End Point Processor (EPP) wants to start a connection it requests a route from the nearest Network Control Unit (NCU).

  The NCU keeps a full topology database of the network, which includes every link with current link utilisations. These link utilisations are continuously calculated by a monitor function within each node and when a significant change in loading occurs, the change is broadcast to every NCU in the network.

- The NCU calculates the route as a sequence of SIDs for *both* directions between the endpoints and sends it back to the requesting EPP.
- The EPP then sends a connection setup request frame to the desired partner EPP with the copy function set.
- The switching systems copy the request to every NCU along the desired route. Each NCU will then check to make sure that sufficient capacity is available for the desired connection.

  This is a complex decision. For a data call the amount of required capacity will (on average) be many times less than the peak rate. The node must allocate an appropriate level of capacity based on some statistical "guess" about the characteristics of data connections. Voice and video traffic have quite different characteristics and the NCU must allocate capacity accordingly.

- For important traffic classes a second path may be precalculated so that it will be immediately available in case of failure of the primary path.
- Each NCU along the path will reserve a certain amount of bandwidth for the connection. This bandwidth reservation is repeated at intervals. If a specified time elapses and there has not been another reservation for this connection, the NCU will de-allocate the capacity.
- An NCU may disallow a connection request and notify the requesting EPP.
- The destination EPP replies (also copying each NCU along the path) using the reverse path sent to it in the connection request. This ensures that both directions of a connection take exactly the same path (same links and nodes). This is important in error recovery and management of the network.

8.3.5 Flow and Rate Control

The primary method of flow control in the Paris network is a "delivery rate control" system rather than a flow control mechanism. The concept is to control the rate at which packets are allowed to enter the network rather than controlling individual flows. All flow and rate control in a Paris network takes place in the EPP.

The scheme used is called "leaky bucket" rate control. In order for a packet to pass the leaky bucket the counter must be non-zero.

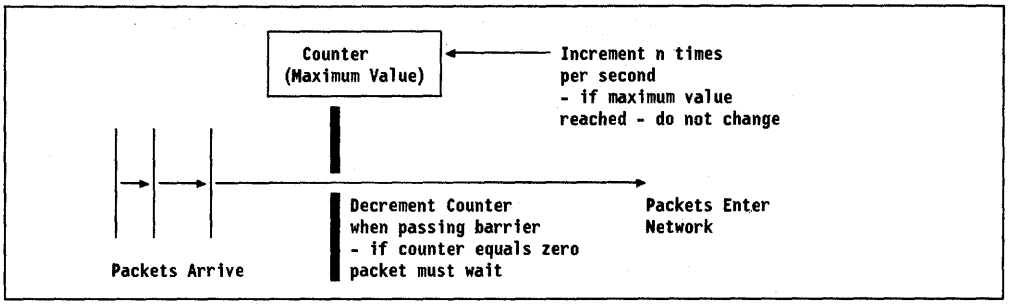


Figure 64. Leaky Bucket Rate Control

The "leaky bucket" is a counter which has a defined maximum value. This counter is incremented (by one) n times per second. When a packet arrives it may pass the leaky bucket if (and only if) the counter is non-zero. When the packet passes the barrier to enter the network, the counter is decremented.

This scheme has the effect of limiting the packet rate to a defined average, but allowing short (definable size) bursts of packets to enter the network at maximum rate. If the node tries to send packets at a high rate for a long period of time, the rate will be equal to "n" per second. If however, there has been no traffic for a while, then the node may send at full rate until the counter reaches zero.

Paris in fact uses two leaky buckets in series with the second one using a maximum bucket size of 1 but a faster clock rate. The total effect is to limit input to a defined average rate but with short bursts allowed at a higher rate (but not the full speed of the transmission medium). The scheme is a bit conservative but allocates capacity fairly. Paris has an adaptive modification based on network congestion. It can alter the maximum rates at which the buckets "leak". The network provides feedback via a congestion stamp on the reverse path. This feedback is used to alter the rate at which the counters are incremented and thus the rate at which packets are able to enter the network.

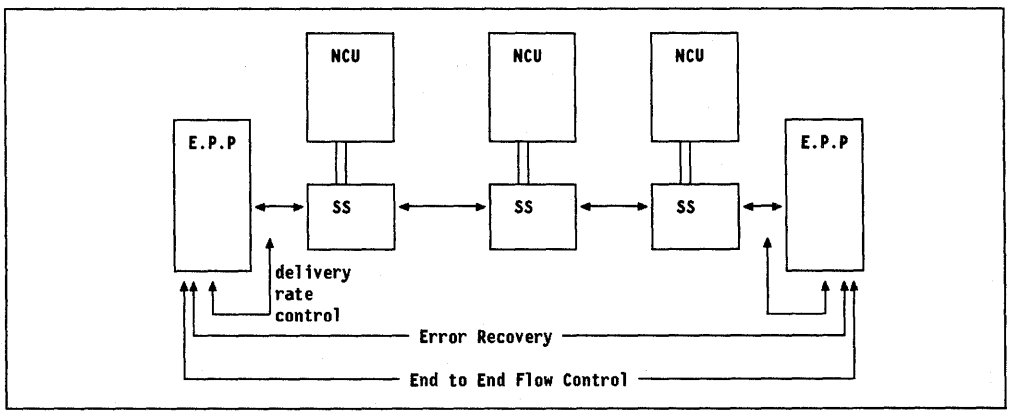


Figure 65. Paris Flow and Rate Control

In addition to the delivery rate control, which restricts the rate at which data may enter the network, there are end-to-end protocols which allow for error recovery and for flow control for each individual connection.

These controls are different depending on the type of connection. For data a protocol is used that allows for error recovery by retransmission of error (or lost)

packets and also provides flow control to allow for speed matching between the end users. Voice traffic does not have any retransmission for recovery from errors but it does need a method of playout that allows for the possibility of irregular network delays and for lost packets etc.

8.3.6 Interfaces

8.3.6.1 Physical Interfaces

The Paris system will operate over almost any clear channel physical interface of appropriate speed. Typical connection could be through G.703 electrical interface to an optical transmitter at (say) 150 megabits. Also possible would be connection to Sonet/SDH at a number of different rates such as STS-3c.

8.3.6.2 User Data Interfaces

End user equipment is connected to the network via an adaptation (or interfacing function). As with ATM (see section 7.1.8, "ATM Adaptation (Interfacing) Layer (AAL)" on page 137), an adaptation layer is needed to interface and adapt the network to different kinds of user traffic. A number of different interfacing layers (or modes of operation of a common protocol) would be needed depending on the type of traffic to be handled.

Paris requires much less adaptation function than does ATM. This is because:

- Paris sends full frames (there is no need to break logical user blocks of data up into small units for transmission).
- Paris performs error detection on the data part of the frame as well as the routing header. There is no need for any additional function to provide error detection on the data.

In addition to the adaptation function there is also an interfacing function needed which converts the network interface protocol to the internal network protocol. This is not quite the same thing as adaptation.

For example, a Frame Relay service could be built using a Paris backbone network very easily. To use a Paris network for TCP/IP traffic requires that IP routers be built at the Paris network entry points. These routers would have to accomodate the difference between the connection oriented nature of the Paris network and the connectionless nature of IP (this adaptation is a standard function of IP).

Handling SNA is a different matter and is discussed in section 5.8.1, "SNA in a High Speed Network" on page 97.

8.3.6.3 Transporting Voice

The principles of handling voice in a high speed packet network are discussed in section 5.2, "Transporting Voice in a Packet Network" on page 79. These principles hold for voice transport in a Paris network. The major difference is that a packet size larger than 48 bytes may be used. This means that echo cancellation is necessary.

This is discussed fully in *A Blind Voice Packet Synchronisation Strategy* (see bibliography).

8.3.7 Performance Characteristics

ANR routing is easily handled in hardware.

- All the information needed to route the data on to the next link is available within the ANR field of the packet header. Data switching can be done in hardware without reference to connection tables or network topology information.
- There is no requirement for the NCU processor to keep updating tables in the switching subsystem when new connections are created or terminated.

Switches frames not cells.

Switching full data link frames rather than small cells has many advantages:

- Proportionally lower overheads due to packet headers.
- Lower overheads in the switching process because the switches tend to require a given amount of processing per packet regardless of packet length. This is discussed in section 5.5, "Transporting Data in Packets or Cells" on page 86.
- Lower overheads in the endpoint processors due to segmentation and reassembly. Some segmentation is still required but that can be done relatively easily in software.
- Lower overhead in the EPPs due to simultaneous reassembly of packets from cells. If cells belonging to multiple packets were able to arrive intermixed with one another, the EPP would need to have many reassembly tasks and buffers to rebuild the packets.

Provides selective copy and multicast.

These abilities make the management and control functions within the network significantly more efficient than more conventional methods.

Low overhead due to routing headers.

Assuming a two-byte SID in a practical network the maximum likely ANR field is 14 bytes (7 hops). (It has been shown that 5 hops is the practical maximum in most public and private large networks.) As the packet is routed through the network the size reduces as SIDs are removed. This is very small compared with existing systems - SNA subarea networks carry a 26-byte routing header (called a "FID_4 TH").

Since packets in the Paris system are full data link frames (rather than short cells), the proportion of overhead caused by the routing system is very low.

Uses clear channel links.

The system requires internode links that operate on "clear channels". That is, it does not need complex link connections with preformatted frames (such as G.704).

Designed as an integrated system.

The whole Paris system was designed to work as a whole with network control and management, user interfaces, and end-to-end protocols included as parts of an integrated whole.

A part of the aim of the research project was to demonstrate the effectiveness of designing the system as a whole rather than as a number of disjoint functions. This has been very successfully demonstrated.

The disadvantages are minimal.

Variable length frames introduce variability into transit delays.

This is certainly true if the link speed is low relative to the maximum packet length. On the very high speed links to be used by Paris, this is not a concern. See Appendix B.1.4, "Practical Systems" on page 287.

Longer voice packets introduce the need for echo control.

Paris uses 128-byte voice packets which require longer to assemble than would 48-byte ATM voice cells. This introduces a "propagation delay" which can increase the effect of echoes. There is no echo in a digital link. But there are echoes from analogue (two-wire) tail circuits and from both analogue and digital handsets. See the discussion in section 5.2.2.2, "The Effect of End-to-End Network Delay on Voice Traffic" on page 81.

Chapter 9. Shared Media Systems (LANs and MANs)

9.1 Basic Principles

Local Area Networks (LANs) and Metropolitan Area Networks (MANs) consist of a common cable to which many stations (devices) are connected. Connection is made in such a way that when any device sends then all devices on the common medium are able to receive the transmission. This means that any device may send to any other device (or group of devices) on the cable.

This gives the very obvious benefit that each device only needs one connection to the cable in order to communicate with any other device on the cable.

> An alternative would be to have a pair of wires from each device to each other device (meaning n × (n-1)/2 connections between n devices would be needed - for 10 devices this would mean 45 separate connections).

Because the LAN is a shared medium if two devices try to send at the same time then (unless something is done to prevent it) they will interfere with each other and meaningful communication will be impossible. What is needed is some mechanism to either prevent devices attempting to send at the same time or to organise transmission in such a way that mutual interference does not result in an inability to operate.

There are three generic ways in which the medium (cable) may be shared and this is often used to classify LANs.

Broadband LANs

In a broadband LAN, frequency division multiplexing is used to divide the cable into many separate channels. This principle is described further in Appendix A.1.1, "Frequency Division Multiplexing" on page 275.

The most common example of a broadband LAN technique is in cable television where many TV signals are multiplexed onto the same coaxial cable via frequency division multiplexing.

Time Division Multiplexing LANs

The method of operation of a TDM LAN is exactly the same as the method used in modern digital PBXs, except that the ring connecting the "users" is no longer enclosed within a single box but is carried around an area over some medium such as twisted pair or coaxial cable.

When one device needs to send to another, a slot is allocated (there are ways of doing this without a ring controller) and this time division slot is used to establish a slower speed channel between the devices. This is exactly as it is done within the PBX but potentially, at least, without the controlling computer.

A very early system of this kind is the very slow speed IBM 3600/4700 B_LOOP protocol.

Another system on the market uses the standard G704 slot structure at 2 Mbps over a twisted pair. This enables the use of mass produced easily available chips. This method is low in cost, allows for the easy use of voice and data on the same medium and can be made reasonably

flexible. However, its use is limited to a maximum of 15 simultaneous communications over the shared ring.

Many of the new high speed LAN protocols (FDDI_II, DQDB...) which are discussed in this book have provisions to handle TDM style traffic.

Baseband LANs

In a baseband LAN the cable is treated as a single high speed channel. If Box A wants to send something to Box B on the LAN in principle all it has to do is put a destination header on the block and send it. But what if another device is already sending? This is the major problem for the LAN. It leads to several techniques (protocols) for control of access to the transmission medium.

Devices on a baseband LAN can be almost anything digital, including digitised voice, data and video, etc.

Some LANs use a baseband technique within one or a small number of frequencies on a broadband LAN (a form of sub-multiplexing discussed in Appendix A.1.4, "Sub-Multiplexing" on page 278).

9.1.1 Topologies

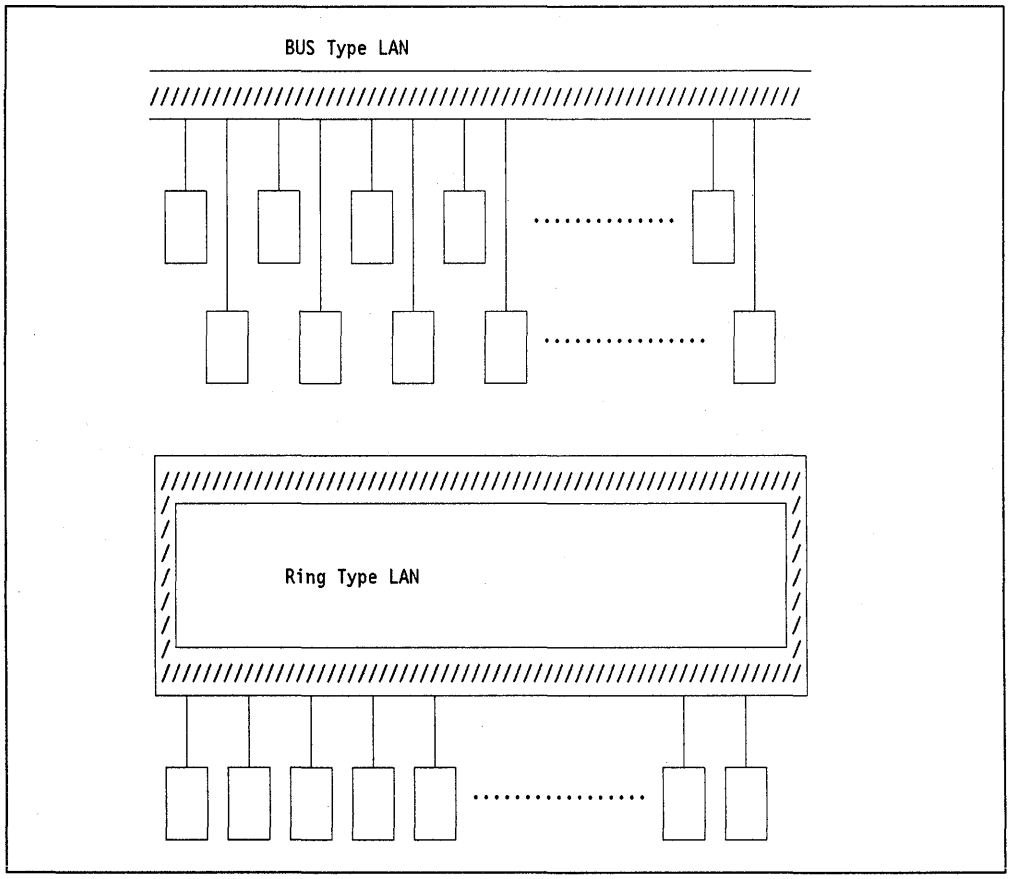


Figure 66. Local Area Networks

There is an almost infinite variety of ways of connecting LAN cable. The two most common types of LAN connection are illustrated in Figure 66.

The LAN protocol used, the configuration of the cable and the type of cable are *always* intimately related. Cables vary from two-wire twisted pair telephone

cable to optical fibre and speeds vary from 1 megabit per second to many gigabits per second.

The principle LAN configurations are:

Rings

A ring style LAN is one where data travels in one direction only and having passed through (or by) every node on the LAN returns to its point of origin.

Ring LANs may be wired in many ways. The ring may fan out from a hub such that while the data path is a ring the wiring may be a star. This is very common as it helps in fault bypass.

Dual Rings

Some LANs use two rings operating in opposite directions.

Busses

The basic idea of a bus is that data placed on it travels in both directions from its point of origin to every connected device.

Directional Busses

In a true bus environment it is hard to synchronise multiple stations with one another. Many times busses are used which are unidirectional and have a head end which generates synchronisation and perhaps framing. To get full data transfer capability unidirectional busses are usually used in pairs (one for each direction).

It is important to note that on a LAN, if communication is to be meaningful, all stations must use the same signaling techniques and access rules (protocols).

9.1.2 Access Control

The biggest problem in a LAN is deciding which device (end user) can send next. Since there are many devices connected to a single communications channel, if more than one device attempts to send then there will be a collision and neither transmission will be successful.[51]

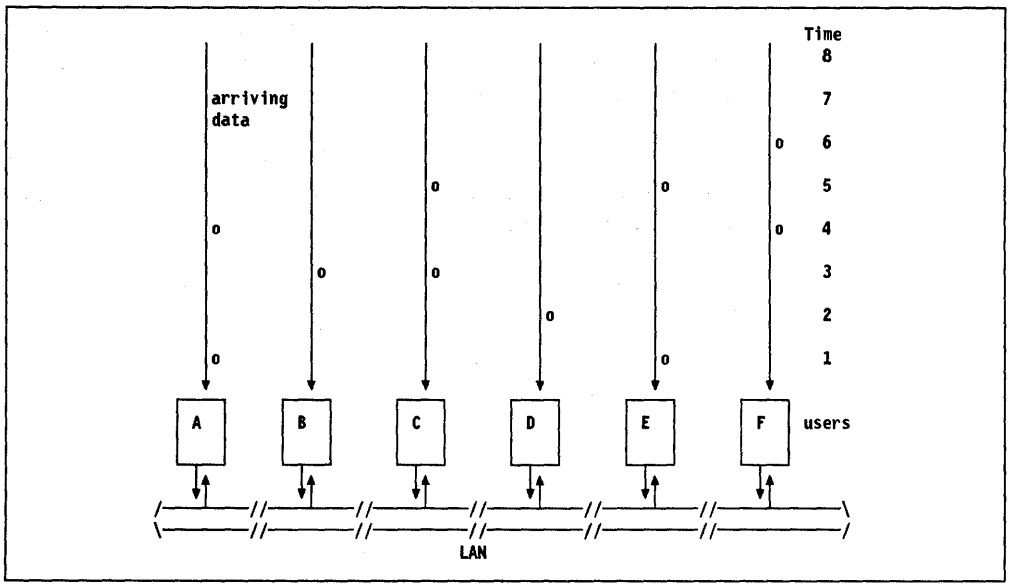


Figure 67. Transactions Arriving at a Hypothetical LAN. Arrivals are separated in time and space. The problem is how to decide which device can send next.

Figure 67 shows a LAN with six devices (labelled A to F) attached. The arrow above each device represents the flow of data being generated by the device for transmission to some other device on the LAN. An "o" represents a block of data generated and the numbers on the right of the diagram represent the passage of time. Thus:

1. At time "1" devices A and E generate some data to send on the LAN.
2. At time 2, device D generates a block of data to send.
3. At time 3, devices B and C generate data...

If the system is to be satisfactory then each user must get a "fair share" of the LAN. This is usually taken to mean that:

- Data should be sent on the LAN in the order that it "arrives" (is generated).
- No device should be able to monopolise the LAN. Every device should get equal service.
- Priority may be given to some types of traffic or user in which case higher priority traffic should receive access to the LAN before lower priority traffic.
- Within each priority level data should be sent in the order that it was generated and every device should get equal service.

Consider Figure 67 again. Even though they are distributed over many locations separated perhaps by great distances, transactions arriving at the LAN form a

[51] There is an exception in the case of analogue radio-based LANs where the use of Frequency Modulation (FM) transmission ensures that the strongest signal is received correctly.

single logical queue. The objective is to give access to the LAN to transactions in the queue in FIFO (First In First Out) order.

If each device was able to know the state of the whole queue and schedule its transmissions accordingly then the system could achieve its objective.

Unfortunately devices are separated from one another perhaps by many kilometers. Because of the geographic separation it takes time for information to travel from one device to another. The only communication medium available to them is the LAN itself!

This then is the problem for any LAN system. **A LAN system aims to provide "fair" access for all attached devices but it is not possible for each device to know about the true state of the notional "global queue".**

Fairness

A real communication system has other things to worry about than fairness. Users are concerned with what functions a system delivers and, importantly, at what cost. The challenge in design of a LAN protocol is to deliver the optimal cost performance characteristic. Fairness and efficiency are important but the resulting system is the objective.

There are many ways of approaching the ideal of "fairness".

SA Send Anyway...

When a device has something to send it just sends anyway without regard for any other device that may be sending.

This has never been seriously used for a cable LAN but was used in the "Aloha" system where multiple devices used a single radio channel to communicate one character at a time. Using Frequency Modulation (FM) the strongest signal will be correctly received and the weaker signal(s) will be lost.

This technique works but at very low utilisations. It requires a higher layer protocol capable of retrying if data is lost.

Contention with Carrier Sense (Carrier Sense Multiple Access (CSMA) with or without Collision Detection (CD))

Using this technique, before a device can send on the LAN it must "listen" to see if another device is sending. If another device is already sending, then the device must wait until the LAN becomes free. Even so, if two devices start sending at the same time there will be a collision and neither transmission will be received correctly. In CSMA/CD, devices listen to their own signal to detect collisions. When a collision occurs the devices must wait for different lengths of time before attempting to retry. This collision detection feature is present in some techniques and not in others. Either way, each user of the LAN must operate an "end-to-end" protocol for error recovery and data integrity.

In all CSMA type LANs there is a gap in time between when one device starts to send and before another potential sender can detect the condition. The longer this gap is, the higher the chance that another sender will try to send and, therefore, the higher the possibility of collision. In practice one of the major determinants of the length of the gap is the physical length of the LAN. Thus the practical efficiency of this

kind of LAN is limited greatly by the physical length of the LAN. The utilisation of the carrier medium (usually a bus) is limited more by collision probabilities than by data block sizes. In some situations, 20% is considered quite good.

Performance:

- As the data transfer speed of the LAN increases, throughput does not increase at the same rate. Faster link speeds do nothing to affect the propagation delays. Thus the length of the "gaps" during which collisions can occur becomes the dominant characteristic.
- There is no way of allocating priorities.
- Fairness of access to the LAN is questionable.
- Low access delay. CSMA techniques do have the advantage that if nothing is currently happening on the LAN, a device may send immediately and doesn't have to wait (as it does in some other techniques). A disadvantage is that as LAN utilisation increases so access delay becomes highly erratic and (potentially at least) unbounded.

The big advantage of CSMA techniques is one of cost:

- The hardware adapters are very simple and low in cost.
- The cables typically used are low cost telephone twisted pair or CATV style coaxial cable.
- They usually run over bus-type networks which use less cable than ring or star topologies.

Token Passing (Token-Ring, Token Bus, FDDI)

A "token" (unique header containing control information) is sent from user to user around a "ring". Only the device with the token is allowed to send at a particular instant in time. "Block" multiplexing is used and blocks are limited in length by a time delay (maximum sending time) which is user specified.

In detail, the token passing protocols differ considerably, but the performance characteristics are as follows:

- The LAN can be utilised efficiently up to quite high capacities. Utilisation of 70% or even more can be achieved.
- Access is fair in the sense that all devices on the ring get an equal opportunity to use the LAN.
- It is possible to have a priority scheme such that, for example, real time traffic can be given priority over more normal data traffic. Even packetised voice may be handled in a limited way. The problems of voice and data mixture do not go away but there is considerable improvement over CSMA/CD.
- Ring techniques also suit fibre optical cables since it is difficult (possible but difficult) to treat optical fibre as a bus and attach many users to the common medium. Fibre technology is, in 1992, primarily a point to point unidirectional technology.
- Geographic length is less of a problem and it is now possible to have practical rings of thousands of miles in length. (There are practical limits on the length and the number of devices imposed by

imperfections in synchronisation ("phase jitter etc.") on the physical medium. On an electrical token-ring, LAN operation is considered problematic if the number of stations or repeaters goes above 250 or so.)

Two problems exist with the token passing approach:

1. There is an access delay due to "ring latency" between when a device has data to send and when it may start sending **even if there is no traffic**. This is because the device must wait for a token to arrive before it is allowed to send.
2. As the data transfer speed increases so does the length of the LAN (measured in bits). That is to say, the higher the speed of the LAN, the more bits can fit on it at one time. This means that on a reasonably sized LAN there would be room for more than one frame to be present simultaneously but that is not allowed because there is only a single token.

   The problem here is not that efficiency gets less but that there is an opportunity to become more efficient that the protocol cannot take advantage of.

The biggest problem with this method has been its cost. Since the token controls everything, there must be something to control the token and, for example, to handle the condition of errors occurring which put a permanently busy token onto the ring. Since it is considered vital that there be no "ring control unit" (which is obviously capable of failure and, therefore, has to be backed up, etc.) then each device attachment must be capable of ring control. There must be a mechanism to control which device is the ring controller when the ring is started and another to ensure takeover by one and only one other device if the current controller fails. All this takes logic and the cost has been significantly higher than for the CSMA technique. Recent improvements in chip technologies have minimised this cost differential however.

Insertion Rings (Metaring)

The principle of Metaring[52] is to allow a device to send anytime provided that no data is arriving on the LAN at the time that it starts sending. Thus, because it takes time for data to travel from one node to another, **multiple nodes can transmit at the same time**. This does not cause collisions because there is a buffering scheme that intervenes and prevents collisions from causing loss of data.

Metaring uses two counter rotating rings so that a control message may travel in the opposite direction to the data. This control message visits each device and essentially allocates LAN capacity (permission to send) among all the devices on the LAN.

This scheme has the following characteristics:

- As the link speed is increased and ring latency (in terms of the number of bits held on the ring at any one time) increases, the ring is able to handle more and more traffic.

[52] There are many kinds of insertion ring. One of the earliest was implemented on the IBM Series/1 computer in 1981. Metaring is a highly sophisticated version of an old principle.

- At relatively low speeds (say 16 megabits per second) the protocol could produce a ring latency that is too high for some applications but at speeds of 100 megabits and above this is much less of a problem.
- A fair access scheme is implemented using the control signal.
- There is very little access delay at low ring utilisation.
- The technique offers significantly higher throughput than FDDI for roughly the same cost.

Distributed Queueing (DQDB)

The distributed queueing protocol of DQDB[53] aims to provide fairness of access by having a device keep track (as far as it can) of the state of the notional global queue and its position in that queue.

The protocol uses two slotted busses to provide communication in both directions. The protocol is described in section 9.5, "DQDB/SMDS - Distributed Queue Dual Bus" on page 201.

The characteristics of this protocol are:

- The busses are managed in slots so that capacity may be allocated for constant rate traffic ("isochronous"- voice).
- Over relatively short distances the protocol provides excellent fairness of access to the busses. This breaks down a bit over longer distances at heavy loadings but can still be very effective.
- A single node can use the entire network capacity effectively.
- Data is sent in cells of 48 data bytes.
- There is no slot reuse so over long distances or at very high speed the maximum capacity is still only the speed of the bus.
- Both busses are used for data transport.
- There is very little access delay at low and medium LAN utilisations because data may be sent in the first free slot when there is nothing already queued downstream.

This technique is used in Metropolitan Area Network equipment currently being installed by many PTTs. It is also the basis of the access protocol called "SMDS".

Using a Ringmaster (CRMA)

The CRMA protocol uses a folded bus (similar to a ring) topology but has a ring controller node. The ring controller sends out a (preemptive) control message at short intervals. This message asks each node how much data has arrived on its queue since the last time it saw the message (cycle). Thus what it is really doing is taking a picture of the global queue at defined intervals.

This information enables the system to grant access to the LAN much more fairly than other protocols. In Figure 67 on page 172 it can be seen that data may arrive at each node in a somewhat random fashion. Protocols that grant equal access for each node (such as token passing protocols) will give access to a block of data that just arrived at a

[53] Distributed Queue Dual Bus

hitherto idle node *ahead* of data that may have been waiting in a queue at a busier node for some time. Thus for example, in token-ring protocol if six blocks arrive in quick succession at node A, and then a single block arrives at node B, the block at node B will get access to the LAN before some of the blocks queued at node A.

CRMA aims to allow access to the LAN for all traffic globally in FIFO order!

The characteristics of CRMA are:

- It gives the best fairness characteristic of any of the protocols being discussed.
- It will operate at almost any speed (the higher the better).
- It does not allow spatial reuse. This means that when data is received by a node, the cell that the data has been received from (and therefore is now logically empty) cannot be used by other stations to carry data. This is a waste of potential capacity.
- It avoids the problem of potentially high access delay due to ring latency by suspending the cyclic protocol at very low loads and allowing a device to send immediately when data becomes available.

9.2 Token-Ring

Many excellent descriptions of the detailed operation of token-ring protocol are available and so it will not be described in detail here. However, there are a number of important features of token-ring that need to be reviewed in order to understand why we need to use different protocols at very high speeds.

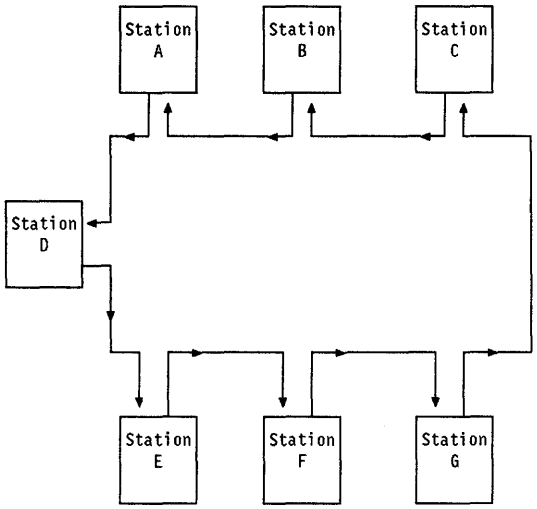


Figure 68. Conceptual Token-Ring Structure

Configuration

A number of devices called "stations"[54] are connected by a series of point-to-point links such that the whole forms a "ring". Data is sent in one direction only so that if station A (in the figure) sends data to station B then the data will pass through stations D, E, F, G and C *before* it is received by station B. Data links are usually electrical (although they may be optical) and the data rate is either 4 or 16 million bits per second.

A ring is usually wired in the form of a "star". A device called a Ring Wiring Concentrator (RWC), which may be active or passive, is used at the centre and two pairs of wires (in the same cable) are connected (point-to-point) from the RWC to each station. This is done so that individual stations may be added to or removed from the ring conveniently.

The RWC has an electromechanical relay in each station connector. This relay is held open by a direct current "phantom" voltage generated by the station. If a station loses power or is unplugged, the relay closes and bypasses that particular station and its connecting cable.

Data Transmission

Data is sent in blocks called "frames" which have a variable length and are preceded by a fixed format header. When station A wants to send data to station F it builds a header with its own address and the address of the destination (station F) in it and appends this header to the data.

[54] Sometimes also referred to as "nodes".

When station A sends the data it *always* goes to the next station on the ring (station D) this station repeats the data onto its outbound link.

The process proceeds around the ring until the data arrives at its destination station (F). Station F also sends the data onward around the ring but recognises itself as the destination address. In addition to propagating the frame on the ring, station F copies the frame into its input buffers.

Access Control

It seems obvious from the structure described above that only one station can transmit at any one time. The problem is how to control (with fairness) which station has permission to send at any one time.

This problem is solved by using a special header format called a "token". The token is sent from station to station around the ring. When a station receives the token it is allowed to send one frame of up to a specified maximum size (specified as a maximum transmission time). When a station finishes sending its frame it must send a free token to allow the next station a turn at sending.

Monitor Function

The token controls which station sends next but what controls the token? Errors can cause tokens to be lost or permanently marked as busy. Physical breaks can occur in the ring.

A special station (called a monitor station) is needed on the ring to control the token and to continually verify proper functioning. In addition this station generates the ring timing (clock) and equalises the accumulated timing error (jitter) etc.

In practice, *every* station has the monitor capability. When the ring starts up one station becomes the "active monitor". This is to remove the need for special nodes which would need backup in case of failure etc.

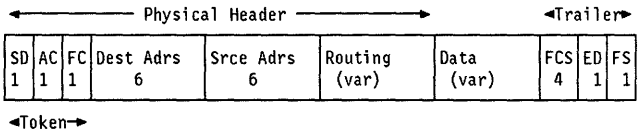


Figure 69. Token-Ring Frame Format. A token consists of the SD (Start Delimiter), AC (Access Control) and FC (Frame Control) fields alone. Numbers denote the field lengths in bytes.

The concept described above is very simple but a number of things must be done to make sure that the concept works in practice:

Minimise Delay in Each Station

In some early LAN architectures, a whole data frame was sent from one station to another and then retransmitted. This meant that the transmit time for the frame was added to the transit time **for every station that the message passed through**. This protocol actually works quite well where there are very few (for example, four) devices in the LAN. In a LAN of perhaps 100 devices the "staging delay" becomes critical.

One key component of TRN is its use of Differential Manchester Coding at the electrical level. This is discussed in section 2.2.11, "Differential

Manchester Coding" on page 28. Potentially, this enables a ring station to have only 1 bit of delay, but in current adapters this is 2½ bits.

The monitor station generates a clock and every other station derives its timing from the received data stream and uses this derived timing to drive its own transmitter. This is done to avoid having a large "elastic buffer" (delay) in each node. But there is a down side. Jitter (the very small differences in timing between the "real" timing and what the receiver is able to recover - see section 2.2.6.2, "Jitter" on page 21) adds up and after a while (240 stations) threatens to cause the loss of data. The monitor station (but only the monitor station) indeed contains an elastic buffer to compensate for jitter around the ring.

Structure of the Token

With only a one-bit delay in each station, how can a station receive the token and then send data? It might repeat the token to the next station before it had the opportunity to start transmission.

The technique used here relies on the fact that the token is a very short (24 bits) fragment of the beginning of a frame header. Bit 3 of the Frame Control field determines whether this is a token or the beginning of a frame. So, when a station has something to send it monitors for a token. When it detects the token it *changes* the token bit in the FC field to mark the token as "busy" and then appends its data.

Removing a Frame from the Ring

An analogy that has been often used in relation to token-rings is that of a railway train. The "train" goes from station to station around the ring. But this gives very much the wrong impression.

At 4 Mbps a single bit is around 50 meters long![55] A data block of (say) 100 bytes is 800 bits or 40 kilometers long! This is much longer than most LANs - so, in fact, the beginning of a data block arrives back at the sending station (usually) long before the station has finished sending the block! If this were a railway train on a loop of track, the front of the engine would arrive back at the start before the back of the engine left (not to worry about the carriages).

> In fact, there is a real problem with small token-rings. This is that a transmitting station could potentially receive the beginning of a token before it has finished transmitting this same token. This gives logical problems in the protocol. It is avoided by having a serial elastic buffer in the active monitor station that inserts sufficient delay to ensure that no ring can be shorter than a token (24 bits).

A frame could be removed from the ring by the destination station. But, a destination station does not know that it is the destination until most of the header has already been propagated around the ring. In any case there are broadcast frames to think about where there are multiple destinations. In addition, if the frame is left on the ring, we can use a bit in the trailer (set by the receiving station) to say "frame copied" to give a basic level of assurance to the transmitting station that somebody out there received the frame.

---

[55] Electricity travels on twisted pair media at about 5 µsec per kilometer.

So, the frame is removed from the ring by the sending station. In the 4 Mbps version of the IBM Token-Ring, after completing transmission of its frame, a sending station transmits idle characters until it receives the header from its transmitted frame. When it has completely received the frame header it releases a new token onto the ring so that the next station may have an opportunity to transmit.

In summary, after completing its transmission, the sending station waits only to receive the header of the frame it just transmitted (not the whole frame) before releasing a free token. However, it will still receive and remove the entire frame it transmitted (including checking the bits in the trailer).

When the ring speed is increased from 4 Mbps to 16 Mbps several things happen:

Data transfer is faster.

It takes less time to transmit a frame.

Staging delay in each node is less.

Delay stays at 2½ bits but a bit now takes ¼ of the time.

The speed of light (and of electricity) hasn't changed at all!

A major component of ring latency (propagation delay) is unchanged.

The bits are shorter.

At 16 Mbps a bit is 12.5 meters in length. (The railway train analogy begins to look a little more sensible!)

But (in the 4 Mbps version of the protocol) the sending station still waits until the header of the frame it has just transmitted is received before it places (releases) a new token onto the ring. The effect of this "gap time", where the transmitting station is sending idles while waiting for the header to be received, is small at 4 Mbps. At 16 Mbps the effect can be significant.

For operation at 16 Mbps the token-ring protocol is modified such that when a station finishes transmission it will immediately send a free token. This is called "early token release".

As ring speed is increased further the TRN principle will still operate but throughput does not increase in the same ratio as the link speed:

Latency

When a station has transmitted its one frame it must send a token to let the next station have a chance. It takes time for the token to travel to another station and during this time no station can transmit - the ring is idle. As ring speed is increased the transmission time becomes shorter but this latency (between transmissions) is unchanged. This means that in percentage terms latency becomes more significant as speed is increased.

The situation could be improved by allowing a station to send multiple frames (up to some limit) at a single visit of the token. (FDDI does just this.)

No "Guaranteed Bandwidth"

There is no mechanism available to guarantee a station a regular amount of service for time critical applications.

No "Isochronous" Traffic

Isochronous traffic (meaning non-packetised voice and video) is different from the guaranteed bandwidth real-time characteristic mentioned in the previous point. TRN does not allow this either.

Potentially Wasted Capacity

Only one active token may be on the ring at any one time. This means that only one station may transmit at any one time. In the case (in the diagram above) where station E is transmitting to station F, station C might perhaps transmit to station A thus doubling the throughput of the ring - if a protocol could be found that made this possible.

9.3 Fibre Distributed Data Interface (FDDI)

FDDI was developed by the American National Standards Institute (ANSI). It was originally proposed as a standard for fibre optical computer I/O channels but has become a generalised standard for operation of a LAN at one hundred megabits per second. In 1991 the FDDI standards are now firm and there are many FDDI devices available on the market. However, mass acceptance in the marketplace has yet to happen. The important characteristics of FDDI are as follows:

Optical LAN at 100 Megabits per Second

FDDI is primarily intended for operation over optical fibre but recently has been proposed for operation over standard copper wire (shielded twisted pair).

Dual Token Rings

There are two token rings operating in opposite directions. The primary ring carries data. The secondary ring is used to "wrap" the ring should the ring be broken. The secondary ring is not normally used for data traffic.

Ring Characteristics

Using multimode optical fibre for connection, an FDDI ring (segment) may be up to 200 km in length attaching a maximum of 500 stations up to two kilometers apart.

Frame (Packet) Switching

Like many other types of LAN (token-ring, Ethernet**...) data transfer takes place in frames or packets. In FDDI the maximum frame size is 4500 bytes. Each frame has a header which contains the physical address of the destination FDDI station.

Guaranteed Bandwidth Availability

In addition to the "equality of access" characteristic of token-ring, FDDI offers a form of "guaranteed" bandwidth availability for "synchronous"[56] traffic.

Token-Ring Protocol

The ring protocol is conceptually similar to the token-ring (IEEE 802.5) LAN but differs significantly in detail. FDDI ring protocol is dependent on timers; whereas, TRN operation is basically event driven.

Ring Stations

An FDDI station may connect to both rings or to only the primary ring. There may be a maximum of 500 stations connected to any one ring segment.

Ring Monitor

Like token-ring, there is a ring monitor function (only one of which may be active on the ring at any point in time). This monitor controls the

[56] In FDDI the word "synchronous" is used to mean "traffic which has a real time requirement". That is, to transmit synchronous traffic a station must gain access to the ring and transmit its frames within a specified time period. This is *not* the usual meaning of the word synchronous. See the description in Appendix C, "Getting the Language into Synch" on page 291.

operation of the ring and performs error detection and handling. A ring monitor station is normally connected to both rings.

Different from token-ring, each station does *not* need to have the ring monitor function.

The ring monitor is not a separate station, any FDDI station may contain a monitor function. When the ring is initialised, the monitors will negotiate and select one to be the active monitor.

9.3.1 Structure

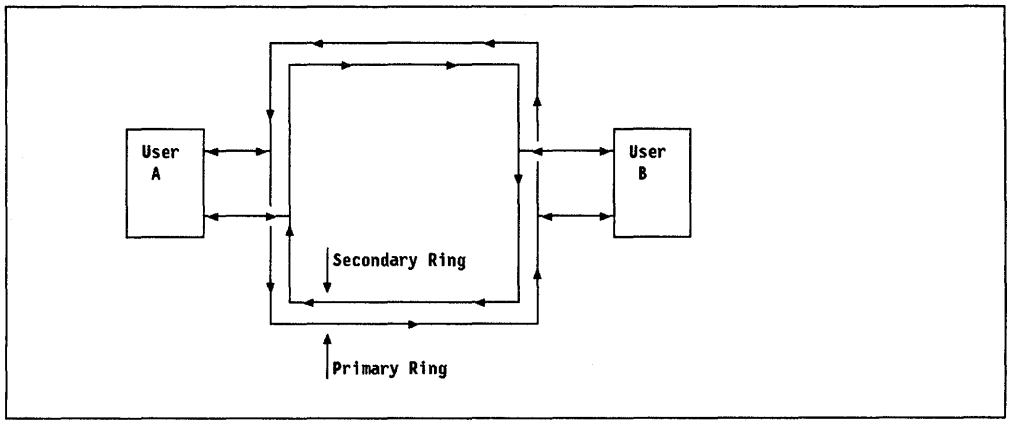


Figure 70. FDDI Basic Structure

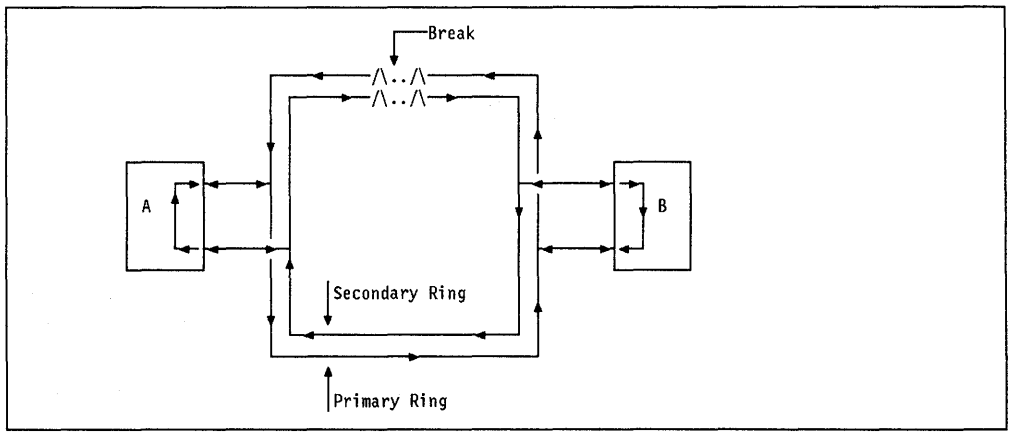


Figure 71. FDDI Ring Healing

Figure 70 shows the basic structure of an FDDI LAN. This consists of counter-rotating rings with the primary one carrying data.

Should there be a break in the ring, the stations can "wrap" the ring through themselves. This is shown in Figure 71. The secondary ring is used to complete the break in the primary ring by wrapping back along the operational route.

There are two classes of station:

Class A stations connect to both the primary and secondary ring and have the ability to "wrap" the ring to bypass error conditions. These are sometimes called Dual Attachment Stations (DAS).

Class B Stations connect only to the primary ring. This is to allow for lower cost (lower function) attachments. These are called Single Attachment Stations (SAS).

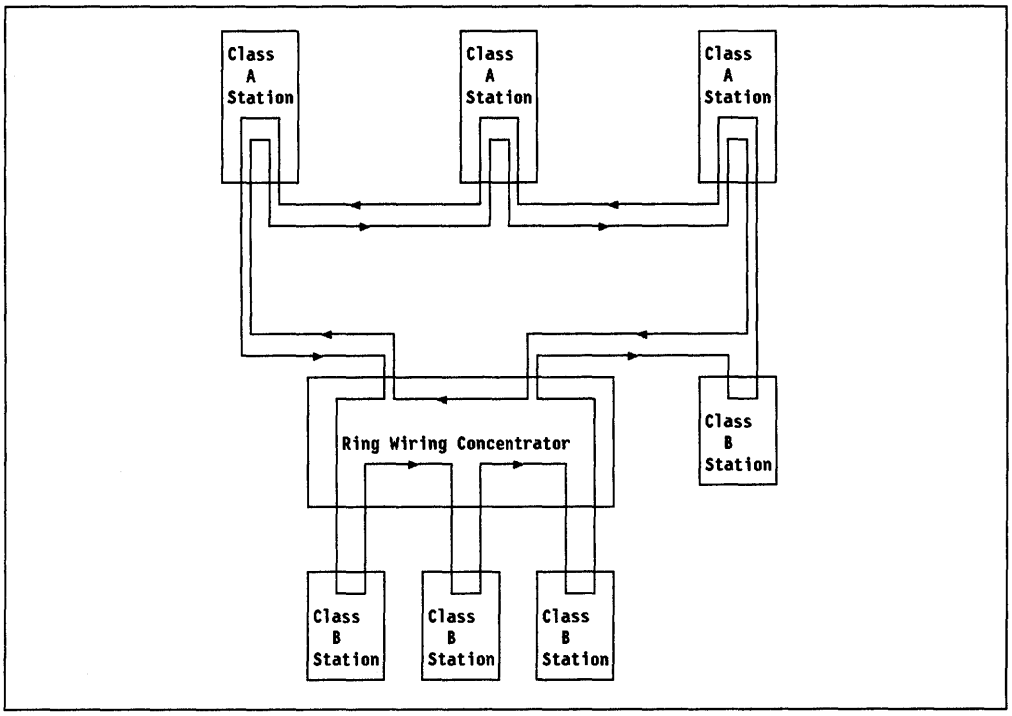


Figure 72. FDDI Ring Configuration

Figure 72 shows class A and B stations connected to a backbone FDDI ring. In addition a Ring Wiring Concentrator (RWC) is present which allows the connection of multiple class B stations in a "star wired" configuration.

The ring wiring concentrator could be a simple device which performs only the RWC function or it could be quite complex containing a class A station with the ring monitor function as well.

Some users choose to attach some SAS stations to the secondary ring. This allows the secondary ring to be used for data transport and of course gives higher aggregate throughput. However, unless there is a bridge between primary and secondary rings, an SAS station on the primary ring cannot communicate with an SAS station on the secondary ring.

9.3.2 Access Protocol Operation

Ring access in FDDI is controlled by a special frame called a token. There is only one token present on the ring at any one time. In principle when a station receives a token it has permission to send (place a frame of data onto the ring). When it finishes sending, it must place a new token back onto the ring.

FDDI is a little more complex than suggested above due to the need to handle synchronous traffic. There are three timers kept in each ring station:

Token Rotation Timer (TRT)

This is the elapsed time since the station last received a token.

Target Token Rotation Timer (TTRT)

This is a negotiated value which is the target maximum time between opportunities to send (tokens) as seen by an individual station. TTRT has a value of between 4 milliseconds and 165 milliseconds. A recommended optimal value in many situations is 8 milliseconds.

Token Holding Timer (THT)

This governs the maximum amount of data that a station may send when a token is received. It is literally the maximum time allocated to the station for sending during each rotation of the token.

When a station receives a token it compares the amount of time since it last saw the token (TRT) with the target time for the token to complete one revolution of the ring (TTRT).

- If TRT is less than the target then the station is allowed to send multiple frames until the target time is reached. This means the ring is functioning normally.

  TTRT - TRT = THT

- If TRT is greater than TTRT it means the ring is overloaded. The station may send "synchronous" data only.
- If TRT approaches twice TTRT there is an error condition that must be conveyed by the ring monitor function to the LAN Manager.
- This implies that each station may observe delays to traffic and thus must be able to tolerate these delays - perhaps by buffering the data.

When a station attaches to the ring, it has a dialogue with the ring monitor and it indicates its desired Token Rotation Time according to its needs for synchronous traffic. The ring monitor allocates an Operational Token Rotation Time which is the minimum of all requested TTRT values. This then becomes the operational value for all stations on the ring and may only be changed if a new station enters the ring and requests a lower TTRT value.

Within the asynchronous class of service there are eight priority levels. In token-ring a token is allocated a priority using three priority bits in the token - a station with the token is allowed to send frames with the same or higher priority. In FDDI the priority mechanism uses the Token Rotation Timers rather than a specific priority field in the token.

The sending station must monitor its input side for frames that it transmitted and remove them. A receiving station only copies the data from the ring. Removal of frames from the ring is the responsibility of the sender.

When a station completes a transmission it sends a new token onto the ring. This is called "early token release". Thus there can only ever be *one* station transmitting onto the ring at any one time.

In summary:

- A token circulates on the ring at all times.
- Any station receiving the token has permission to transmit synchronous frames.
- If there is time left over in this rotation of the token the station may send as much data as it likes (multiple frames) until the target token rotation time is reached.

- After transmission the station releases a new token onto the ring.
- Depending on the latency of the ring, there may be many frames on the ring at any one time but there can be only one token.
- The transmitting station has the responsibility of removing the frames it transmitted from the ring when they return to it.

9.3.3 Ring Initalisation, Monitoring and Error Handling

In an FDDI ring there is no single "ring monitor" function such as in an IEEE 802.5 token ring. This is because all stations on the ring perform part of the function cooperatively.

- The elastic jitter compensation buffer that exists in the active monitor of 802.5 does not exist, because every node regenerates the clock and there is no jitter propagation around the ring.
- All stations monitor the ring for the token arriving within its specified time limit.
- When the ring is initalised all stations cooperate to determine the TTRT value.
- When a break in the ring occurs all stations beacon but give way to any received beacon on their inbound side. In this way the beacon command that circulates on the ring identifies the immediate downstream neighbor of the break in the ring.

9.3.4 Physical Media

There are two types of media currently used for FDDI:

- Multimode fibre

  This is the originally defined mode of operation and the predominant mode of adapters on the market in 1991.

- Monomode fibre

  This has been included in the standard by ANSI but as yet has only minor usage.

In addition, there are other options under consideration by the ANSI committee:

- Shielded Twisted Pair (STP) copper wire

  The use of FDDI over STP cable, while it doesn't have the electrical isolation advantages of fibre, promises to be about half the cost of using FDDI on fibre. This could make FDDI an economic alternative for the desktop workstation. As yet the standard is not agreed.

- Unshielded twisted pair

  In 1991 this is also a proposed option. The proposal involves changing the data link encoding scheme and would thus require more extensive modification to the chip sets than for the STP proposal. There are many problems with the use of UTP as discussed in section 2.3.6, "LAN Cabling with Unshielded Twisted Pair" on page 41. Nevertheless, many users have installed low grade Telephone Twisted Pair (TTP) cabling for Ethernet connections. Many of these users now wish to use the same cable for FDDI.

- SDH/Sonet links

An FDDI structure could be operated over a wide area using channels derived from the public network. It has been proposed that channels derived from SDH/Sonet be used[57] to construct an FDDI ring over a wide area using public network facilities. The proposal is to map the full 125 Mbps rate into an STS-3c channel. This proposal has considerable support and (in 1991) looks likely to be accepted.

9.3.4.1 Media Specifications

FDDI is specified to use multimode fibre at a wavelength of 1,300 nanometers. The standard size is 62.5/125 but the other sizes of 50/125, 85/125 and 100/140 are optional alternatives. This means that an LED is used as the light source (rather than a laser) and that the detector is a PIN diode (rather than an avalanche photo diode).

The power levels are expressed in dBm.[58] They are:

```
Input:  −16 dBm
Output: -27 dBm
```

(Input and output here refer to the optical cable.) What this says is that an FDDI transmitter should transmit at a power level of −16 dBm and that a receiver should be able to handle a signal of −27 dBm. This means that you have 11 dB for loss in cables etc. If cable loses 3 dB per kilometer then if devices are two kilometers apart the cable loss will be 6 dB and there is 5 dB left over for losses in splices and connectors etc.

In practical terms (as discussed in section 3.1.2.11, "Fibre Cables" on page 55) cables vary in their losses and loss varies by temperature. Calculation of maximum allowable distance is something that needs to be done carefully, in conjunction with the cable manufacturer's specifications.

On another point of practicality. Some devices on the market actually transmit at a higher power level than that specified here. In addition. it is easy to overload a PIN diode receiver if the power level is too high. This means that if FDDI stations are installed close together (for example, in the same room), attenuators may be needed in the cable to cut down the light to a level acceptable to the receiver.

9.3.4.2 Optical Bypass Switch

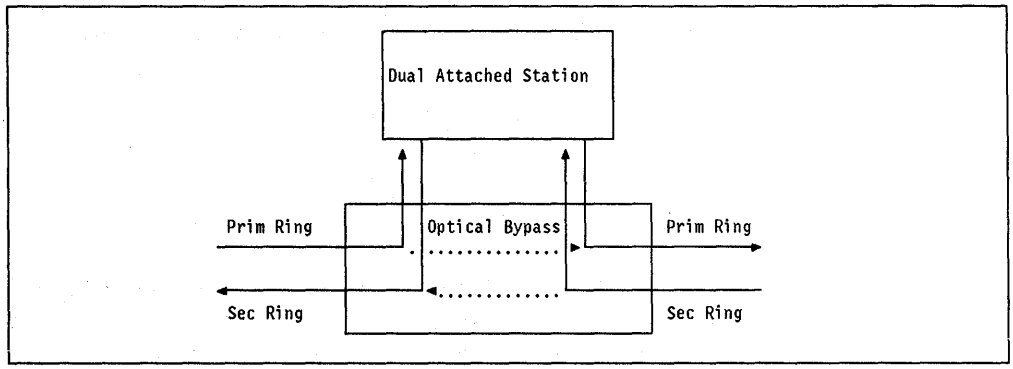


Figure 73. Optical Bypass Switch

[57] See section 6.2, "SDH and Sonet" on page 117.

[58] This is a measure of absolute power - decibels per milliwatt.

Optical bypass switches may be used either built into the station or as separate devices to maintain connectivity of the rings when power is turned off or when the node fails.

These switches are mechanical devices (switching is mechanical but operation could be electrical) usually operating by moving a mirror. They depend on mechanical movement being precise. Since mechanical operations cannot be exactly precise, they introduce additional loss to the ring even if the station is operating correctly. This limits further the possible distance between nodes.

9.3.5 Physical Layer Protocol

The basic functions of the physical layer are:

1. To transport a stream of bits around the ring from one station to another.
2. Provide access to the ring for each individual station.

To do this it must:

- Construct a system of clocking and synchronisation such that data may flow around the ring.
- Receive data from a station and convert it into a form suitable for transmission.
- Receive data from the ring and convert it into the form expected by the node access protocol.
- Provide a transmission system that allows the station to send and receive any arbitrary bit stream (transparency).
- Signal the station (node access protocol) at the beginning and end of every block of data.
- Keep the ring operational and synchronised even when there is no data flowing.

9.3.5.1 Ring Synchronisation

In FDDI, each ring segment is regarded physically as a separate point-to-point link between adjacent stations. This means that the exact timing of data received at a station *cannot* be the same as the timing of data transmitted. Since it is not possible to build (at an economic cost) oscillators that are exactly synchronised there will be a difference between the data rate of bits received and that of bits transmitted! This is solved by the use of an "elasticity buffer".

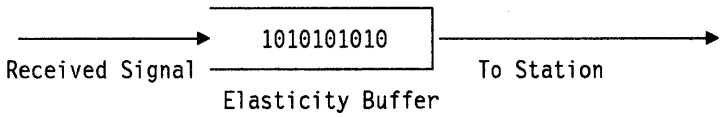


Figure 74. Ten-Bit Elasticity Buffer Operation

Most of the time a station simply passes data on around the ring. The need is to pass this data with minimal delay in each station. This means that we need to start transmitting a block towards the next station *before* it is completely received. The received signal arrives at the rate of the upstream station's transmitter. When data is sent onward to the next station in the ring it is sent at the rate of this station's local oscillator. So data is being received at a different rate from the rate that it is being transmitted!

This is handled by placing an "elastic buffer" between the receiver and the input port to the ring station. The ring station is then clocked at the rate of its local oscillator (that is, the transmit rate).

The FDDI specification constrains the clock speed to be ± .005% of the nominal speed (125 megahertz). This means that there is a maximum difference of .01% between the speed of data received and that of data transmitted.

When a station has no data to send and is receiving idle patterns the elasticity buffer is empty. When data begins to arrive the first 4 bits are placed into the buffer and nothing is sent to the station. From then on data bits are received into the buffer and passed on out of the buffer in a FIFO manner.

If the transmit clock is faster than the receive clock then there are (on average) 4.5 bit times available in the buffer to smooth out the difference. If the receive clock is faster than the transmit clock there are 5 bit positions in the buffer available before received bits have to be discarded.

This operation determines the maximum frame size:

(4.5 bits / .01%) = 45,000 bits = 9,000 symbols = 4,500 bytes

A 16-bit idle pattern is sent after the end of every frame (between frames) so that the receiver has time to empty its elasticity buffer if necessary before the arrival of another frame.

While this mechanism introduces additional latency into each attached station, it has the advantage that it prevents the propagation of code violations and invalid line states.

9.3.5.2 Data Encoding

Each four data bits is encoded as a five-bit group for the purposes of transport on the ring. This means that the 100 Mbps data rate is actually 125 Mbps when observed on the ring itself. Figure 75 on page 191 shows the coding used. This provides:

- Simplification of timing recovery by providing a guaranteed rate of transitions in the data. Only code combinations with at least two transitions per group are valid. This helps in the recovery of timing by a receiver.
- Transparency and framing. Additional unique code combinations (5-bit codes that do not correspond to any data group) are available. These are used to provide transparency by signaling the beginning and end of a block.
- Simplification of circuitry. One of the serious problems of high speed communications is that electrical technology becomes more and more costly as speed is increased. The electrical circuits that interface to the line must run at the line rate. However, if bits are received in groups then they can be passed to other circuits in parallel at a much slower rate.

  In Figure 76 on page 192 all the functions below the 4B/5B encode/decode box must run at 125 megahertz. But the access protocol gets data passed to it in 4-bit groups at only 25 megahertz.

  This means that the ring interfacing functions might be implemented in (relatively expensive) bipolar technology and the rest in inexpensive CMOS.

	Symbol	Code Group	Meaning
Line State Symbols	I	11111	Idle
	H	00100	Halt
	Q	00000	Quiet
Starting Delimiter	J	11000	First byte of SD
	K	10001	Second byte of SD
Control Indicator	R	00111	Logical Zero (reset)
	S	11001	Logical One (set)
Ending Delimiter	T	01101	Terminates data stream
Data Symbols	0	11110	B'0000' (Hex '0')
	1	01001	B'0001' (Hex '1')
		...	
		...	
		...	
	9	10011	B'1001' (Hex '9')
	A	10110	B'1010' (Hex 'A')
	B	10111	B'1011' (Hex 'B')
	C	11010	B'1100' (Hex 'C')
	D	11011	B'1101' (Hex 'D')
	E	11100	B'1110' (Hex 'E')
	F	11101	B'1111' (Hex 'F')
Invalid Symbols	V	00001	Invalid
	V	00010	..
		...	..
		...	..

Figure 75. 4B/5B Coding as Used with FDDI. Four bit groups are sent as 5-bit combinations to ensure there are at least two transitions per group.

9.3.5.3 NRZI Modulation

The bit stream resulting from the above encoding is further converted before transmission by using "Non Return to Zero Inverted" (NRZI) procedure. This adds more transitions into the data stream to further assist with timing recovery in the receiver. In NRZI procedure, a "1" bit causes a state change and a "0" bit causes no state change.

A sequence of IDLE patterns (B'11111') will result in a signal of 010101 thus maintaining synchronisation at the receiver. Some valid data sequences (for example X'B0' coded as B'10111 11110') can contain up to 7 contiguous "1" bits and these need to have additional transitions if the receiver is to synchronise satisfactorily.

The net effect of 4B/5B encoding and NRZI conversion is that the maximum length of signal without a state change is 3 bits.

9.3.5.4 Physical Layer Operation

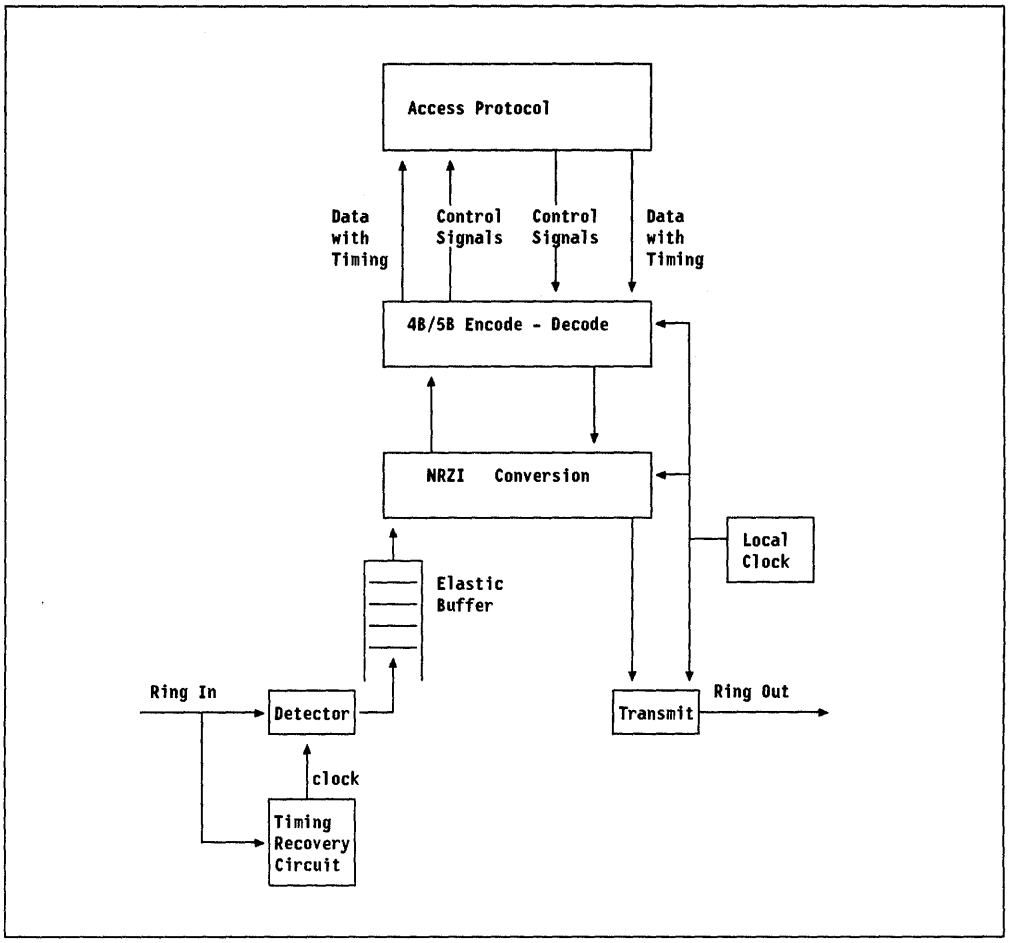


Figure 76. Physical Layer Structure

Figure 76 summarises the operation of the physical layer(s) of FDDI.

9.3.5.5 Physical Level Comparison with Token-Ring

In some discussions of FDDI a comparison is drawn with token-ring and the point made that since FDDI uses a data rate of 125 Mbps to send a data stream of 100 Mbps, and token-ring uses two signal cycles per bit (16 Mbps is sent at 32 Mbps), that therefore token-ring is somehow "inefficient" by comparison. Nothing could be further from the truth.

Because FDDI is intended primarily to operate over a multimode fibre connection, physical level operation was designed for this environment and is therefore quite different from the electrical operation of token-ring.

In the token-ring architecture, a major objective is to minimise delay in each ring station. This is achieved by having only a single bit buffer in each station for the ring as it "passes by". Operation with such a short delay requires that the output data stream be *exactly* synchronised (both in frequency and in phase) with the input data stream.

There are two problems here:

1. Simple identification and recovery of the data bits.

This requires fairly simple circuitry and usually takes the form of a Digital Phase Locked Loop (DPLL).

2. Reconstructing the exact timing of the incoming bit stream.

   This means that a new timing signal must be constructed as nearly as possible identical with the signal that was used to construct the received bit stream. To do this requires a very complex analogue phase locked loop.

This then is one of the major reasons for using the Manchester code for TRN. Because of the guaranteed large number of state transitions in the code, the recovery of accurate timing information is much easier and the necessary circuitry is simpler and lower in cost.

On an optical link data is sent as two states: light or no light. Recovering an accurate clock is more difficult here (especially on multimode fibre). At the speed of FDDI, there is less need to minimise buffering in the node. Also, because of the "early token release" protocol, there is much less loss of efficiency due to node delay.

9.3.6 Node Structure

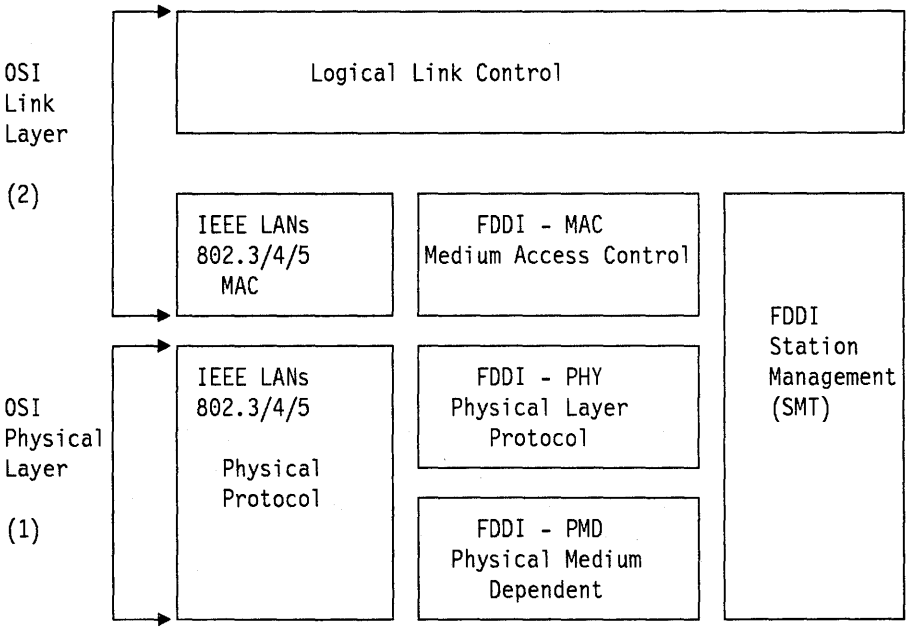


Figure 77. FDDI Node Model

Figure 77 shows the theoretical model of FDDI compared to the IEEE LANs (token-ring, Ethernet...). Their relationship to the OSI model is shown on the left.

It is assumed by FDDI that IEEE 802.2 logical link control will be used with FDDI but this is not mandatory.

The FDDI standard is structured in a different way to the others. Station management is not a new function. Most of its functions are performed for example by token-ring, but the functions are included in the physical and MAC components. Also, the physical layer is broken into two to facilitate the use of different physical media.

The functions of the defined layers are as follows:

Physical Medium Dependent Layer (PMD)

- Optical link parameters
- Cables and connectors
- Optical bypass switch
- Power budget

Physical Layer Protocol (PHY)

- Access to the ring for MAC
- Clocking, synchronisation, and buffering
- Code conversion
- Ring continuity

Media Access Control

- Uses token and timers to determine which station may send next.
- Maintains the timers.
- Generate and verify the frame check sequence etc.

Station Management (SMT)

- Ring Management (RMT)

  This function manages ring operation and monitors the token to ensure that a valid token is always circulating.

- Connection Management (CMT)

  This function establishes and maintains the physical connections and logical topology of the network.

- Operational Management

  This function monitors the timers and various parameters of the FDDI protocols and connects to an external network management function.

9.3.7 High Speed Performance

Compared with the 16 Mbps token-ring, of course FDDI data transfer (at 100 Mbps) is much faster. Ring latency is, however, another matter. Propagation speed is much the same.

The delay in a TRN node is around two bits. In an FDDI node the delay will depend on the design of the particular chip set but it is difficult to see how the delay could be less that about 20 bits. This means that an FDDI ring will have a longer latency than a 16 Mbps token-ring of the same size and number of stations.

FDDI follows the same "early token release" discipline as 16 Mbps token-ring but still only one station may transmit at any one time.

Token Holding Time (THT) is the critical factor. If THT is relatively short, then a station may only send a small amount of data on any visit of the token. If it is set large then a station may transmit a lot of data. Since it can take a relatively long time for the token to go from one station to another, during which no station may transmit, the longer the THT the greater the data throughput of the ring.

A short THT means that "ring latency" can be relatively short so that the delay for a station to gain access to the ring is also short. A short THT therefore is suitable for support of real time applications. If the THT is very short the system gives better response time but low overall throughput. If it is set very long, then you get a high throughput but a relatively poor response time.

The key tuning parameter is the "Target Token Rotation Time" (TTRT). At ring initalisation all stations on the ring agree to the TTRT (the shortest TTRT requested by any node is adopted). Stations then attempt to meet this target by limiting their transmissions. TTRT is a parameter which may be set by system definition in each node.

Work reported by Raj Jain (referenced in the bibliography) suggests that a value of eight milliseconds is a good compromise in most situations.

Of course, there may be only one token on the ring at any time and only one station may transmit at any time. In a long ring (a large number of stations and/or a great geographic distance), this represents some wasted potential.

9.4 FDDI-II

FDDI-II is an extension of FDDI that allows the transport of synchronised bit streams such as traditional (not packetised) digital voice or "transparent" data traffic across the LAN. This is called "isochronous" traffic.[59]

Most varieties of LAN (Ethernet, TRN etc.) can handle voice traffic if an appropriate technique of buffering and assembly into packets is used. FDDI is the best at this because its timed protocol allows a station to get access at relatively regular intervals. However, these protocols are primarily data protocols and cannot provide transparent carriage for an isochronous bit stream such as unpacketised voice.

The key to understanding FDDI-II is the fact that the FDDI protocols are used *unchanged* but travel within one channel of a time division multiplexed frame. Isochronous traffic (voice etc.) is handled by the TDM quite separately from the FDDI data protocol. Looked at from the viewpoint of what happens on the LAN cable itself, FDDI and FDDI-II are utterly different.

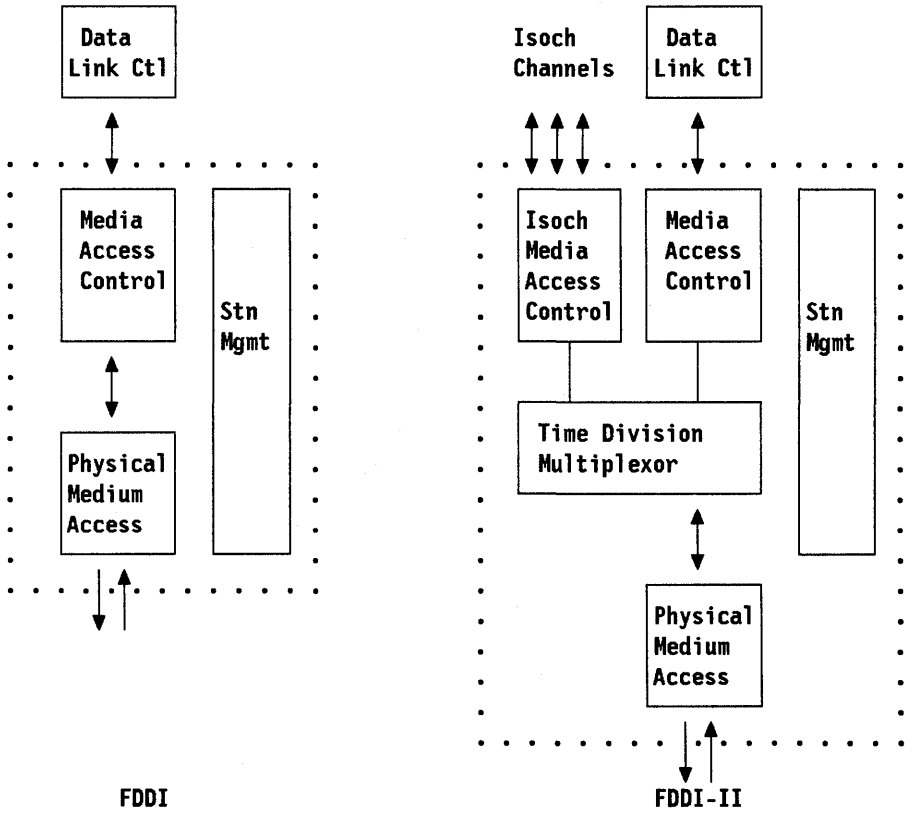


Figure 78. FDDI and FDDI-II Conceptual Structure. In FDDI-II a time division multiplexor is used to divide the LAN into multiple bit streams. For data transport one of the bit streams is mediated using FDDI access control protocols unchanged.

Figure 78 shows a highly conceptualised structure of the FDDI and FDDI-II nodes. The difference is that a time division multiplexor has been placed between the

[59] It is very easy to get confused here by the terminology. See Appendix C, "Getting the Language into Synch" on page 291 for more information on the use of the words "synchronous", "isochronous" etc.

FDDI media access control and the physical layer protocol. In addition there is an "isochronous medium access control" which provides continuous data rate services.

9.4.1 Framing

Like most voice oriented TDM systems (ISDN_P, DQDB, SDH/Sonet etc.), FDDI-II uses a fixed format frame that is repeated every 125 µsec.[60] Each 125 µsec frame is called a "cycle". At 100 Mbps each cycle contains 1560 bytes plus a preamble. The ring may operate at different speeds but can only change in 6.144 Mbit increments (because of the frame structure).

Each cycle has four components:

Preamble

When the frame is sent by the cycle master it is preceded by five idle symbols[61] (20 bits). This is used as a buffer to account for differences in clock speeds between the received data stream and a ring station's transmitter. Subsequent stations may vary the length of the preamble. This principle is discussed more fully in section 9.3.5.1, "Ring Synchronisation" on page 189.

Cycle Header

The cycle header is 12 bytes long and consists of the following fields:

- Starting Delimiter (8 bits)
- Synchronisation Control (4 bits)
- Sequence Control - C2 (4 bits)

  When the ring is operating in hybrid mode every 125 µsec cycle carries a sequence number. The sequence control field indicates whether the sequence number field is valid or not.

- Cycle Sequence - CS (8 bits)

  This is just the sequence number of this cycle. During initialisation the field is used for other purposes.

- Programming Template (64 bits - 16 symbols)

  Each 4-bit symbol in the programming template corresponds to one wideband channel. Only two states are valid. One state indicates that the corresponding wideband channel is used for packet data and the other state indicates that this WBC is in use for isochronous traffic.

- Isochronous Maintenance Channel (8 bits)

  This is a single 64 Kbps isochronous channel which is available for maintenance purposes.

Dedicated Packet Group

Twelve bytes in each cycle are concatenated to form a single 768KB channel. This is to ensure that even if all the wideband channels are in

---

[60] One frame per 125 µsec equals 8000 frames per second. If a single slot is 8 bits then this gives 64 Kbps - the speed of "standard" digitised voice.

[61] Because FDDI-II uses 4/5 code for sending on the link FDDI defines fields in terms of four bit "symbols".

use for isochronous traffic then there will still be some remaining capacity for data packets.

Wideband Channels (WBC)

At 100 Mbps there are 16 wideband channels each carrying 96 bytes in each cycle. This means that each wideband channel has an aggregate data rate of 6.144 Mbps (96 x 64 Kbps). This is the same rate as a "T2" channel in the US digital TDM hierarchy.

Each WBC may be allocated to either packet data or isochronous service. A WBC must be wholly dedicated to either mode of operation.

The WBCs and the DPG are byte interleaved with one another within the frame.

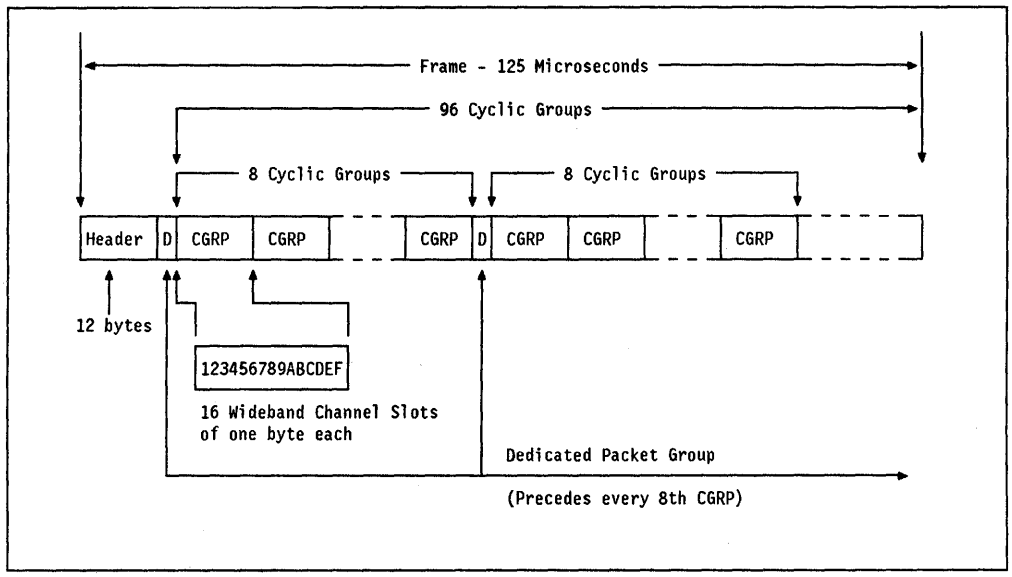


Figure 79. FDDI-II TDM Frame Structure

There is only one packet data channel. When one or more wideband channels are allocated to packet data they are concatenated with each other and with the dedicated packet group to form a single continuous bit stream. This continuous bit stream is recovered and reconstructed at every node. The FDDI protocol is used to operate this channel exactly as though it was the only bit stream on an ordinary FDDI ring.

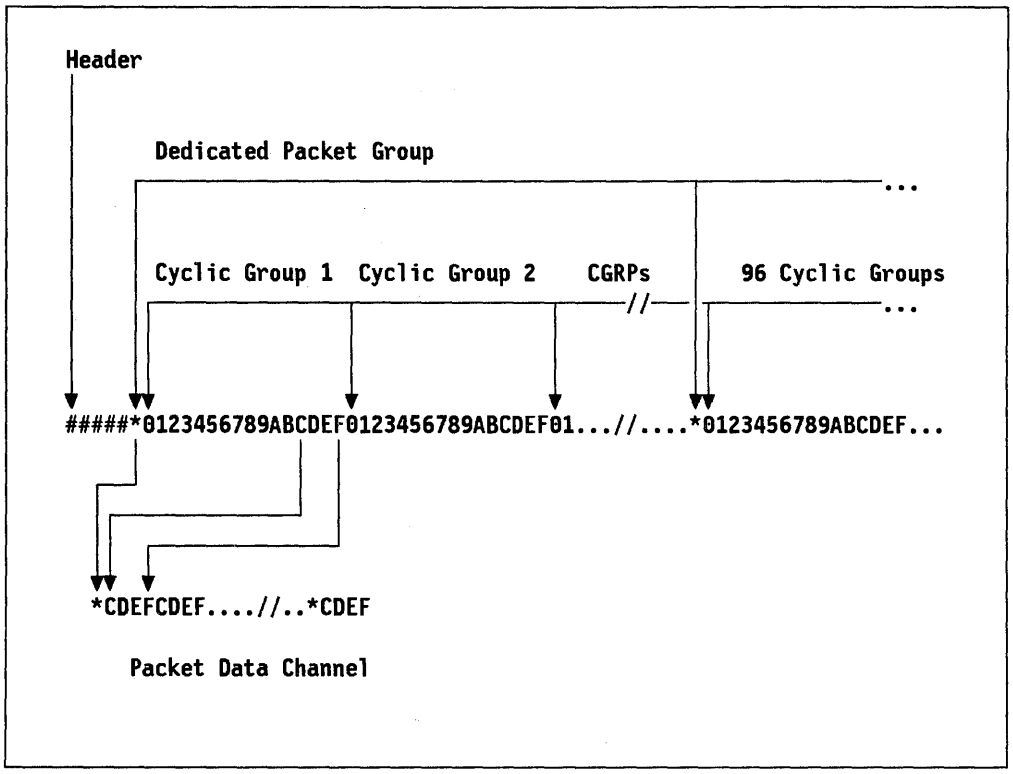


Figure 80. Derivation of Packet Data Channel from the TDM Frame. In this example WBCs 12 to 15 (labelled C, D, E and F) are concatenated with the DPG to form the packet data channel.

Figure 80 shows the packet data channel as it exists within the TDM frame. In the example four WBCs (labeled C to F) are concatenated with the DPC to form a single contiguous "clear channel".

Notice that each DPG contains a single byte from each WBC. So there are 96 DPGs in each frame. Because the frame rate is 8000 per second, (frame is 125 µsec long) each byte represents a rate of 64 Kbps.

9.4.2 Cycle Master

In hybrid mode a station called the "cycle master" generates 125 µsec frames (called cycles), assures cycle integrity and contains a latency adjustment buffer to ensure that there is always an integral multiple of cycles on the ring. A cycle master is an FDDI monitor station (and therefore it must be connected to both rings and it must contain the FDDI Monitor function) with the additional capability of generating, controlling and handling errors for the TDM mode of operation. It is not necessary for every ring station in an FDDI ring to be capable of being a monitor or a cycle master. However, for the ring to operate in hybrid mode (FDDI-II mode) every connected station must be capable of FDDI-II operation (handling the TDM frame structure).

9.4.3 Operation

Initialisation

The FDDI-II ring is initialised in "basic mode". Basic mode is the name given in FDDI-II for regular FDDI. This is used to set up the timers and parameters for the FDDI data protocol.

If every active station is capable of operating in "hybrid mode" then after initialisation, the ring may switch its operation into this mode. In hybrid mode the "cycle master" station creates and administers the TDM structure thus making the isochronous circuit switched service available.

Changing the Channel Allocations

The cycle master station may change the channel allocations (change a WBC from isochronous operation to packet mode or vice versa) at almost any time.

When a station wants to set up an isochronous circuit its management function sends a request to the management function in the cycle master station. As a result of that request the cycle master may need to allocate another WBC to isochronous operation.

In order to change modes without disrupting anything the cycle master waits until it has the token. This means that no data traffic is using the WBCs. The cycle master then changes the programming template in the cycle header to reflect the new status. Stations inspect the cycle header in each cycle to demultiplex the contents of that cycle, so the allocations change in the very next cycle.

Communication between Ring Stations and the Cycle Master

Ring stations exchange control messages with the cycle master and with each other by sending packets on the packet data channel. (Hence the need for the dedicated packet group to allow some packet communication even if all the WBC capacity is allocated to isochronous traffic.)

This capability is used, for example, in the set up of an isochronous channel.

1. The requesting station sends a request to the cycle master for a channel allocation.
2. The cycle master allocates the slot(s) and notifies the requesting station of the allocation.

   The cycle master *may* need to change the allocation of WBCs between the data packet channel and isochronous service to satisfy the request.
3. The requesting station then uses the packet data channel to send a request to the destination station to set up a circuit. The requesting station must tell the destination station the slot allocations provided by the cycle master.

Isochronous Data Transfer

The cycle master may allocate part (or all) of a WBC in units of a single byte slot for use by a station.

Once capacity is allocated by the cycle master, it is the responsibility of the requesting station (and the destination station) to agree on how this capacity will be used.

Error Processing

Error handling for the TDM frame structure as well as for the packet channel is handled by the cycle master.

9.5 DQDB/SMDS - Distributed Queue Dual Bus

DQDB protocol is the basis for the internal operation of a number of Metropolitan Area Networks (MANs) currently being installed in many countries. It is also the basis of a proposed US standard for user access to a MAN called SMDS.

The DQDB protocol was designed to handle both isochronous (constant rate, voice) traffic and data traffic over a very high speed optical link.

9.5.1 A Protocol by Any Other Name...

DQDB is also known by several different names and each has a different connotation.

QPSX

When the protocol was invented (by two people at the University of Western Australia) it was named QPSX (Queued Packed Synchronous eXchange). Later when a company was set up to build equipment based on the protocol, that company was called "QPSX Communications".

DQDB

Because the company was called QPSX the name of the protocol was then changed to "DQDB" to avoid confusion with the company name.

IEEE 802.6 MAN Subnetwork

In 1990, this protocol was accepted by the Institute of Electrical and Electronic Engineers (IEEE) as a standard for metropolitan area subnetworks and numbered IEEE 802.6.

SMDS

"Switched Multi-Megabit Data Service" is the name given to the service based on DQDB in the United States. DQDB is used as an **access protocol** for a high speed packet network. Minor changes were made to the IEEE 802.6 recommendation to enable it to fulfill this role.

It should be noted that networks using SMDS as their access protocol are not constrained to use 802.6 internally. Some networks will but others may work quite differently.

CBDS

Connectionless Broadband Data Service is the service name given to networks using QPSX equipment in Europe.

Fastpac

Fastpac is the service name given to the network service by Telecom Australia who are implementing the first fully commercial (tariffed) network to use the QPSX technology in the world.

9.5.2 Concept

A DQDB MAN is in many ways just like any other LAN. A number of stations (nodes) are connected to a common medium which allows data to be sent from one node to another. Because of the use of a shared medium there must be an access protocol to control when a node is allowed access.

Data is transmitted in short "segments" (cells) of 48 data bytes.[62]

Different from most LANs it is intended for operation over a wide geographic area and at bit rates of up to 155 Mbps. Another difference from many LANs is that it is also designed to handle isochronous (for example, non-packetised voice) traffic.

9.5.3 Structure

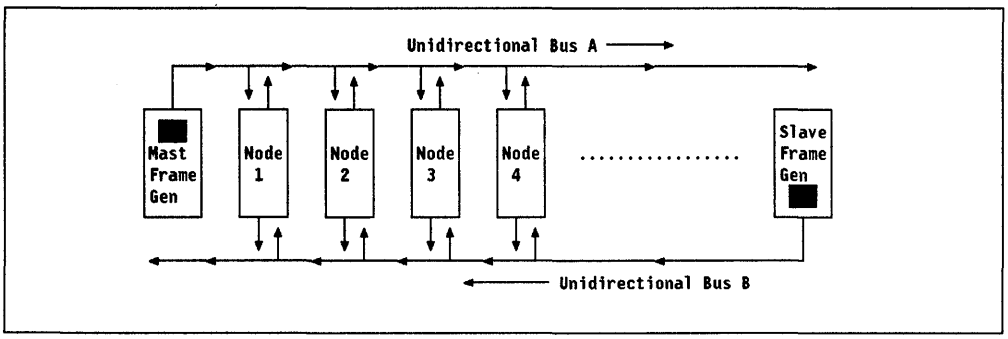


Figure 81. DQDB Bus Structure

- As shown in Figure 81 a DQDB network consists of two busses denoted by Bus A and Bus B in the figure.
- The two busses transport data in opposite directions.
- At the head end of each bus a slot generator creates a timed signal and formats it into 53-byte slots. The slot format is shown in Figure 82 on page 203.
- Each node is connected to *both* busses.
- When a node wants to transmit data it does so into the first available empty slot traveling in the desired direction. Slot availability is determined by the Medium Access Control protocol which all nodes must obey.

  The node must know the relative location (upstream/downstream) of the other connected nodes so that it can determine on which bus to send the data. In Figure 81 if node 2 wants to transmit data to node 4 then it must use bus A. To send data to node 1 it would use bus B.
- Data is never removed from the bus. The busses are terminated electrically and slots "drop off the end".
- There is no scheme for reuse of slots after data has been copied from them. For example if node 1 sends data to node 2 on bus A then potentially the slot could be reused by node 3 to send to node 4. This is not possible in DQDB at the present time.[63]

---

[62] Notice this is the same length as for Asynchronous Transfer Mode (ATM).

[63] An enhancement aimed at developing an "eraser node" which could allow slot reuse is under consideration by the IEEE 802.6 committee.

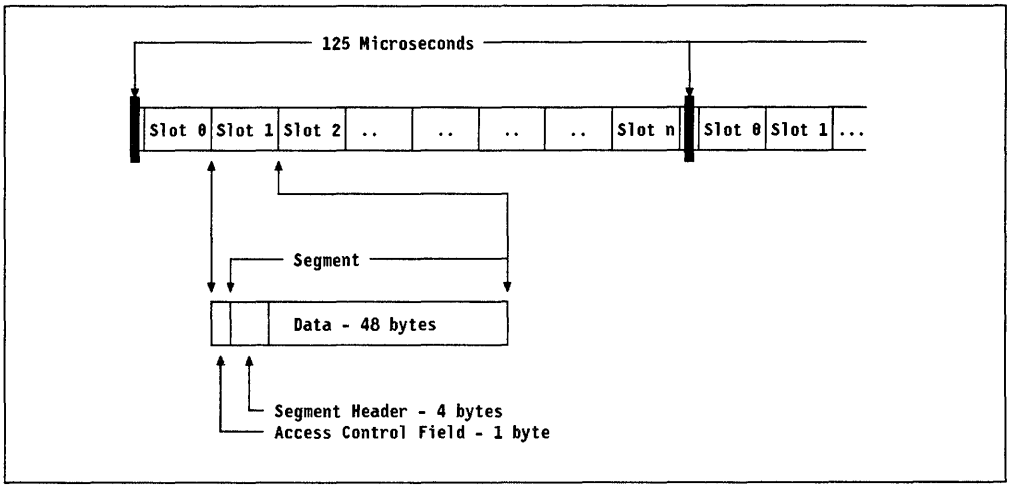


Figure 82. DQDB Frame Format as Seen by the DQDB Protocol Layer. Not shown is a 4-byte slot prefix used by the physical layer.

The framing structure is shown in Figure 82. At the physical layer, preceding each slot there is a 4-byte field consisting of a 2-byte slot delimiter and 2 bytes of control information, which are used by the layer management protocol. The slots are maintained within a 125 µsec frame structure so that isochronous services can be provided.

There are two types of slot:

Pre-Arbitrated (PA) Slots

These slots are assigned to a specific node by the frame generator. Thus they are called "Pre-Arbitrated". The frame generator will generate these at specified rates (for example one every 125 µsec) for use by isochronous traffic. They are ignored by the distributed queue medium access procedure.

The number and timing of these slots is variable depending on how much isochronous traffic is being carried. Remaining capacity is formated as Queued Arbitrated (QA) slots.

Queued Arbitrated (QA) Slots

These slots carry normal data traffic and are allocated through the "distributed queue" MAC procedure. (In FDDI terminology this is asynchronous data traffic - DQDB does not handle synchronous traffic in the FDDI sense.)

9.5.4 Medium Access Control

In light load situations, or in situations where each node was unable to transmit at a rate close to that of the bus, a strategy that said "send data into the next available slot" could work quite well. In the real world a system like that would not be practical because a single upstream node (close to the frame generator) could take 100% of the bus capacity leaving downstream nodes unable to gain access for any purpose. A means is needed to control when a node is allowed to send. This is the heart of the DQDB system. **Each node keeps track of as much of the global queue as it needs to determine its position in that queue.**[64]

[64] See section 9.1.2, "Access Control" on page 172 for an explanation of the notion of a global queue.

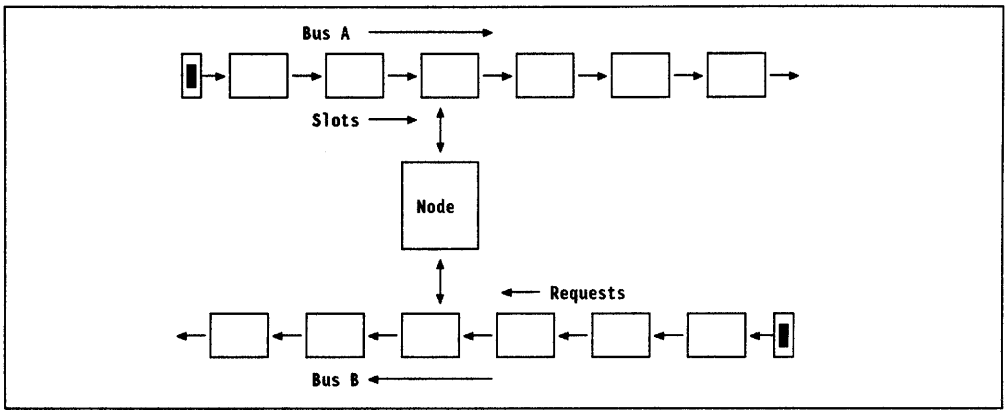


Figure 83. DQDB Medium Access Principle

The protocol works in the following way:

The protocol is full-duplex and it is also symmetric. Node access to Bus A is arbitrated (controlled) by requests flowing on Bus B. Node access to Bus B is arbitrated by requests flowing on Bus A. Both processes happen simultaneously and operate completely independently from one another.

Considering only Bus A:

- When a node has nothing to send it monitors the bus to keep track of the queue of data waiting.

  This is done by counting requests for slots from downstream nodes (requests flow on Bus B) and cancelling a request every time an empty slot (that would satisfy a request) passes by (on Bus A).

- If some data arrives to be sent, the node looks to see if any other node (downstream of itself) has requested, but not yet received, a slot. (How it does this will be discussed in a moment).

- The node sends a request on Bus B to tell nodes upstream of itself that it needs a slot.

- If there are no requests pending from nodes downstream of itself, the node may send into the next empty slot that arrives on Bus A.

- If there were already pending requests from downstream, it monitors Bus A and allows sufficient empty slots to pass to fulfill all of the pending requests from downstream nodes.

- When all the requests for slots that were pending when data arrived have been satisfied, then the node is allowed to send into the next empty slot.

- Now, the node probably wants to send again and hence must have kept track of what happened on the bus while it was waiting to send the previous segment.

  A node is not allowed to make multiple requests (not allowed to have more than one request outstanding at any one time).

To make this work, the node keeps two counters (for each bus independently).

Request Counter

When the node has nothing to send it monitors the bus.

- The request counter for Bus A is incremented (increased by one) every time a slot passes by on Bus B with the request bit set.

- The request counter is decremented (decreased by one) every time an empty slot passes by on Bus A.
- The request counter is *never* decremented below zero.

In this way a node knows at any instant in time, the number of pending downstream requests for slots.

Waiting Counter

When the node wants to send and there are pending downstream requests it must wait until sufficient empty slots have passed by to satisfy all pending downstream requests. To do this it keeps a Waiting Counter.

When the node wants to send, it:

- Copies the request count into the waiting counter.
- Sends a request upstream by setting the request bit in the next available slot on Bus B (provided it is not already set).

The request counter continues to operate normally.[65] As empty slots pass on Bus A now both request and waiting counters are decremented as empty slots pass.

When the waiting counter reaches zero, the node may send into the next empty slot. Notice now that the request counter still contains an accurate status of requests queued from downstream.

This is ingenius. Each individual node does not (and cannot) know the true status of the global queue. All it knows is the state of requests from itself and from nodes downstream of itself. Yet, the total system will allocate slots in the exact order in which they were placed.

What the node really does is to keep track of its position in the order of requests from itself and from nodes downstream of itself on the bus.

It must be emphasised that the protocol operates completely separately and independently for each direction of data traffic on the busses. Thus there are two sets of counters operating completely independently for each direction of data transfer.

This protocol when operated by all nodes on a DQDB MAN allows:

- Access in time order of access requests. Notice that this applies each time a cell (the 48-byte data part of a slot) is to be sent. A node cannot request more than one slot at a time. Once it makes a request it must wait until that request is satisfied before it is allowed to make another request.
- Each node knows enough about the status of the global queue for the access to be functionally equivalent to the operation of a centralised queue.

[65] The description here is conceptual. In reality the counters operate slightly differently to the way described but the net effect is the same.

9.5.4.1 Priorities

There are four priorities defined in DQDB. To implement this there are 4 request bits in a cell header. What happens is that each node must keep 4 request counters. A passing slot decrements the highest priority non-zero counter. A passing request increments the request counter for the corresponding priority level *and the request counters of all lower priority levels*. High priority traffic is always sent before traffic of lower priority.[66]

9.5.5 Node Attachment to the Busses

In most descriptions of "QPSX protocol" or of DQDB a diagram similar to Figure 84 is included.

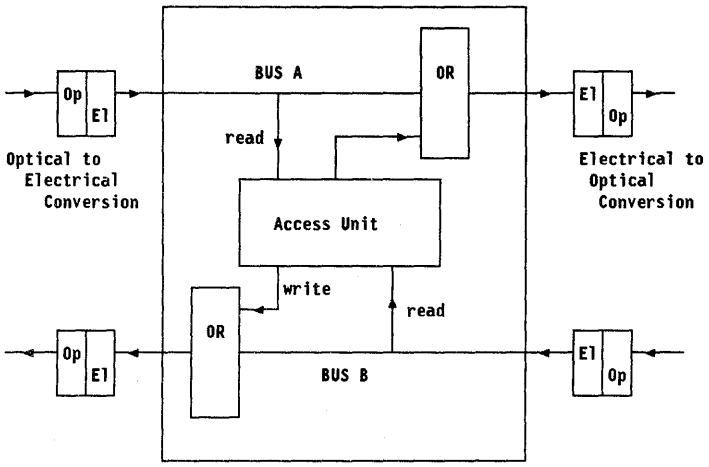


Figure 84. DQDB Node Conceptual Operation

The node is described as being "external" to the busses. That is, data "passes by" the node and the node writes into empty slots using an OR function. (Slots are preformatted to all zeros and when data is written to them it is done by ORing the data onto the empty slot.) It is said that this gives better fault isolation than token-rings because node failures that don't result in a continuous write operation will not affect the operation of the bus.

This should be regarded as an objective rather than an achievement. The current state of technology makes an "OR" write to an optical bus almost impossible. The optical signal must be converted to electrical form and then decoded. Of course if power to this function is lost (or if the circuits malfunction) then the bus will fail.

---

[66] In detail there are some problems with priorities. In the presence of "bandwidth balancing" priorities don't work and in any case, the need for priorities in this environment is questionable.

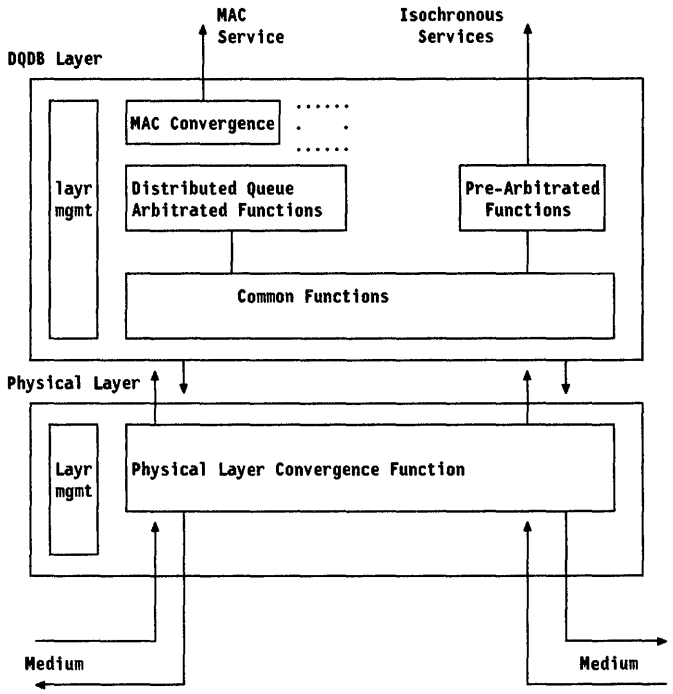


Figure 85. DQDB Physical Convergence Layer

A simplified node structure is shown above: This structure is one of the best features of DQDB. The definition allows the use of many different physical connection mechanisms and specifies rigorously the boundary between the DQDB protocol layer and the physical medium dependent layer.

The figure above, which shows the OR writing to the bus, really shows the *interface* between the DQDB layer and the physical convergence layer. The physical convergence layer is different for each supported medium. Media defined so far include:

- Fibre connection at 35 and 34 Mbps
- Sonet connection at 45 Mbps
- Fibre connection at 155 Mbps
- Copper connections at T1 (1.544 Mbps) and E1 (2 Mbps) are under study.

The line codes used are different depending on the medium. For example, on optical media DQDB will use an 8B/10B code similar in principle to the 4B/5B code used in FDDI. (See section 9.3.5.2, "Data Encoding" on page 190.) On a copper medium a different code is used.

Also, the exact node structure with respect to link connection may be different for different media. On optical media, a node structure similar to FDDI will be used (see Figure 76 on page 192). Links between nodes are asynchronous with respect to each other. An elastic buffer is used in the node to accommodate speed differences.

On copper media a synchronous operation of the bus (similar to token-ring) is used.

9.5.6 The Great Fairness Controversy

DQDB has been associated with considerable controversy due to the assertion by some people that the heart of DQDB, the media access protocol, doesn't work. Or more accurately, that DQDB doesn't provide fairness as claimed under heavy load situations.

The problem is propagation delay... the amount of time it takes for a slot (carrying a request or data) to pass from node to node along the bus.

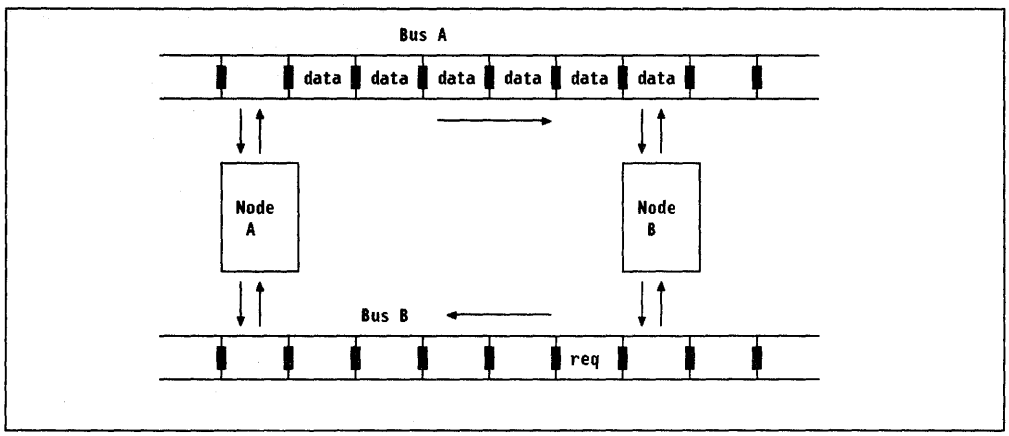


Figure 86. DQDB Fairness. Problem is propagation delay along the bus.

In Figure 86 consider the extreme condition where propagation delay consists of five slot times between Node A and Node B.

- If Node A has a lot of data to send then it may send immediately provided there are no outstanding requests.
- Let us assume that this is the case and Node A wants to send at full rate.
- Now Node B wants to send. It queues the first segment for sending and sends a request upstream. (Nodes always send a request even when the request counter is zero.)
- Node A continues to send until the request from Node B reaches it.
- When the request reaches Node A it will dutifully allow the next slot on Bus A to pass in order to satisfy the request.
- Then Node A will continue to send into every available slot on Bus A.
- When the empty slot arrives at Node B, it will use the slot and send some data.
- Typically, Node B will then want to send another segment of data.
- To do this it must place another request.

Notice the effect: Node A is able to send ten segments of data for every one segment that Node B can send! This is because it takes ten segment times after Node B sends a request for a free slot to reach it.

At a transmission rate of 155 megabits per second a single slot is 536 meters long! So the situation described above can be reached with a bus length of less than three kilometers. **There are many ways of describing this situation but "fair" is not one of them.**

9.5.6.1 Bandwidth Balancing

The way of providing fairness is called "Bandwidth Balancing". (The problem is, objectively, not as bad as it sounds.) In bandwidth balancing, a node that is sending must allow an empty slot (that doesn't have a matching request) to pass unused at defined intervals (in fact quite a low rate). This is done by the node entering a dummy request into its request counter for every n (specified parameter) segments it sends.

This has a much greater effect than might be thought at first. In the example above, while the situation described is happening, Node A lets another free slot pass. Node B will get a free slot, "think" it was the result of its last request, send a segment and another request. Thus Node B now has two requests in transit between itself and Node A. If/when another slot is released there will be three. What is happening is that the small number of free slots let pass by Node A, satisfy requests which then generate more requests.

This solves the problem of throughput unfairness reasonably well but:

1. Bandwidth balancing does *not* work in the presence of priorities.
2. There is a relatively long "convergence time". That is, it takes some time for the node lower down the bus to get enough free slots to get to the point where the DQDB algorithm is operating as intended. This means that there is still some unfairness in the access delay.

In practical systems, at least for the moment, the nodes will not be fast enough to use every available slot anyway, so bandwidth balancing will be unnecessary in early systems.

9.5.7 Data Segmentation

Because data in the DQDB system is sent as very short cells (48 bytes) data blocks, which may be of any length up to about 9 KB[67], must be segmented to fit into slots.

In order to send a data block there must be a header containing, as a minimum, the origin and destination node addresses. (DQDB allows the use of either 48-bit LAN addresses - compatible with other IEEE LAN protocols, or 60-bit ISDN compatible addressing.) Of course other information is needed in the header in addition to just the addresses. A trailer containing an FCS (Frame Check Sequence field) is highly desirable.

[67] The currently defined maximum data block length is 9188 bytes.

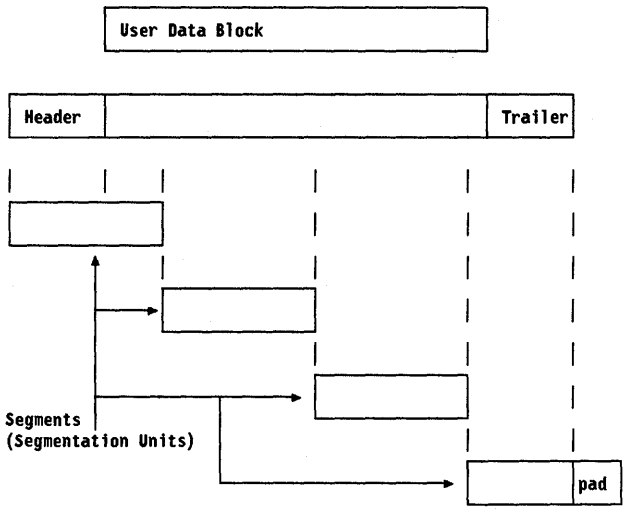


Figure 87. Segmentation of a Data Block to Fit into a Slot (Cell)

Consider Figure 87. Segments are placed into cells and sent on the bus. But how can a receiver decide which ones are intended for it? The first segment of a data block has the destination node address inside it. So a receiver only has to monitor for cells containing its address to determine which cells should be received. But what about all the rest of the cells in the data block? There is no header and thus no destination LAN address.

Here is another unique aspect of DQDB. DQDB (like most LANs) is a "connectionless" system[68] but a "connection" is established between the sending node and the receiving node for the duration of each data block. As shown in Figure 82 on page 203 the cell format contains a 5-byte header in front of the 48-byte data field. Within that header there is a 20-bit Virtual Channel Identifier (VCI) field. This VCI field is used to identify segments (after the first) of each data message.

- Each node continuously monitors passing slots for slots marked "beginning of block" containing its node address.
- When such a slot arrives, the receiving node records the VCI in that slot.
- From this point on, it will monitor passing slots and receive all those that contain a matching VCI until one is received with an end of data indication in the header.

VCIs are reused by sending nodes. Each VCI only has meaning within an individual data block, so each sending node would strictly only need one VCI. In practice, each node is allocated a few (about 4) VCIs which are cyclically reused.

There is a problem here that is not present in other LAN protocols. The node receives a block in multiple segments. What if two nodes decide to send a block of data to the same receiving node simultaneously? Segments belonging to multiple user data blocks will be received mixed up with one another. To handle this each node must have multiple receive buffers capable of holding the maximum size user data block. The receiver must be implemented in such a way as to allow multiple blocks to be reassembled simultaneously.

[68] See section 5.6, "Connection Oriented versus Connectionless Networks" on page 88.

9.5.8 Cells, Slots and Segments

The following diagram summarises the structure of a slot:

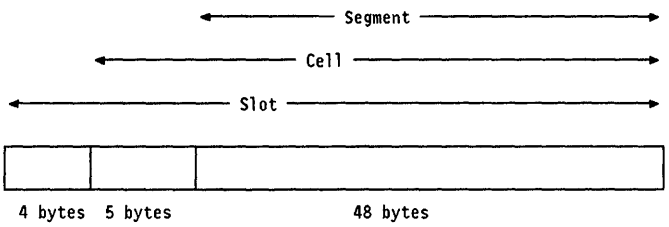


Figure 88. Slot Format

A slot is actually 57 bytes long. The slot header contains physical synchronisation and maintenance information and may be different depending on the particular physical medium used.

9.5.9 Isochronous Service

As described elsewhere in this document, isochronous data is data that must be delivered at a constant rate. Voice communication (unpacketised voice) is a typical example of isochronous service. This can also apply to digitised video of the kind that yields a continuous bit stream rather than being built into packets.

All of the above description related to DQDB protocol operation has nothing whatever to do with isochronous service. As mentioned earlier, there are two kinds of slots:

1. Queued Arbitrated (QA) slots
2. Pre-Arbitrated (PA) slots

Queued Arbitrated slots are managed by the DQDB protocol.

Pre-Arbitrated slots are allocated by the head of bus function (node containing the frame generator) at predetermined fixed time intervals (typically once every 125 µsec). Preformatted slots containing a flag to say they are prearbitrated are created by the frame generator. They are identified by the VCI (Virtual Channel Identified number) in the slot header. Because a slot contains 48 usable bytes (every 125 µsec) a single slot identified by a single VCI gives 48, 64 Kbps channels (the same as two US T1 circuits).

Nodes may use preallocated slots in any desired way. For example, single bytes may be used to carry single voice channels, or a group of 32 contiguous bytes may be used to carry a 2 Mbps clear channel link.

Precisely how the PA capability should be used is *not* defined by the IEEE 802.6 specification.

9.5.10 Fault Tolerance

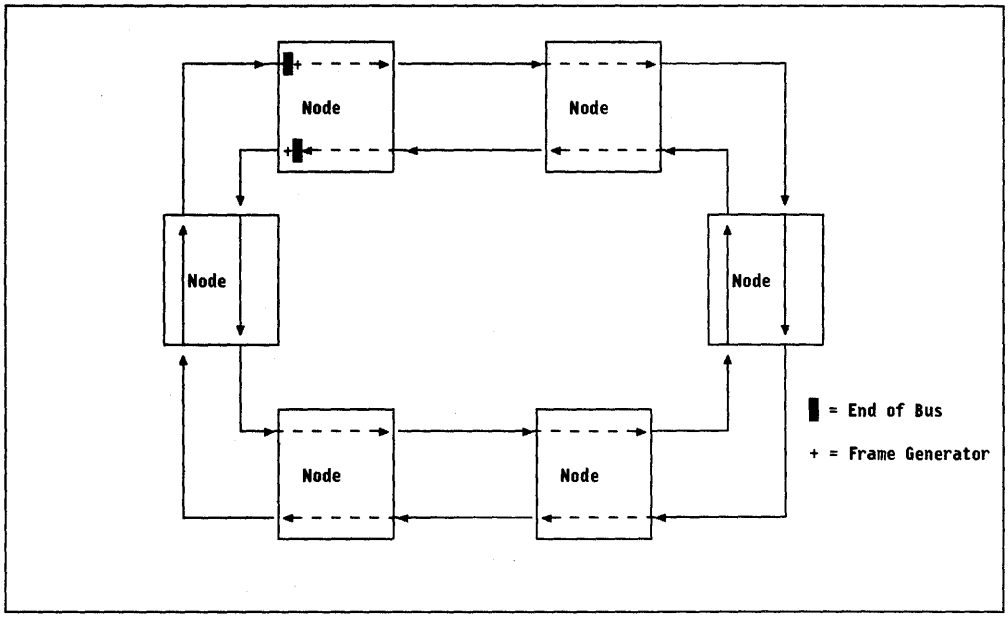


Figure 89. DQDB Looped Bus Configuration. In this configuration one node doubles as the slot generator and the busses are looped into a ring configuration.

The primary method of recovery from link failure in a DQDB system is provided by the ability to loop the busses as shown in Figure 89. The network topology is still a dual bus but it is configured as a ring. All of the nodes are capable of being frame generator.

Figure 90 on page 213 shows what happens when the ring is physically broken. The original frame generator node ceases the role of frame generator and the two nodes on either side of the break in the ring take up the frame generator role.

This allows the reconfigured system to operate at full throughput regardless of the break in the ring. Other dual ring systems (such as FDDI) fall back to half capacity when there is a break in the ring - or (again like FDDI) keep a spare unused.

In addition, the "OR-WRITE" mode of attachment to the bus isolates the bus from a large proportion of potential node malfunctions.

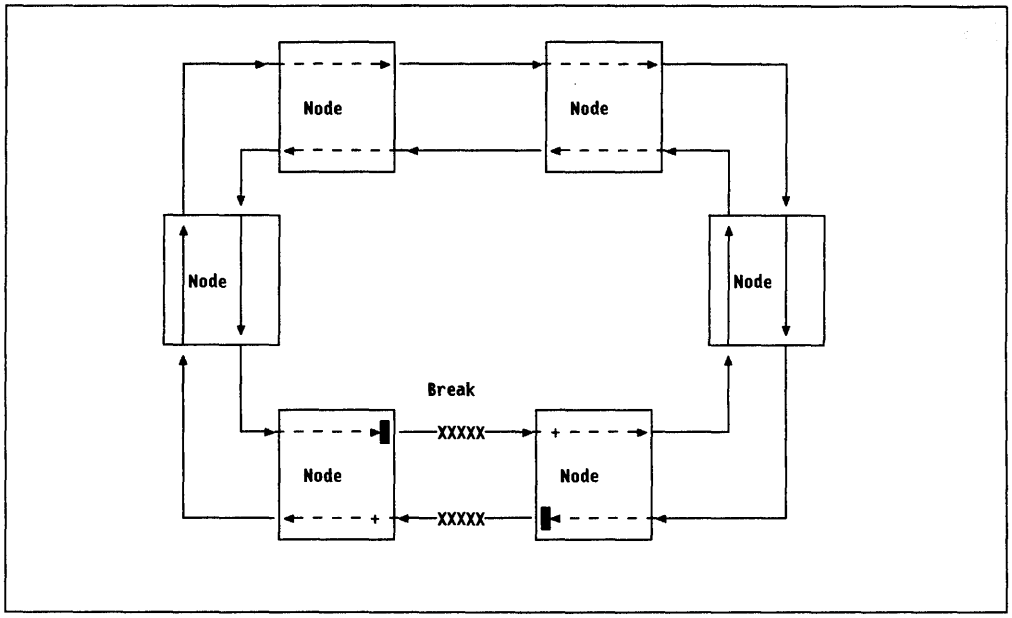


Figure 90. DQDB Reconfiguration. Each node is able to be a slot generator. Here, the bus has been broken and the two nearest nodes have taken over the role of head of bus.

9.5.11 Metropolitan Area Networks (MANs)

A Metropolitan Area Network is a new type of public network which will be provided by telecommunications carriers in various countries. The IEEE has named their standard 802.6 "Metropolitan Area Sub-Network". Thus DQDB is the IEEE protocol standard for Metropolitan Area Networks.

In some ways, a MAN is just like a big LAN - that is, a LAN covering a large geographic area but there is a critical difference: **A user device must never interface directly to the MAN**. That is, the MAN must *never* pass through end user premises. The reason is obvious, data on the MAN belongs to many different organisations and (even with security precautions such as encryption) most users are not willing to have their data pass through a competitor's building. Thus, nodes which access the MAN are always on telephone company (PTT) premises. End users are connected through "access" nodes on point-to-point links.

From a user perspective the MAN is just a fast cell switching network. The fact that a LAN type of structure is used by the PTT is irrelevant to the user provided that the user interface stays the same.

An operational MAN is built as a series of interlinked subnetworks as shown in Figure 91 on page 214. Interlinking between subnetworks (either locally or over remote point-to-point connections), is performed by bridges. End users (subscribers) are connected to the MAN through access nodes (gateways) over point-to-point links.

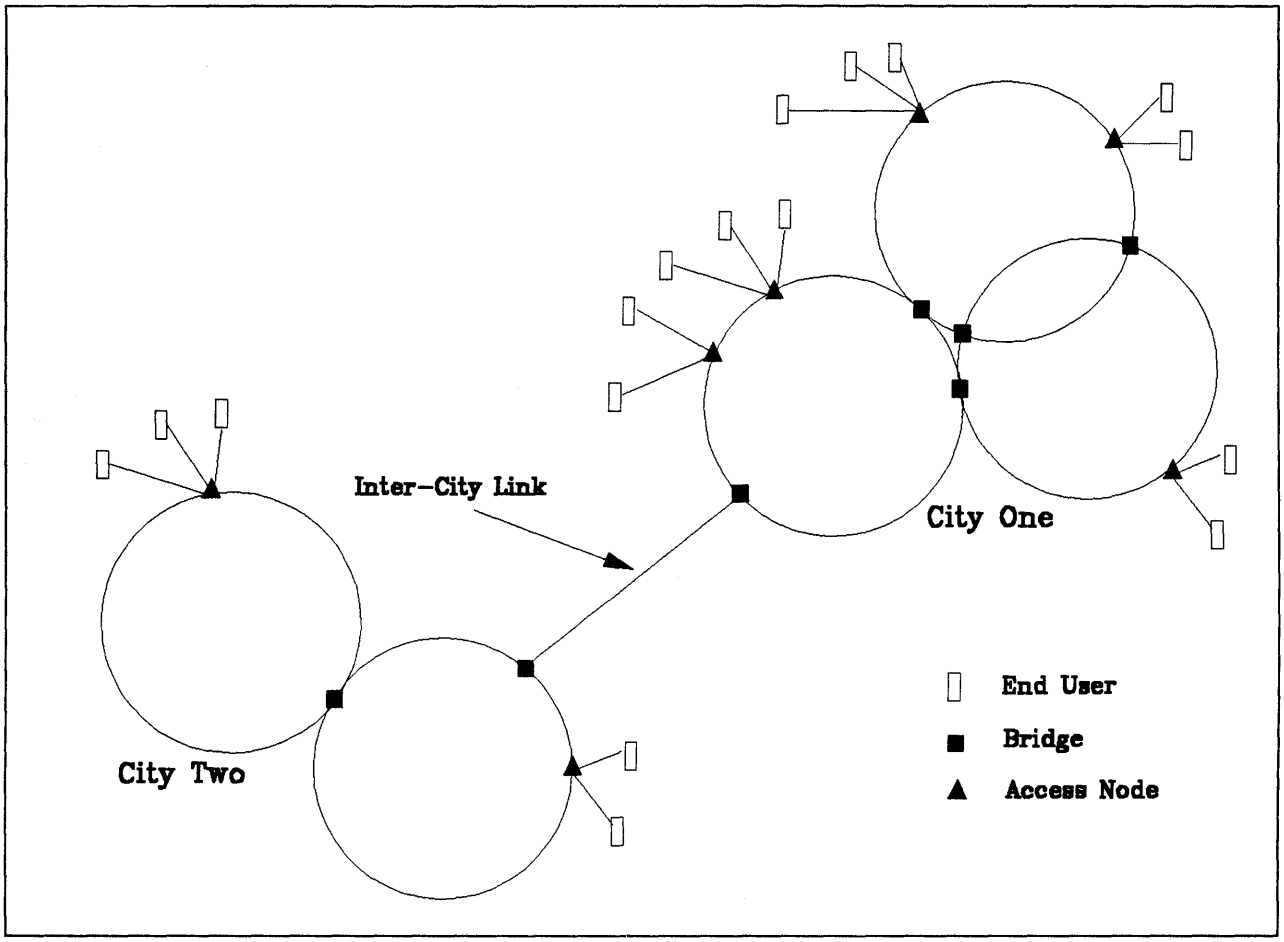


Figure 91. Configuration of a Metropolitan Area Network

Services offered by the MAN to the subscriber are the same as those offered by a LAN - albeit that they are implemented in quite a different way.

Send Data

The subscriber may send data to any other subscriber anywhere on the MAN. Since other subscribers may not want to receive data from anywhere, the access nodes must provide a filter to prevent the reception of unwanted messages.

Closed User Groups

Each subscriber address may be a member of one or more closed user groups. A closed user group is just a filter that allows free communication within members of the group (a list of addresses) but prevents communication with addresses that are not in the group (except where specifically allowed).

Broadcast

Each subscriber may send a broadcast message (indeed some protocols require this ability for correct functioning). However, it would be very dangerous for a subscriber to be able to broadcast to every address on the MAN. This must be prevented. To do this means that the true "broadcast" function of the DQDB protocol cannot be accessed (even indirectly) by a subscriber.

What actually happens is that the subscriber equipment sends a broadcast message to the access node and this node **replicates the message** and sends a copy to each member of the appropriate closed user group. That is, although the user sends a "broadcast" it is treated by the network as a number of separate messages.

As will be seen later in section 9.5.11.3, "Switched Multi-Megabit Data Service (SMDS)" on page 218, the protocol used between the MAN and the subscriber is DQDB itself. This means that the access nodes are really a type of bridge with an active filtering function built in. Access nodes also have network management and accounting (billing) functions.

In the first MAN trial networks the link to the end user is either T3 (45 Mbps) or E3 (35 Mbps) on fibre or T1 (1.544 Mbps) or E1 (2 Mbps) on copper. The inter-city links are either T3/E3 or 140 Mbps single mode fibre.

9.5.11.1 MAN Subscriber (End User) Interface

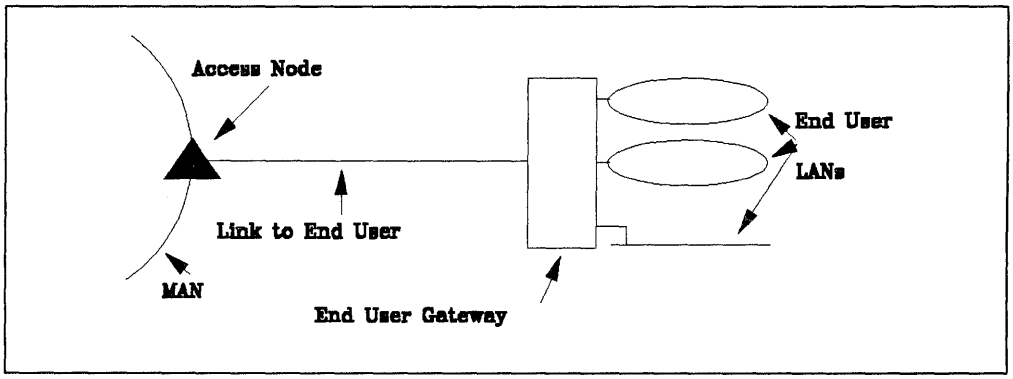


Figure 92. End User Access to an IEEE 802.6 MAN

Figure 92 illustrates the end user connection as it is forseen in many countries. The service provided (at least initially) in many countries is the wide area interconnection of LANs.

In the US, the connection between the end user and the network is the "link to end user" as shown in the diagram. This link will use the SMDS protocol and the customer (user) may purchase the end user gateway equipment from any supplier.

Outside of the US (at least in some countries) the network supplier (PTT, telephone company) will supply the end user gateway equipment and thus the end user interface to the network would be the LAN interface. How this will work legally and administratively is not yet settled and further discussion is outside the scope of this document.

Using the Australian "Fastpac" network as an example,[69] there are two accesses to the network available - a two megabit (4-wire copper, E1) interface and a 35 megabit (fibre, E3). The two megabit interface is published and users may attach equipment from any supplier to this interface. The 35 megabit interface is considered proprietary and users purchase the end user gateway equipment as

[69] Because it is the only fully commercial implementation yet announced.

part of the network service. Different countries may adopt quite different approaches to this problem.

Another feature of MAN networks is their pricing. Since the network is shared the user will be billed for data traffic (just as with X.25). In addition there will be a charge for link connection to the network (again just as X.25). However, since there are no virtual circuits (no connections) there can be no charge for connection holding time (a significant part of the cost in many X.25 networks). Precise prices must wait until network providers decide on their tarrifs.

9.5.11.2 User Interface to Fastpac

The first fully commercial (with universal service and published tarrifs) MAN is being built by Telecom Australia and is called "Fastpac". One problem encountered in building the network was that there is no internationally standardised protocol available suitable for an end user interface. At high speeds (35 Mbps) the DQDB protocol itself could be used even though that requires special hardware (interface chips).

At the slower (2 Mbps) interface speed the need was for a *connectionless* interface that was easy for equipment suppliers to implement and nevertheless gave users full access to the network's facilities.

The solution adopted was to design a new interface constructed solely from standardised protocol elements but put together in a new way. Diagrammatically the interface is as follows:

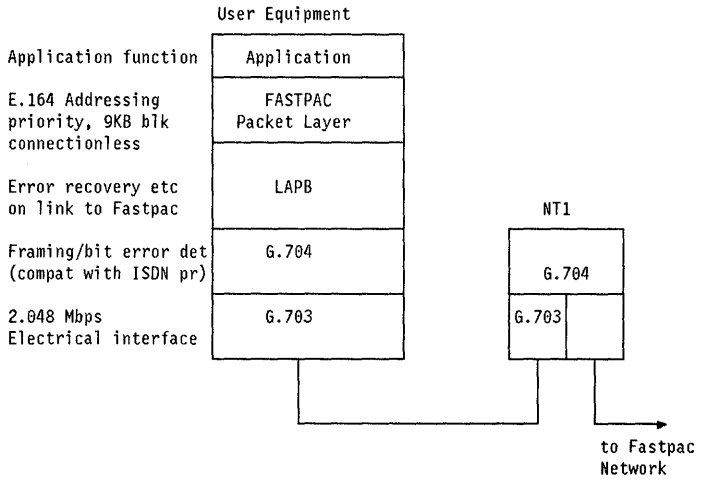


Figure 93. Fastpac 2 Mbps Access Structure

The concept is extremely simple. The user equipment builds an IEEE 802.6 frame including the frame header and sends it to the network over an E1 (2 Mbps G.703/G.704 connection) using LAPB link control to protect against errors on the local access link.

This design is reasonably clean and uncomplicated. Logically it is very close to the structure of X.25 (albeit that it is connectionless) and follows the OSI layering. There is very little that is new in the interface - just existing standards used together in a new way.

- Layer 1 (Physical Access) is done with G.703/G.704 standards.

These are the physical layers of ISDN (primary rate access). The specification is identical to CCITT ISDN specifications at this layer.

- Layer 2 (Link Layer) uses LAPB link control.

  LAPB is an implementation of the international standard link control HDLC (HDLC is a superset of IBM SDLC). LAPB is the link control used in X.25 (and accepted as part of OSI).

  Note that the scope of LAPB is from the user device to the network access point - NOT across the network. That is, LAPB is used to protect the transmission of user data to/from the Fastpac network access point. LAPB is NOT used across the network (unless the end user decides to implement this as an end user function).

  There is an additional mode allowed for HDLC where data may be sent as Unnumbered Information (UI) frames. This means that if an error occurs on the link then the data will be discarded. This is an optional mode of operation for products like LAN bridges which don't need this level of error recovery.

- Packet layer is IEEE 802.6 DQDB (Distributed Queue Dual Bus) access packet headers.

  There is no "protocol" as such implied by the use of this header. The protocol is "connectionless" and "stateless" and therefore has no responses or command sequences. There is no flow control protocol. A packet header contains the origin and destination Fastpac addresses as well as some control information indicating what the user wishes to be done with the packet (when sending) or network status conditions (when receiving).

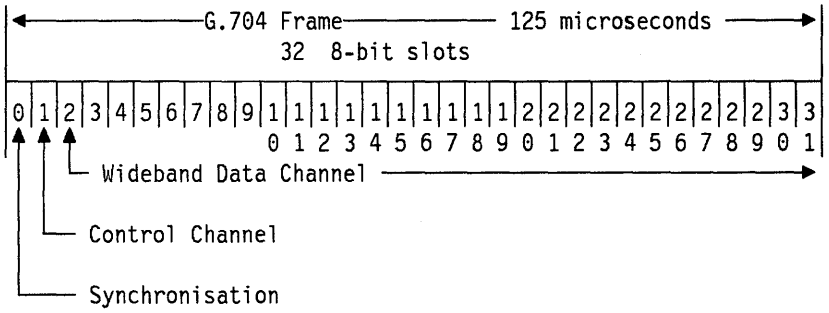


Figure 94. G.704 Frame Structure as Used by Fastpac Interface

The above structure allows a user to construct as many as 31 channels of one slot each, one channel using a concatenation (aggregation) of all 31 data slots, or any combination of slots to form any number of channels up to 31. This structure is further discussed in section 6.1.4, "ISDN Primary Rate Interface" on page 115. Different from ISDN, where it is used as a signaling channel, slot 16 is used for data in Fastpac in the same way as any other slot.

The connection to Fastpac uses only two logical channels. The first logical channel is used for data as is made up by aggregating as many slots as needed from the G.704 frame. This is similar to the "wideband" mode of ISDN.

For example, a bandwidth of 640 Kbps could be achieved by using slots 2 to 11 from the G.704 frame. The maximum number of slots used is 30 (numbers 2 to 31). Slot 1 is used as the control channel.

At first sight this system looks ridiculous. The access from the user to the network is a point-to-point link which runs at two megabits per second full-duplex regardless. What purpose is served by limiting the throughput of the link? This is actually very sensible. Australia is a very large geographic area (2.9 million square miles) and one objective of Fastpac is to provide service to all locations in the country. Fastpac will cover the major cities but what about small towns a long distance from the city (as much as 1500 miles). The structure enables the access link to be multiplexed through Telecom Australia's digital backbone network. By limiting the number of slots Telecom can provide access to Fastpac from anywhere in a reasonably cost effective way.

At a joint press announcement in June 1990, IBM and Telecom Australia announced that IBM is developing a token-ring bridge that will operate across this Fastpac interface.

9.5.11.3 Switched Multi-Megabit Data Service (SMDS)

SMDS is a definition of the features and functions to be offered by a high speed public data network. Initially, the scope of such a network was defined to be within a local area (in the US this is called a LATA) but there is no reason why the service cannot be extended to cover any geographic area desired. It does define the technical interface from the customer to the PTT premises but it does *not* define how the service is to be implemented. Figure 95 shows the reference configuration.

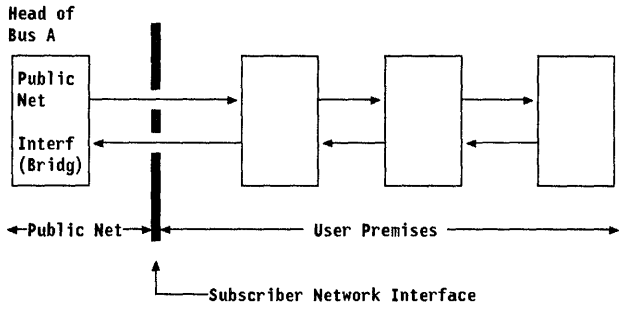


Figure 95. SMDS Subscriber Network Interface

Reference Configuration

The most important characteristic of the reference configuration is the interface from the user to the network. This is called the "Subscriber Network Interface" (SNI). The interface between the subscriber and the network provider is the link from the user's premises.

There may be one or many pieces of customer equipment attached to the same link. In the case where multiple customer devices are attached, these devices have concurrent access to the network over the same link.

Interface Characteristics

SMDS is defined to use two speeds - US "T1" (1.544 Mbps) and US "T3" (45 Mbps). The interface protocol is IEEE 802.6 (DQDB).

Service Features

- Connectionless Transfer

  The user sends "datagrams" to the network containing a header with both source and destination network addresses included. All data is sent this way, there is no such thing as a "virtual circuit". Datagrams may be up to 9188 bytes in length.

- Closed User Groups

  The service defines a number of features which provide much the same facilities as the "closed user group" does in an X.25 public network. These are called:

  - Destination Address Screening
  - Source Address Screening
  - Source Address Validation

  These enable a user to define a list of addresses to which this address is allowed to send and from which it may receive. A user may decide that all the devices belonging to this organisation may only communicate with each other. Or they may be organised in groups. Or some may be able to send/receive to/from any address in the public network.

- Broadcasting

  The service defines a feature called "group addressing" which allows a user to send to a group of other users. This means that users can define a set of nodes and have the services of a "virtual LAN".

- Access Classes

  Access classes are really throughput classes. The network uses a form of "leaky bucket" flow control to limit the flow of data from a user to the network. See section 8.3.5, "Flow and Rate Control" on page 163. Different parameters for the "leaky bucket" can be specified to give different users different throughput characteristics. Presumably different throughput classes will attract different charges for network usage.

- Performance Objectives

  An important point about SMDS is that network performance objectives are stated. The initial objective is a 20 millisecond delay between users connected in the same local area.

Conclusion

The important thing to remember about SMDS is that it is a service and an interface specification not a network architecture. There are already network providers who have announced their intention to provide SMDS services using MAN technology internally. In the future as broadband ISDN is introduced it is likely that SMDS services will become a service of a much wider broadband ISDN (ATM) public network.

Chapter 10. The Frontiers of LAN Research

It is widely accepted in the research and standards communities that current LAN and MAN technologies are not adequate for speeds in the one gigabit per second and above range. FDDI LANs become less efficient as the geographic size, number of stations, or link speed increase, because only one station is allowed to transmit onto the LAN at any one time (where potentially many could do so). DQDB exploits concurrent access but suffers from the fairness problem which becomes greater as the geographic size and link speeds are increased. In addition DQDB requires that a station should be able to receive data blocks from many senders "simultaneously". At very high link speeds this becomes both difficult and costly to implement.

Researchers throughout the world are developing many proposals for LAN protocols to operate at speeds above one gigabit per second. Morten Skov (1989)[70] asserts that more than 50 such protocols have been reported in the literature. This chapter deals with two prototype LAN systems developed by IBM Research which are designed to operate in the very high speed environment. Although both systems have been built in prototype form and publicly demonstrated it must be emphasised that **these are experimental prototypes only**. They were built to gain a better understanding of the principles and problems of operation at speeds above one gigabit per second. Information about them is included here for educational purposes only.

[70] *Implementation of Physical and Media Access Protocols for High Speed Communication.*

10.1 MetaRing

MetaRing is an *experimental* high speed LAN protocol designed to improve the functioning of token passing rings at very high speeds. It was developed by IBM Research at Yorktown Heights, New York.

In most LAN architectures (TRN, FDDI, CSMA/CD...) only one device can be transmitting onto the LAN at any one time. LANs with dual rings or busses can sometimes support two simultaneous devices but that is the limit.[71] In the very high speed LAN environment where the geographical extension of a single frame becomes small compared to the LAN size, we have the opportunity to do significantly better than this, improving the throughput of the LAN many times.

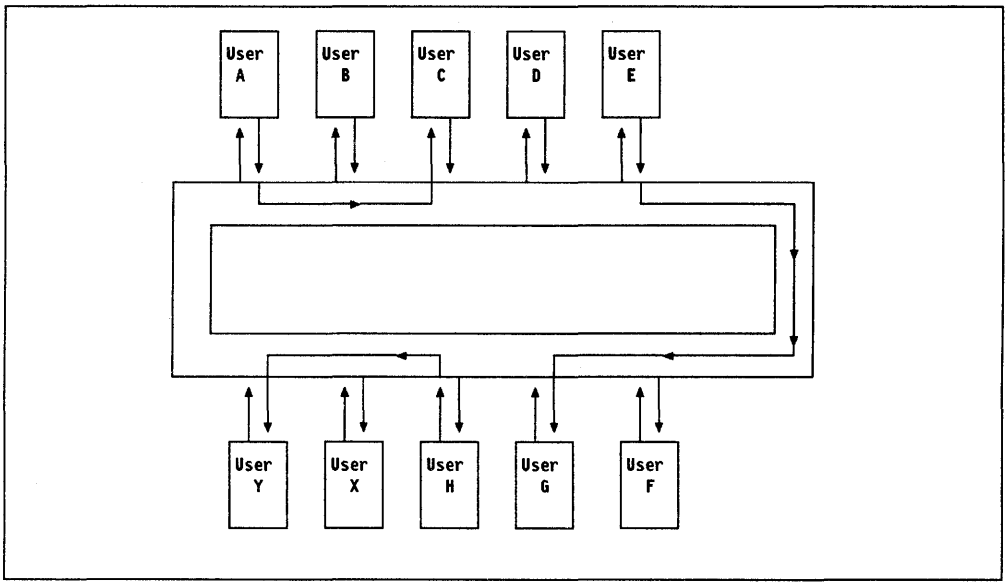


Figure 96. Multiple Simultaneous Communications on a Ring Topology

Figure 96 shows a hypothetical ring LAN with user A sending to user C, user E sending to user G and user H sending to user Y. If this kind of operation were possible then the total throughput of the LAN would be improved significantly (in the hypothetical example, by 300%. Of course, when user A sends to user Y then only one user may transmit.

Looking at the problem of achieving this on a token ring (including FDDI) a number of observations can be made:

- At four megabits per second a single bit on a wire is about fifty meters long. Since with TRN protocols there is only a 2-bit buffer in each ring attachment, in most LANs, the beginning part of the frame is being received and discarded by the sending node before the end of the frame is transmitted! The conceptual picture of a frame on a LAN looking like a railway train on its track gives totally the wrong impression.

  Even at 16 Mbps a single bit is still 16 meters long. This means that at the usual token ring speeds there is not much "storage effect" available on the LAN itself for internode buffering.

[71] The term LAN here is used to mean "LAN Segment". A LAN consisting of multiple segments connected by bridges or routers can of course have one (or two) device(s) transmitting simultaneously on each LAN segment.

- Perhaps the desired result could be achieved by using many (more than one) active tokens. But if this technique was used then the tokens would tend to catch up with one another and group together in a very short time - negating the benefit of having more than one.
- Even if there were multiple tokens if two stations started sending at one time then there could be collisions - a frame arriving at a node while that node is itself transmitting.

One solution is to use a slotted bus technique with fixed length slots. Once a "free" slot is detected the node may confidently transmit into that slot without danger of collision. This is done in for example, in DQDB. But there are other problems here. If you want to send a single block into consecutive slots, the same problem recurs unless there is some other protocol in operation to ensure that sufficient empty slots will arrive consecutively. (See the description of CRMA in this chapter.)

The approach taken with MetaRing is called buffer insertion:

- The idea of having a token is to remove the possibility of collisions. If there are multiple tokens then there is the possibility of collision which must be handled. Therefore a token is not much use in this environment and therefore MetaRing does not use a token.
- If one node starts transmitting while another (downstream) node is transmitting then when the data from upstream reaches the downstream node a collision will occur unless something is done to prevent it.

  The MetaRing solution here is for each node (attached device) to have a large (larger than the longest possible frame) elastic buffer on its receive side. If data arrives from upstream while the node is transmitting then the upstream data is received into the insertion buffer. As soon as the node finishes its own transmission, data from the insertion buffer is immediately sent onto the LAN without waiting for the whole frame to be received. Data received when the node is not sending is sent through the insertion buffer with minimal delay.

  > Because the transmit clock is independent from the clock speed of the received data a buffer is needed to prevent overruns if the receive data stream is faster than the transmit one. The insertion buffer performs this function for data passed through the node. Underruns are not a problem since the transmitter will send idles if data arrives too slowly.

  This technique usually results in unacceptably long latency delays around the ring. This however can be overcome by using very high LAN speeds (100 Mbps and above).

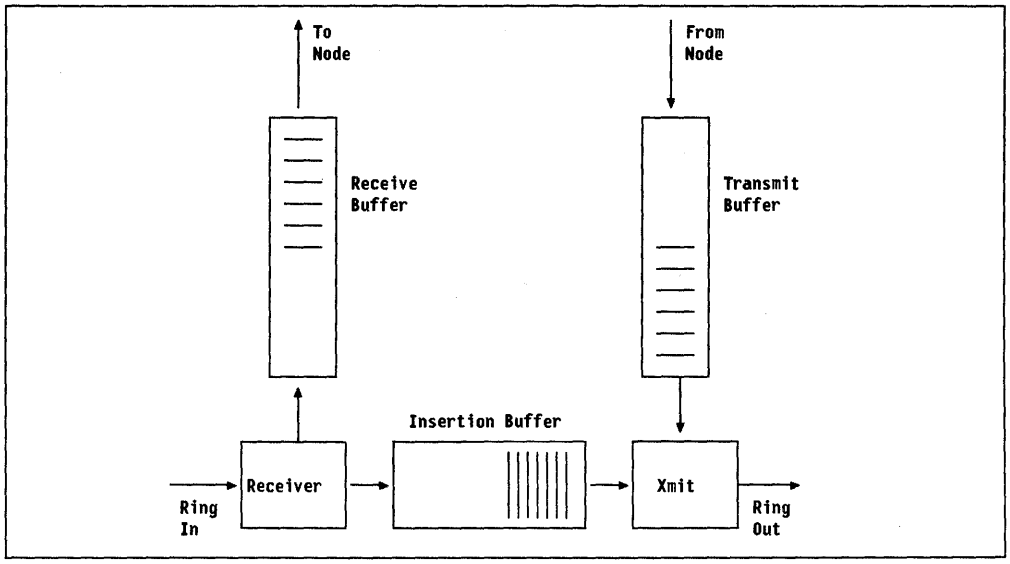


Figure 97. Buffer Insertion

The basic structure of a MetaRing node is shown in Figure 97. The first operational rule is:

- **If** nothing is currently being received into the insertion buffer **and** the insertion buffer is empty **then** the node may transmit.

An analogy

MetaRing operation is like the operation of an English traffic roundabout (some of which are more than 200 meters in diameter and have six or more entrance roads!). Traffic on the roundabout has priority over traffic entering but once a car starts to enter then traffic already on the roundabout must slow up and make room. Under medium load conditions this works very well.

But, as any connoisseur of English traffic roundabouts will be quick to point out, if one entering road has very heavy traffic then traffic on other entrances can be locked out for considerable periods of time.

This can be solved by the installation of traffic lights at each entrance to the roundabout.

- When a node receives anything from upstream it checks the address in the frame header to determine if it is the destination of the message. If the frame is addressed to this node then the data is directed into the node's receive buffer and does not go into the insertion buffer. (Of course, broadcasts go into both buffers.)

This is a good principle provided:

- The insertion buffer is large enough to accommodate the longest allowable frame, and
- The ring speed is such that the additional buffering does not cause a problem with the additional insertion buffer delays.

According to Cidon and Ofek[72] simulation studies suggest a worst case total delay of one millisecond on a 100 megabit per second ring. (This depends on the number of active stations and the maximum frame size.)

Looking again at Figure 96 on page 222. If any user starts transmitting around the ring to a close upstream neighbour (an extreme example would be User B sending to User A) then this user could "hog" the ring and lock all other devices out until it ran out of data. (Exactly what happens on English traffic roundabouts in peak hour.) A means of ensuring "fairness" is needed.

10.1.1 Fairness

MetaRing uses a counter rotating control signal to allocate ring capacity to requesting nodes. This control signal is called a SAT (short for SATisfy). A SAT is not like a token *and* it travels in the opposite direction to the data.[73] In order for the SAT to travel in the opposite direction to the data, a path is needed for it to travel on.

In MetaRing there are *two* rings which rotate in opposite directions - just like FDDI without a token. Both rings are used for data transport (different from FDDI where only one of the two rings is used for data - the other is a standby). The SAT on one ring allocates access rights for the other ring. Both rings operate in exactly the same way. There are two SATs, one on each ring, and each SAT allocates capacity for access to *the other* ring.

But if the SAT has to queue behind the data in all those potentially long insertion buffers then the SAT would be limited in effectiveness. SATs are sent around the ring preempting data transmission as they go! When a SAT is to be sent, data transmission is suspended, the SAT is sent, and then data transmission is resumed.

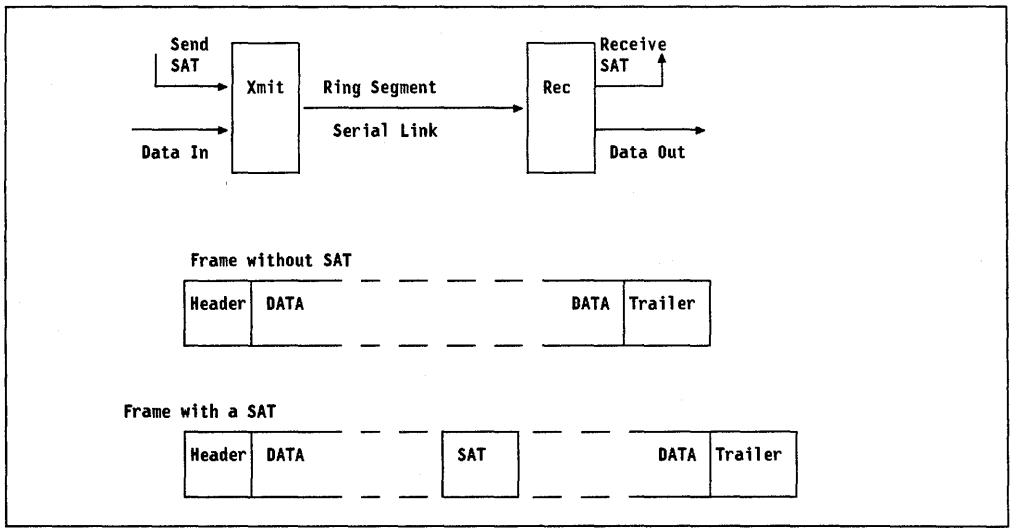


Figure 98. The SAT Control Signal

Because the SAT can preempt data transmission it travels around the ring at maximal speed (with only link delays and minor buffering in each node).

---

[72] *MetaRing - A Full-Duplex Ring with Fairness and Spatial Reuse*

[73] A little like a European driver on arrival in England... the first roundabout outside the ferry terminal in Dover is hell!

When a SAT is received by a node the node is given a predefined quota of data it may send onto the other ring.

- A node is given permission to send a frame (or a quota of frames) when it receives a SAT.
- If it has nothing to send or if it has sent its quota since the last SAT was received then the SAT is forwarded.
- If not (meaning the node does have data to send but the ring has been busy all the time since the last SAT arrived), then the SAT is held until the node has sent a frame (quota of frames).

More formally:

The SATisfied Condition

The node is SATisfied if a quota of data has been sent between two successive visits of the SAT message or if its output queue is empty.

The SAT Algorithm

DO when the SAT message is received:

- If the node is SATisfied then forward the SAT.
- Else hold until SATisfied and then forward the SAT.

After forwarding the SAT the node obtains its next quota of frames.

The SAT algorithm results in fair access.

- Each rotation of the SAT message gives the subset of busy nodes permission to transmit the same amount of data.
- The SAT algorithm is deadlock free.

10.1.2 Priorities

A priority mechanism is implemented by assigning a priority number to the SAT. When a SAT has a priority number a receiving node may only send frames with that priority or a higher one.

When a node has priority traffic to send it may increase the priority number (when it forwards the SAT) to ensure that its priority traffic is given preference. Other nodes may increase the priority number further if they have still higher priority traffic. When the node that increased the priority number detects that there is no more priority traffic then it must decrease the priority number in the SAT to what it was before that node increased it.

10.1.3 Control Signaling

The communication of the SAT signal around the ring is only a special case of control signaling. There are more control signals than the SAT only.

Control signaling is achieved by using a redundant serial codeword in the transmission code. MetaRing uses the same physical layer encoding as FDDI (this 4/5 encoding is discussed in section 9.3.5, "Physical Layer Protocol" on page 189).

Each group of four data bits is actually sent as five bits. This leaves a number of five-bit combinations that may be used to signal special conditions.

As noted above (for the SAT) and illustrated in Figure 99, control messages preempt data transmission with no loss in efficiency.

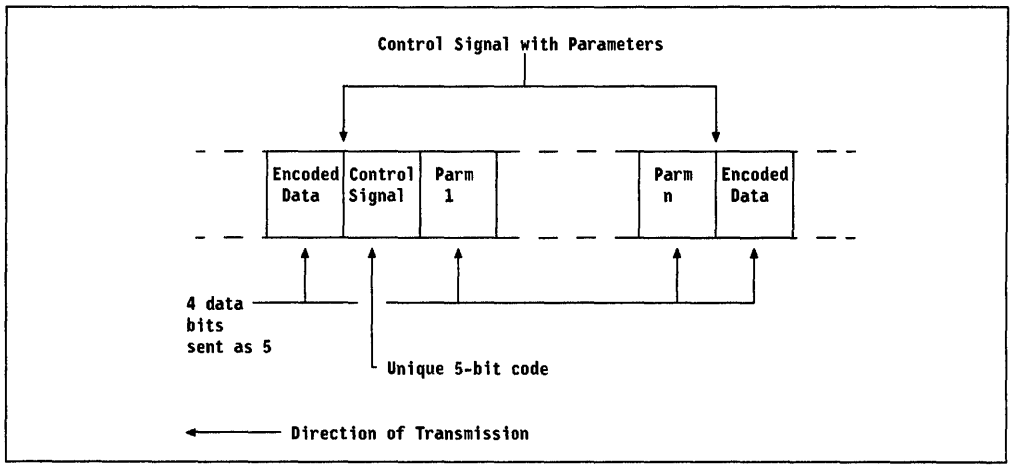


Figure 99. Control Signal Format

Control messages may travel on either ring. Some control messages (such as the SAT) travel in the opposite direction to the function they control. Others travel in the same direction as the controlled function.

10.1.4 Ring Monitor Functions

The functions required to maintain the ring are similar to those functions required in FDDI and token-ring. There must be a ring monitor function to handle:

- Setting up the rings.
- Creating SATs and handling error conditions such as lost or multiple SATs.
- Re-configuration after a break in either ring.

10.1.5 Addressing

When the ring is initialised, nodes are allocated a temporary address called a Physical Access Name. Physical access names are really just the sequence of nodes on the ring (1, 2, 3...). Each node must keep a table relating the physical access name of each node on the ring to the node's physical (unchanging) address. When a new node enters the ring a reinitalisation process takes place and all physical access names are reassigned.

The physical access name is used partly to determine which ring should be used to send data to another node. There is a selective copy ability in addition to the usual broadcast and point-to-point addressing modes. These are implemented using the physical access names.

10.1.6 Fault Tolerance

Since MetaRing uses dual counter-rotating rings they may be wrapped onto one another should a break occur. This is just like the method in FDDI.

A protocol inconsistency arises in that the SAT messages normally travel in the opposite direction to the ring they control. When the rings are wrapped, they become one ring (albeit one that passes through each node twice). There is a means included in the protocol that takes account of the situation of connecting

the two rings as one and the resulting condition of possible multiple SATs and lost SATs.

The wrapping mechanism is so designed that any arbitrary disconnected section of a MetaRing may continue to operate as a disconnected ring.

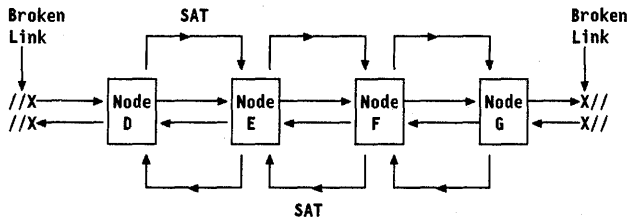


Figure 100. MetaRing Operation on a Disconnected Ring Section. Protocols operate correctly (and maintain their fairness property) over a disconnected section of the full duplex ring.

10.1.7 Throughput

In practical LAN situations (in LANs with a large number of nodes), a high percentage of the LAN traffic is local in scope. That is, most LAN traffic is within a work group. Also, typically, people site servers close to the work group being served (a print server will usually be close enough for a person to walk over and pick up the printout).

This locality of reference characteristic enables MetaRing LANs to be very efficient indeed. Throughput of many times that of FDDI at the same speed is achievable. In a uniform traffic situation a throughput improvement of eight times can be achieved. Because both rings are used for data traffic, even in the worst case throughput will be twice that of FDDI.

10.1.8 Slotted Mode

In some system implementations there is a concern with variable ring latency. There is a mode in MetaRing which is designed to eliminate the insertion buffer at each node by formatting the ring into fixed length slots.

When this happens one node performs the function of slot generator and there is a busy/empty bit in the beginning of each slot. A node may send into an empty slot subject to the fact that the SAT protocol still operates normally. This reduces ring latency at the cost of losing the capability to send complete frames longer than the fixed slot size in a contiguous manner.

10.1.9 Synchronous and Isochronous Traffic

There is also provision in MetaRing for the transport of isochronous traffic such as there is in FDDI-II. After a connection is set up by a monitor, isochronous bandwidth is available with a periodic rate of 125 microseconds. It can consist of single or several multiplexed channels, potentially with different bit rates. Isochronous streams bypass insertion buffers and preempt all other transmissions. Specific isochronous start and end delimiters allow network stations to recognise this type of traffic.

Traffic with a real time (guaranteed bandwidth) requirement (in FDDI terminology, "synchronous" traffic) is managed by using a mechanism very

similar to the timed-token method used in FDDI. See section 9.3.2, "Access Protocol Operation" on page 185.

A control message (called ASYNC-EN) similar in function to the SAT, also travelling in the opposite direction to the flow of data, is used to control the integration of asynchronous and synchronous traffic.

10.1.10 Practical MetaRing

In its experimental implementation MetaRing uses dual, counter-rotating, 100 megabit per second (optical) rings (the same as FDDI).

10.1.11 Advantages of MetaRing

Operates Efficiently at Any Speed

Traditional token-rings (including FDDI) decrease in efficiency as the ring speed is increased. MetaRing will operate efficiently at speeds well into the gigabit per second range.

Not Particularly Sensitive to Ring Length

In both FDDI and token-ring, as the ring length increases (both in terms of physical length and in number of attached devices) throughput efficiency decreases substantially. MetaRing throughput efficiency is not affected by ring length. This is because the SAT allocates capacity before it is used and then passes on. It is not like a token that is held during transmission of a block.

Significantly Improved Throughput

Again, compared to token-ring or FDDI, MetaRing can achieve very high overall throughput. This comes from the ability for many devices to transmit onto the ring at the same time. The amount of gain is highly dependent on the "locality of reference" but is generally many times higher than an FDDI ring of the same speed.

Same Cost as FDDI

An engineering evaluation shows that the circuit complexity needed to implement MetaRing is comparable to that of FDDI albeit that you need more storage on the interface chip than is required in FDDI.

A MetaRing chip set can be constructed for much the same cost as an FDDI chip set.

Since the throughput is much greater than FDDI at much the same cost, the cost performance is much better.

10.2 Cyclic Reservation Multiple Access (CRMA)

CRMA is an experimental LAN protocol implemented as a prototype by IBM Zurich Research Laboratory. The prototype was demonstrated (using a link speed of 1.13 gigabits per second on single mode optical fibre) at the Telecom'91 exhibition in Geneva. The demonstration included prototype connections to IBM PS/2 systems, IBM RS/6000 workstations, IBM 3172 Interconnect Controllers and connection to FDDI rings.

The advantages of CRMA are:

- It will operate with fairness over metropolitan distances (hundreds of kilometers).
- It does not degrade in efficiency as link speed is increased.
- Although data is sent in cells, blocks of user data are sent in a contiguous stream of cells, so receivers don't need multiple reassembly buffers as, for example, are required in DQDB.

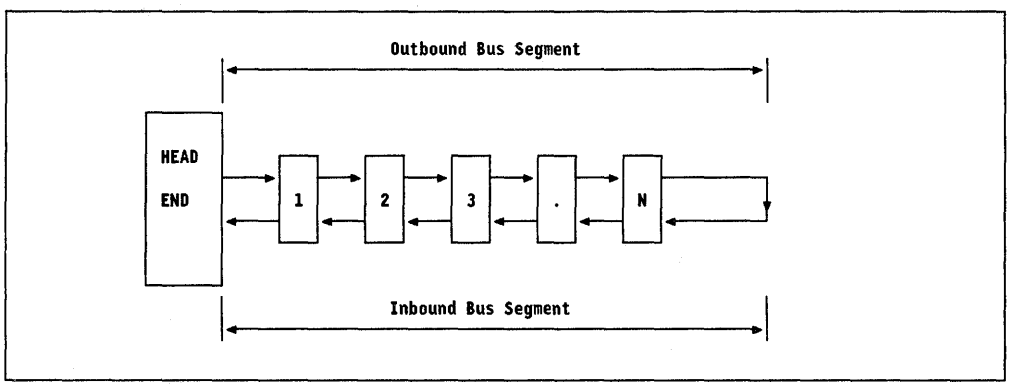


Figure 101. CRMA Folded Bus Configuration

A CRMA network is configured as either a "folded bus" or a "dual bus".

10.2.1 Bus Structure

The folded bus configuration is shown in Figure 101. The bus itself is unidirectional but passes through each node twice (once in each direction). Nodes transmit data on the "outbound bus segment" (that part of the bus traveling *away* from the head-end). They receive data on the "inbound bus segment" (that is, where the data flow is toward the head-end. (This is one advantage of the folded bus configuration, that the nodes do not need to know the position of other nodes on the bus in order to send data to them.) As will be seen later, nodes receive control information on the outbound bus segment as well as transmitting on that segment.

In practical systems the head-end function (and the tail-end too) would be incorporated into every node so that the network would be configured as a physical loop. Only one node at any time would perform the role of head-end (and also of the tail-end). Then, if a break occurs anywhere in the bus it may be reconfigured around the break. A node on one side of the break would become the new-head end and a node on the other side of the break the new tail-end. This technique would enable the bus to continue operation after a single break. DQDB networks use this same principle as described in section 9.5.10, "Fault Tolerance" on page 212.

10.2.2 Slotted Format

Like DQDB, CRMA uses a slotted bus structure. The slot format is conventional and is shown below:

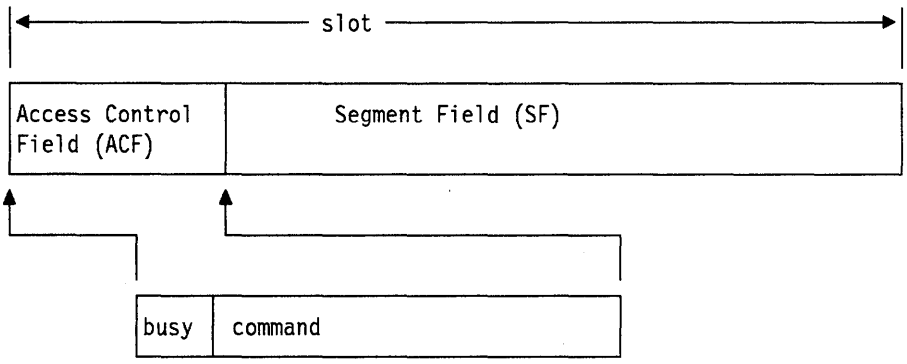


Figure 102. CRMA Slot Format

10.2.3 The Global Queue

In order to understand CRMA it is necessary to consider the concept of a global queue. As described in section 9.1.2, "Access Control" on page 172, many users (or nodes) on the LAN generate data blocks at different times. It is reasonable to consider these blocks, as they are generated, as forming a global queue for the single resource of the LAN. Because the queue is distributed geographically we can't know the true state of this queue at any particular instant in time. If we conceive of "fairness" to mean FIFO operation then there is the logical problem for the LAN protocol of how to control which node is allowed to send data next.

CRMA solves this problem by appointing a single node (the head end node) to:

1. Keep track of the global queue size as it builds up.
2. Control a time reference so that attached nodes can know their positions in the queue.
3. Control when nodes are allowed to transmit.

10.2.4 The RESERVE Command

The head end of the bus controls the whole protocol. At intervals,[74] it sends out a special slot with a command in it called a "RESERVE". This command contains two important things:

1. A cycle number that is used by the node to determine its place in the queue and
2. A counter field that keeps track of how much data has become available for transmission since the last RESERVE command.

When the RESERVE command arrives at a node the slot count in the command is increased by the amount of data (number of slots) that has become available for transmission at this node *since the last RESERVE command.*

[74] The size of the interval is a tuneable value. It is normally predetermined but may be modified by the head end depending on traffic conditions on the bus.

Notice that the RESERVE command doesn't give any node permission to send anything. When it returns to the head-end node it contains a count of the amount of data (expressed in slots) that has become available for transmission since the last RESERVE command. The RESERVE command returns to the head-end node in the minimum time of one network latency delay (outbound and inbound bus) - because (unlike a token) it cannot be held by any node. The attaching nodes just add to the counter, they can't delay the slot.

10.2.5 The Cycle

Permission to send data is allocated in "cycles". The head-end node sends a START command including the cycle number. When a node receives the START command it has permission to send into the next n available slots. The number of slots the node may send "n" is the number it requested on the previous RESERVE command with the same cycle number.

The START command travels around the bus and each node in turn may send the amount of data that it requested for this cycle.

The head-end node then keeps a queue of cycles that it will allocate in the future. With each cycle number it keeps the number of slots needed for the cycle and also keeps track of the total number of slots for all outstanding cycles.

When the head-end node begins a cycle it generates a START command followed by exactly the number of empty slots needed for that cycle. When it has finished generating one cycle it will then send a START command for the next cycle immediately.

This whole process results in a much better FIFO ordering in the sending of data than does, for example, a token controlled process. In addition it removes the throughput inefficiency of the token principle caused by the latency delays between when a node finishes sending and when another node may start. Under conditions of load, every slot time on the LAN is useable.

The process requires cooperation between the head-end node and the other nodes on the LAN. The head-end node *never knows* any detail about individual nodes. All it sees is the total amount of data requested by all nodes for each cycle.

10.2.6 Operation of the Node

When a node receives a RESERVE command, it checks to see if any new data is available to send. If so it adds the number of slots required to contain that data to the count field within the RESERVE command. It also remembers that it "made a reservation" to send this number of slots in this particular cycle. So the node must keep a record of cycle numbers and the number of slots it requested for each cycle.

For example, in Figure 67 on page 172 data arrives (or is generated by the node) at each node at different times. If this was a CRMA LAN a RESERVE command sent out at time 1 (let's call it cycle 12) would tell the head-end node the total amount of data that was available at nodes (users) A and E. But user A would keep a record that it requested a number of slots in cycle 12 (so would user E). A RESERVE command sent out at time 2 (call it cycle 13) would only show the amount of data that had arrived at node D (since cycle 12). At time 3

(or cycle 14) the head-end would hear about still more data, this time from nodes B and C.

Let us assume that the bus was busy with previously queued data until after the RESERVE for cycle 14 had returned to the head-end. At this time then, nodes A and E have reservations for cycle 12, node D has a reservation for cycle 13 and nodes B and C have reservations for cycle 14. The head-end node knows only the amount of data that has been reserved in each cycle - not the order of the queue.

The head-end then issues a START command for cycle 12. The START command will occupy a slot and will be followed by the total number of empty slots requested by all reservations for cycle 12. The START command contains a cycle number. When a node receives this command, it has permission to send data into the next empty slot. It is allowed to send into as many empty slots as it reserved for cycle 12 in the previous RESERVE command. Notice that nodes B, C and D all have data waiting but are not allowed to send until START commands are received with a cycle number matching the reservation they made in a previous RESERVE command.

As soon as the head-end has finished generating the correct number of empty slots for cycle 12 it will immediately issue a START command for cycle 13. Notice that when the data is sent on the bus it is sent in the order in which reservations were made. This gives a much better FIFO characteristic than token controlled access.

In the meantime, at fixed intervals, the head-end continues to issue RESERVE commands. (Slots containing a RESERVE command may appear anywhere - they don't have to wait for the end of a cycle.)

Of course there are limits. Each node can be allocated a maximum number of slots that it can RESERVE on any one cycle. (This is to stop a node "hogging" the bus. But when a node sends a block (frame) of data, it sends that data into contiguous slots. This means that a node *must* be able to RESERVE sufficient slots for the maximum sized frame that it can send.

10.2.7 Limiting the Access Delay

The above mechanism will guarantee FIFO operation of the global queue at very high utilisations. But is FIFO operation always appropriate? In a congested situation several high capacity nodes could make reservations of the maximum allowance on every RESERVE cycle. If another node now has data to send then it will have to wait until all the previously reserved data has been sent. In heavy load situations this could result in very long access delays. We are now saying that FIFO is *not* always the best strategy.

CRMA has two additional commands - REJECT and CONFIRM. These commands are used to implement a "backpressure" mechanism which limits the size of the forward global queue so that access delay can be bounded (contained within some limits).

The basic CRMA protocol described above is modified. The RESERVE command no longer implies certainty. The RESERVE command is a request from the nodes and must be accepted by the head-end. When the head-end sees the return of a RESERVE command it may CONFIRM the cycle (that is, accept the reservation),

REJECT the cycle (and all previous cycles that have not been confirmed), or START the cycle (if there are no cycles queued ahead).

This means that each node must now maintain three queues:

- A queue of data (by cycle number) for which confirmation has been received.
- A queue of data (again by cycle number) for which reservations have been made, but as yet no confirmation has been received.
- A queue of data for which a reservation has not yet been made.

The head-end node has a predefined limit on the allowed length of the global reservation queue. When a RESERVE returns to the head-end node it looks to see how many slots have been reserved.

- If the total number of slots reserved exceeds the limit, then the head-end confirms the reservations just received (sends a CONFIRM with the same cycle number as that of the RESERVE just received) but then issues a REJECT command to terminate any outstanding RESERVE commands. The head-end will then suppress issuing further RESERVE commands until the number of reserved slots drops below the limit.
- If the number does not exceed the limit (but there are other cycles pending), the head-end issues the CONFIRM command. After a predetermined time interval, the head-end will issue another RESERVE command.

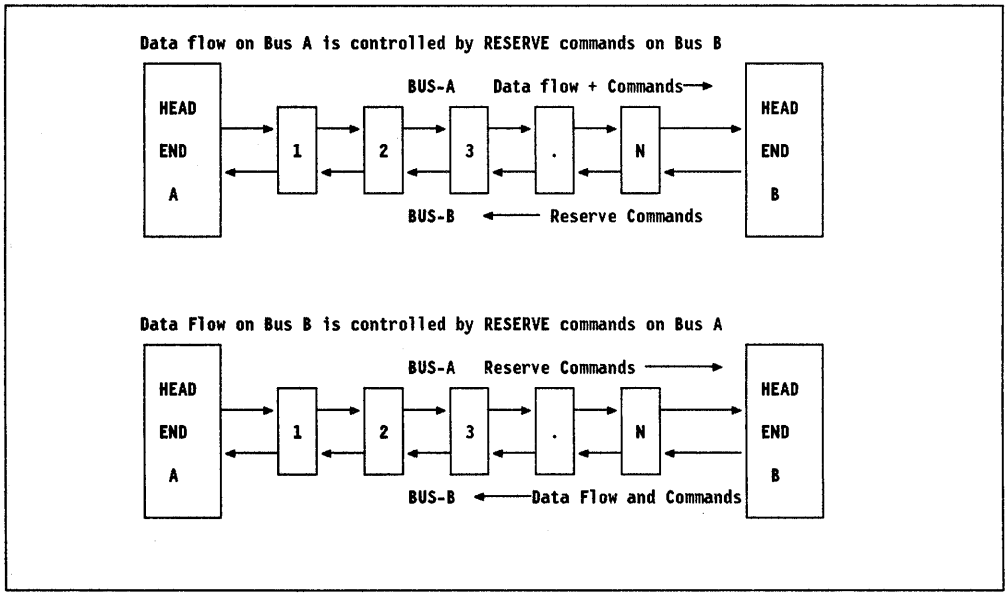


Figure 103. CRMA Dual Bus Configuration. The system is logically two separate parts. Data flow on one bus is controlled (allocated) by RESERVE commands on the other bus.

10.2.8 Dual Bus Configuration

CRMA can use a dual bus configuration as shown in Figure 103. The advantage of the dual bus configuration is that it can double the potential throughput. However, all the functions of the single bus configuration must be doubled (which adds significant cost to the adapter). In addition the nodes must know the location on the bus of all other nodes. (For example, referring to the diagram, if Node 3 wants to send to Node 1 then it must use Bus B; if it wants to send to node N then it must use Bus A.) To do this there must be an information exchange protocol so that each node can discover the location (upstream or

downstream) of each other node that it may send to (in order to determine which bus to send on). The upstream or downstream location may change when the bus is reconfigured so that a node location table must be built whenever the bus is initialised (or reconfigured).

As illustrated, operation can be thought of as taking place in two independent halves - the protocol is completely symmetric. Considering only data transport on Bus A:

- RESERVE commands are sent out by head-end B on Bus B.
- CONFIRM, START and REJECT commands are sent out on Bus A by head-end A.
- Head-end A generates the cycles on Bus A.
- The only modification of the protocol is caused by the fact that the head-end node generating the RESERVEs cannot know the state of the global queue. In the folded bus case, when a RESERVE is returned such that the amount of data requested pushes the outstanding total requests for data above the limit, the head-end rejects any RESERVEs that may be in progress and stops issuing RESERVEs until the amount of reserved slots falls below the limit again.

  In the dual bus case it can't stop the reservation process because the head-end node generating the RESERVEs does not know the status of the applicable global queue (that's at the other end of the bus). So, when head-end A rejects a RESERVE (and all RESERVEs currently in progress), head-end B immediately reissues a RESERVE with the same cycle number as the rejected one. When this RESERVE arrives at the other head-end, it may be confirmed or cancelled depending on the current status of the global queue.

Operation for data transport on Bus B is exactly symmetric.

10.2.9 Priorities

A priority scheme can be implemented by associating each cycle number with a priority. That is, there might be a cycle 2 for priority 1 and a cycle 2 for priority 2. RESERVEs would be issued separately for each priority at very different rates. This could lead to cycle 2,110 at priority 1 interrupting cycle 3,125 at priority 2. There is no necessary link between cycle numbers at each priority.

The priorities would operate with high priorities preempting the lower ones. Thus, a priority 1 START command (and the slots associated with it) could be issued in the middle of a sequence of vacant slots being generated for priority 2. So, the whole protocol is repeated at each priority level and higher priorities preempt lower ones.

10.2.10 Characteristics

As a result of the method of operation CRMA exhibits the following characteristics:

- Efficiency. Cycles may be scheduled with no time gaps between them so there is no time wasted on the LAN looking for a device that has data to send next.
- Speed insensitivity. The protocol results in high bus utilisations even at very high data rates (gigabit per second speeds).

- Because of its slotted structure the protocol is easily extendable to handle isochronous traffic. In this case the head-end node would generate premarked slots for isochronous traffic at predetermined intervals. These slots would be ignored by the CRMA data transfer protocol. (This is the way isochronous traffic is handled in DQDB.)

The protocol is especially suitable for gigabit per second LANs and MANs that are geographically long and have a large number of attached nodes. For short LANs with a small number of nodes there would seem to be little advantage over a token passing approach. In long LANs at very high speed there is a significant advantage.

10.3 CRMA-II

Cyclic Reservation Multiple Access - II (CRMA-II) represents the frontier of on-going LAN and MAN research. Although not implemented, the protocol has been extensively studied and simulated. It was developed as a result of the experience gained from the CRMA and MetaRing prototype projects.[75]

It should be noted that CRMA-II (like CRMA and MetaRing) is a Medium Access Control (MAC) protocol. There are many other functions that must occur on a real LAN or MAN that are not part of a MAC protocol. These are primarily management functions such as error recovery, monitoring, initialisation and the like.

10.3.1 Objective

CRMA-II uses many detailed features of the existing LAN/MAN protocols already discussed, especially CRMA and MetaRing. Each of these had characteristics which were very desirable and other characteristics which needed to be improved. The objective of CRMA-II is to adopt the best features of these protocols so as to arrive at the best possible result.

Cyclic Reservation

The cyclic reservation principle of CRMA has proven excellent at high utilisations. Access delay for low utilisation nodes has a strict upper bound and fairness of access is extremely good. But:

1. CRMA is less good at very low utilisations. The minimal access delay on a lightly loaded bus is the waiting time for a RESERVE and a START command - which corresponds to one network roundtrip delay.
2. There is no reuse of slots on the bus once data has been received at the destination. On a bus with many active nodes, there is considerable potential for reuse of slots which increases the capacity of the LAN significantly.

Buffer Insertion

Buffer insertion (MetaRing) on the other hand is an excellent principle at low and medium utilisations - it gives low access delay and maximal reuse of LAN capacity. At high utilisations, however, some precautions must be taken:

1. A "hog" node can completely prevent its immediate downstream neighbors from transmitting. (This was effectively solved in MetaRing by the use of the SAT protocol).
2. At very high utilisations ring latency can become dominant if many nodes have data in their insertion buffers.
3. Access delay while minimal at low loadings can also become significant at high LAN utilisations.

CRMA-II uses the cyclic reservation technique of CRMA which is mainly active at high utilisations combined with the buffer insertion principle to allow for immediate access at low utilisations whilst still preserving slot contiguity in

[75] CRMA-II is the result of work performed at the IBM Research Division, Zurich, Switzerland. A list of journal articles and conference papers relating to CRMA-II may be found in the bibliography.

frame transmission. Operation takes place in both modes simultaneously at all times but naturally shifts from being predominantly one mode to the other. In this way CRMA-II gains the best aspects of both protocols. CRMA-II is less complex than CRMA.

10.3.2 Principles of CRMA-II

Perhaps the first principle of CRMA-II is generality. Most individual mechanisms of CRMA have been extended and generalised so that they can be considered independent of their original context.

10.3.2.1 Topology

CRMA-II is able to use ring, bus or folded bus topologies. The protocol is designed to allow the building of a common set of interface chips that can be used for any of the three topologies.

10.3.2.2 Physical Layer Coding

CRMA-II (as with MetaRing and CRMA) proposes an "8 out of 10" (8B/10B) coding scheme similar in principle to the 4B/5B code used in FDDI (see section 9.3.5.2, "Data Encoding" on page 190). This means that every 8-bit group is coded into 10 bits on the optical medium. Only bit combinations that have a mixture of "1" bits and "0" bits are allowed. This means that no string of longer than three consecutive bits is allowed to be either all ones or all zeros. This is done for three reasons:

1. It provides frequent transitions in the code to allow a receiver PLL to derive accurate timing from the incoming bit stream. (See section 2.2.6.1, "Phase Locked Loops (PLLs)" on page 20.)
2. It allows some valid (sufficient transitions) combinations that do not have a corresponding data value. These are used for delimiters and synchronisation.
3. It minimises the amount of high speed circuitry necessary to implement an adapter.

In CRMA-II this principle (of encoding groups of bits on the medium) is generalised. To simplify hardware, so called Atomic Data Units (ADUs) are introduced. Although the line coding is 8B/10B, the protocol is arranged such that the smallest unit of data that may be coded or decoded is either 16 or 32 data bits (that is, 20 or 40 bits on the medium). These units are called ADUs (Atomic Data Units). (32-bit ADUs are deemed appropriate for 2.4 gigabits per second operation). No coding or decoding operation takes place in CRMA-II on anything smaller than the 16 or 32-bit ADU *except* synchronisation. The synchronisation sequence is a unique 8B/10B code that forms the first 8 data bits of a delimiter ADU.

This is done because as the speed of the medium increases into the multi-gigabit range (5 gigabits per second is a practical speed today) the logic speed of available circuitry cannot keep up. Very fast logic (Gallium Arsenide technology) is very costly and not very dense. Slower logic comes in many shapes and sizes but recently a combination of "Bipolar" technology and the common CMOS technology (called "BiCMOS") which allows reasonably high speeds at containable cost has become available.

By using ADUs, the circuitry that handles the bit stream in serial form must operate at the speed of the medium and therefore it must continue to use expensive technology (such as GaAs). But as soon as a 40-bit group is received

it can then be processed in parallel (as a 32-bit data group) at 1/40th of the medium speed using significantly lower cost circuit technology.

This involves almost no loss of efficiency in coding.

10.3.2.3 Slots

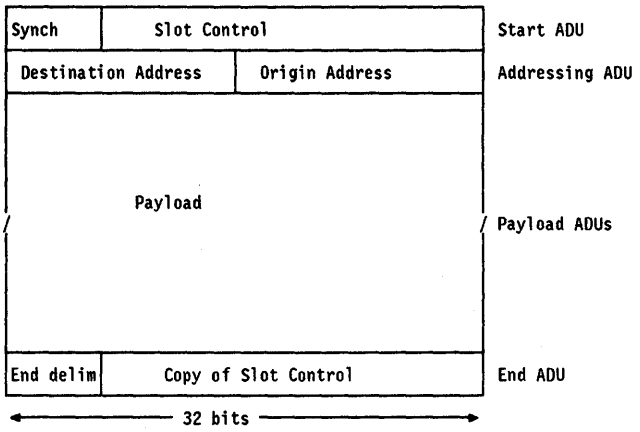


Figure 104. CRMA-II Slot Format

Data in CRMA-II is transferred in a slotted format to allow for capacity allocation and scheduling.

The principle involved is very similar to that used in DQDB, or CRMA, but there is a basic difference. In other LAN/MAN slotted systems a slot is a fixed entity identified by a delimiter (such as a code violation) followed by a fixed number of bits and immediately followed by another slot. That is, on the medium slots are synchronously coupled with one another. In CRMA-II a slot is fixed in size but special variable length frames carrying scheduling information are inserted as needed *between* slots. This means that slots are loosely coupled with one another.

A slot is delimited by a START ADU and an END ADU. The first 8 bits of the start ADU is a synchronisation character. The format for 32-bit ADUs is shown in Figure 104.

MAC commands which travel between the "scheduler" and the nodes *no longer "piggyback" in the slot headers of data slots*. They are carried as special (varying length entities) *between* regular data slots.

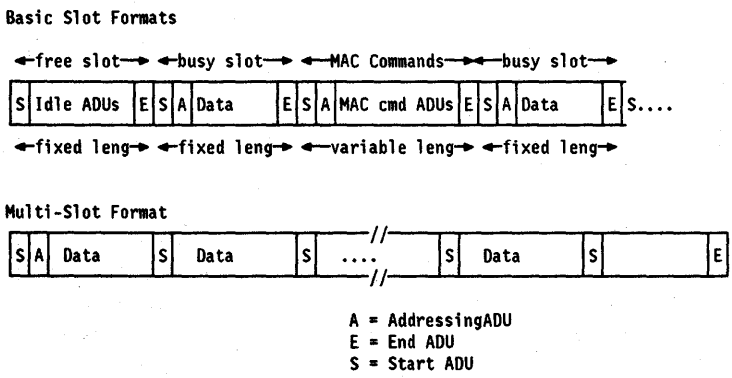


Figure 105. CRMA-II Slot Formats

When a node sends data over multiple slots the basic slot marking is maintained but much of the slot overhead is avoided by having only one END ADU (for the whole frame) and only one addressing ADU at the start of the frame. See Figure 105.

10.3.2.4 The Scheduler

The functions performed by the "head-end" node in CRMA are generalised into a capacity scheduling function in CRMA-II. The scheduler allocates capacity individually, on a node by node basis in response to requests made by the nodes during a reservation cycle. This differs from CRMA where the head-end node does not know about individual node demands - it only knows about the total capacity requested in each cycle. This is required to control fairness in the presence of slot reuse.

Information is exchanged with individual nodes by MAC commands of variable length (see Figure 105 on page 239). The scheduler sends out a RESERVE command as a special start/end ADU pair. Individual nodes that want to send data insert their requests (individually) between the Start/End ADU pair. When the frame returns to the scheduler, it contains an ADU from each node requesting a capacity allocation. The scheduler then responds by sending a CONFIRM command (delimited by a Start/End ADU pair) containing an individual response ADU for each node that made requests. In this way each requesting node is given an individual capacity allocation for this cycle.

In a ring network the scheduler function would normally be part of a generalised monitor function. In a ring network a monitor is necessary to detect and purge errored (permanently busy) slots etc.

10.3.2.5 Capacity Allocation Concept

The principles behind the allocation of capacity are:

1. Slot reuse. After a slot has been delivered to a destination, this slot becomes available for immediate reuse either by the destination node itself or by any other node.
2. Immediate access to unreserved slots. At periods of light loading, the necessary wait to receive a RESERVE followed by a CONFIRM could cause an unnecessary delay. If a slot is not currently being controlled by the scheduler and is "free" it can be used by any node that it passes.

Thus there are two types of slots "Reserved" and "Gratis". A Reserved slot is one that has been allocated by the scheduler. A Gratis slot is one that is not currently under the control of the scheduler. Slots dynamically change from Reserved to Gratis status and back again.

A Reserved slot may only be used by a node that has been granted a capacity allocation through a CONFIRM command from the scheduler. (Whenever the scheduler sends a group of CONFIRMs it immediately begins marking enough slots Reserved to satisfy the amount of capacity it just confirmed.)

A slot that contains data is always marked as Busy/Gratis even if it was previously Reserved. When the data is received by the destination (or back at the originator if it was a broadcast) the slot is marked Free/Gratis. In this state it may be claimed and reused by any node receiving it.

10.3.2.6 The Cyclic Reservation Protocol

Although CRMA-II will work in bus, folded bus and ring topologies, the protocol is most general in ring operation and therefore it is best studied in the ring configuration. The scheduler function may be active in any node on a ring but only in the head-end node of a bus.

The cyclic reservation protocol is similar to that of CRMA.

- The scheduler sends out a RESERVE command but, different from CRMA, every node that wants to make a reservation inserts its own request into the variable length RESERVE command which increases with each insertion.
- RESERVE requests contain the number of requested slots and the number of slot transmissions done by this node since the last reservation cycle.
- It is important to note that the scheduler does not know the identity of each requesting node. No information is kept by the scheduler about previous cycles or about previous requests from individual nodes. The scheduling decision is made based solely on information returned in the RESERVE command.
- The scheduler then decides how much data (how many slots) will be allowed to each node on this cycle.
- The scheduler then sends a CONFIRM command which contains an individual response to each requesting node. (The CONFIRM is a block of responses each one corresponding to a request made on the ALLOCATE. All outstanding ALLOCATE requests are replied to in the same CONFIRM.)
- Immediately after sending the CONFIRM command, the scheduler begins marking slots as reserved.

  In a ring topology this means marking Gratis slots. In a bus topology it means generating them.
- This cyclic reservation protocol is quite different from CRMA in that there is only one cycle in progress at any one time, so that no cycle number is needed and there is no need to correlate cycle start commands with previously made requests.
- When the CONFIRM is received by a node, it records the number received as its "confirmation count". This means that the node is now allowed to use this number of Free/Reserved slots as they arrive. The CONFIRM command shrinks (gets shorter) with each information removal.

10.3.2.7 The Scheduling Algorithm

The scheduling algorithm is quite intelligent. At periods of high loading some nodes may want to use much more than "fairness" would dictate. Because any node may use a Free/Gratis slot at any time these high throughput nodes could continue to use LAN capacity even when the scheduler wants to slow them down a bit.

This is prevented by the scheduler sending either a reserved slot allocation or a *"DEFERRED Allocation"* to each node.

A node with a deferred allocation may access only after it has let pass the indicated number of Free/Gratis slots.

This occurs only at times of extremely high loading.

10.3.2.8 Cycles

As described above individual reservation cycles are run one at a time with no reservation in advance for future cycles. The single cycle concept simplifies the reservation-based fairness control significantly because only two commands (Reserve, Confirm) are alternatively on the medium. This makes the protocol extremely robust and enables the system to recover from command failures without any additional command.

When one cycle finishes (when the scheduler has marked the allocated total number of slots as Reserved) the next cycle is started by issuing the Reserve command to collect requests from the nodes. The reservation cycle itself starts when the Reserve command has returned (after having circulated on the LAN) and the scheduler has allocated the reservations.

All this produces a significant gap between two reservation cycles (i.e. between the end of slot marking and the start of the next cycle). This however does not mean that the system looses throughput. On the contrary, transmissions continue to take place in Free/Gratis slots and since access to these slots are less restricted (only when a node must defer) system throughput must higher than for the case of back-to-back cycles. In fact, slots are only marked as reserved to correct unfairness and to guarantee a low bounded access delay.

10.3.2.9 Addressing

The system uses "short" addresses. A single ADU contains both destination and origin LAN addresses. This means that the full LAN/MAN address isn't used for sending data. During the initialisation procedure a set of local addresses are allocated. Each node keeps a table relating the real (long form) LAN address and the shortened addresses needed for send/receive operation.

10.3.2.10 Data Transfer

- When a node has been granted an allocation, this means that the node may use the allocated number of Free/Reserved slots. (A Free/Reserved slot is one that has been allocated by the scheduler but as yet has not been used by any node.) There are two ways in which a node may use a Free/Reserved slot. It may send data in that slot or it may use the slot to empty its insertion buffer (this subject is treated later).
- When a node puts data into a Free/Reserved slot it changes the slot status to Busy/Gratis. (Busy because it has data in it, Gratis because it has been used and therefore is no longer under allocation control of the scheduler.)
- When the Busy/Gratis slot is received by its destination, the data is copied and the slot marked Free/Gratis.
- A Free/Gratis slot is always available for use by any node receiving it (including the node that marked it Free/Gratis). So this slot may be immediately reused.

In operation, at very low loads, most slots will be Free/Gratis and may be used immediately by any node wanting to send. As the load builds up, the scheduler will begin getting allocation requests from nodes. The scheduler will begin marking passing Free/Gratis slots to Free/Reserved in order to make sure that requesting nodes can get the allocated capacity.

If the scheduler were to mark only Free/Gratis slots there would be an apparent problem here. Then, what if a node immediately before the scheduler on the ring decides to use all the passing Free/Gratis slots thus preventing the

scheduler from getting any slots to reserve? Therefore the scheduler does not only mark Free/Gratis slots to Free/Reserved it also marks passing Busy/Gratis slots (these are slots containing data) to the Busy/Reserved status. When a Busy/Reserved slot is received by a node, the Busy status is changed to Free resulting in the creation of a Free/Reserved slot that may be used by a node having a capacity allocation. So, ultimately the scheduler has caused the creation of the correct number of Free/Reserved slots. Thus when the scheduler allocates x slots it satisfies the allocation by immediately marking passing Gratis slots (either Busy or Free) to the reserved status.

When operation is examined two characteristics should be noted:

1. Some Free/Reserved slots will pass by the scheduler. This is fine and is a result of a Busy/Gratis slot being marked Busy/Reserved on its last trip past the scheduler and then later being marked Free by a receiving node.
2. The slot contiguity property, as given in CRMA requires an additional mechanism as described below. In CRMA, frames of data are sent in a contiguous stream (or block) of cells. Successive cells are used to transmit a frame until the end of that frame. Between the first and last cells of a frame of data *no* other data is allowed.

   In CRMA-II slot reuse causes the fragmentation of blocks of cells to the point where there is no way of guaranteeing any contiguous stream of free cells. This means that a node cannot know when there will be (or if there will be) a stream of consecutive cells in which to send a frame of any particular size.

10.3.2.11 ATM "Mode" of Data Transfer

If we wish to use the protocol as a basis for a distributed ATM (see section 7.1, "Asynchronous Transfer Mode (ATM)" on page 129) switch then nothing more is needed. The cell format would be:

- Start ADU.
- Address ADU (contains origin and destination short addresses).
- ATM cell (48 bytes of data with 5 bytes of header) in 14 ADUs. For ease of processing you might put the ATM header in the first two ADUs and the ATM data segment in the following 12 ADUs.
- End ADU.

Since the whole ATM concept is based on the principle of the asynchronous multiplexing of separate cells then the property of slot contiguity for frame transmission is not relevant.

10.3.2.12 Buffer Insertion

If we wish to use CRMA-II as a traditional LAN or MAN architecture the ability to send a frame of data into contiguous slots is very valuable. Without this ability a receiving node must maintain multiple frame reassembly buffers (and logic to reassemble different frames) so that it may receive from many senders "simultaneously". At the speeds involved, this is quite complex and expensive to do.

CRMA-II meets this objective by introducing the buffer insertion principle discussed above (see section 10.1, "MetaRing" on page 222).

In order to use this principle the node has a buffer large enough to accommodate the maximum sized frame *between* its receiver and its transmitter. This is shown below:

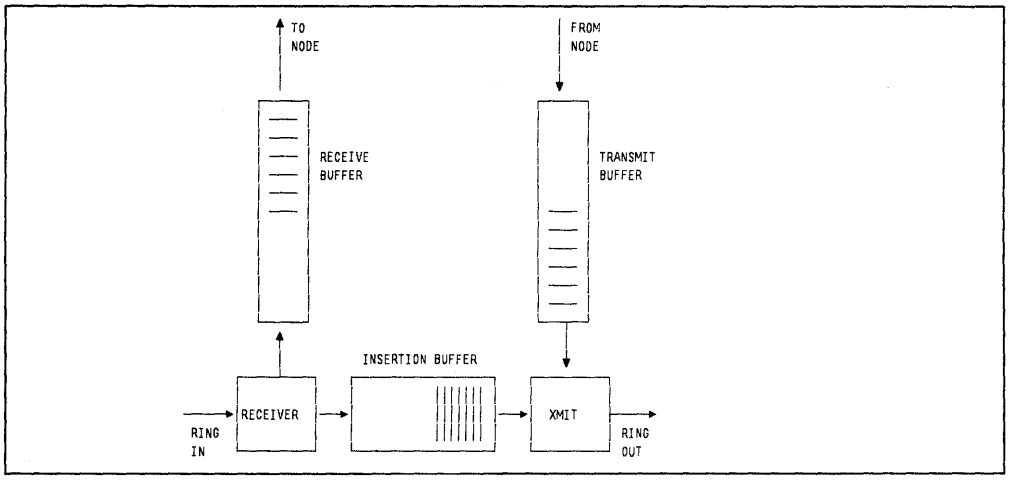


Figure 106. Principle of Buffer Insertion

The principle is basically the same as that described for MetaRing but with a few differences:

- In MetaRing the rule is that a node may start sending provided there is nothing being received from the ring *and* there is nothing in the insertion buffer.

  Because of the slotted transmission structure, the rule for sending in CRMA-II is that when a Free/Gratis slot (or a Free/Reserved slot if the node has reservations) is detected on the ring (or bus) segment *and* there is nothing in the insertion buffer, the node may commence sending.

- Data is sent as a contiguous frame but with interspersed Start ADUs at slot boundaries. This is illustrated in Figure 105 on page 239.

  When the data is received at the destination node the destination node reformats the multi-slot into a stream of single Free slots.

- While a node is transmitting a frame in this way, slots continue to arrive on its inbound side. If a Free/Gratis slot arrives then it is discarded. If a Free/Reserved slot arrives *and* the node has reservations then this slot too may be discarded (the node must of course decrement its allocation of reserved slots in this case).

- When Busy slots arrive (and/or Free/Reserved slots if the node does not have an allocation) they are held in the insertion buffer until the node finishes its transmission.

At the end of transmission of the frame, data is sent onto the ring from the insertion buffer and new slots arriving are entered into it. The insertion buffer also performs the function of elastic buffer to accommodate differences in the clock speeds at various nodes. The node is not allowed to send again until its insertion buffer becomes empty.

As operation continues:

- Incoming data from the medium is delayed because of the data queued ahead of it in the insertion buffer.

- When a Free/Gratis slot arrives it is discarded and since a slot full of data is now being transmitted from the insertion buffer this empties the insertion buffer of a slot full of data.
- In a busy ring, the node may find that the insertion buffer will not empty quickly enough by just waiting for Free/Gratis slots. In this case the node will request an allocation the next time the scheduler sends out a RESERVE command. If the node has more data to send it will request slots from the scheduler sufficient to both empty its insertion buffer and to send its next frame.
- When the node receives a CONFIRM command containing a slot allocation it will begin treating Free/Reserved slots in the same way as Free/Gratis slots and discarding them. This process empties the insertion buffer.
- Once the insertion buffer is empty, data passing through the node from upstream is no longer delayed.
- The node is allowed to send again as soon as it receives a Free/Gratis slot or (if it has an allocation) a Free/Reserved slot.

10.3.3 Summary

CRMA-II is designed to provide optimal fairness under varying load conditions on a very fast LAN or MAN. (The principle will work over a very wide range of speeds but the objective is to operate well at 2.4 gigabits per second.)

1. The buffer insertion protocol is used to provide almost instant access at low loadings and to allow for the sending of a frame of user data as a stream of contiguous slots.
2. The reservation protocol allows fairness in operation (and in particular low access delays) at from medium to very high utilisations.
3. Operation is such that both protocols operate at all times but one will tend to dominate the other depending on load conditions.
4. It exhibits the same properties as discussed for MetaRing with respect to efficient operation at any speed, throughputs well beyond the medium speed due to slot reuse, insensitivity to ring length and the number of nodes, as well as support of asynchronous, synchronous and isochronous traffic.

Chapter 11. Networks of LANs

High speed digital communication is already in very general use both for LAN operation and for the wide area interconnection of LANs.[76]

Many people feel that in the LAN bridges and routers of today we are seeing the beginnings of the generalised packet networks of the future.

Large LANs are usually composed of several rings or busses. These rings or busses are often referred to as LAN segments[77] or sometimes LAN "subnetworks". Due to such physical constraints as signal attenuation, propagation delays or noise susceptibility, LAN segments are limited in size or the number of stations that can attach to them. For example:

- A 4 or 16 Mbps IBM Token-Ring segment is limited to 260 stations using Type 1 or Type 2 media.
- An ANSI X3T9.5 FDDI LAN is limited to 500 stations.
- An IEEE 802.3 baseband 10BASE2 LAN is limited to a 925 m network.
- A 4 Mbps IBM Token-Ring segment is limited to 72 stations using Type 3 media.

LAN segments with similar or dissimilar physical media can often be combined to form a single large logical LAN.

An example of a LAN composed of four segments is shown in Figure 107. In this configuration, there are three token-ring segments and one IEEE 802.3 CSMA/CD segment interconnected using three bridges.

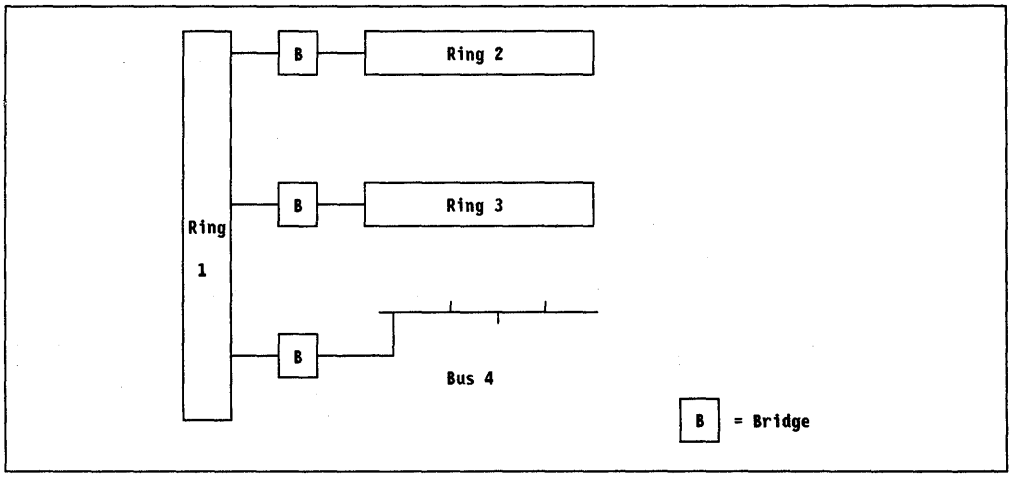


Figure 107. An Example of a Multisegment LAN

LAN segment interconnection can be achieved through bridges, routers and sometimes gateways, as discussed in section 11.2, "LAN Interconnection Techniques" on page 248. Bridges are frequently preferred as they provide the greatest degree of protocol independence and potentially provide optimal performance by interconnecting different LAN segments at the lowest possible

[76] This chapter is abstracted from *IBM Multisegment LAN Design Guidelines* by Roy Evans of IBM UK.

[77] In IEEE 802.3 and Ethernet LANs the term "*segment*" is used to describe the section of cable between repeaters. In this document LAN segment will be used to describe the bus or ring between bridges.

level in the protocol stack, that is the Medium Access Control (MAC) sublayer. By using MAC bridges, the interconnection allows multiple higher level protocols to operate concurrently in a multisegment LAN, such as SNA, TCP/IP and NetBIOS.

11.1 Why Interconnect LANs?

There are however many reasons for considering LAN segment interconnection. A prime reason of course is to expand the networking capability by:

- Increasing the overall size of the LAN and the number of attached devices or wiring closets above that supported on a single LAN segment.
- Providing connectivity between stations attached to different LAN segments so that segment differences (because of use of different MAC protocols, speeds or frequencies) are transparent to higher layer protocols.
- Increasing the bandwidth available to stations on a single segment by splitting a LAN segment into one or more bridged LAN segments. In this way fewer stations share the available bandwidth.
- Protecting users from the impact of wiring changes or media errors since bridges isolate errors or disruptions on one segment from the other segments they are connected to.
- Increasing the geographic area covered by the total LAN.
- Introducing bridges to control traffic through the use of filters.
- Supporting coexistence and interoperability between established LANs and LAN segments reflecting new technology such as FDDI.

11.2 LAN Interconnection Techniques

The techniques for general network interconnection may be classified into three categories as shown in Figure 108 on page 249. The numbered boxes refer to the seven layers of the OSI reference model, andan equivalent diagram could be drawn for SNA or TCP/IP protocol stacks. In broad terms a bridge is less protocol-specific than a router, which is less protocol-specific than a gateway. [78] A bridge is transparent to multiple concurrent higher-level protocols, such as SNA, TCP/IP, and NetBIOS, while a gateway such as Interlink's SNS SNA/Gateway™ is for specific protocols, in this case converting SNA 3270 protocols on a System/370* channel to DECnet™ protocols on a LAN.

Bridged local area networks support communication between stations attached to separate LAN segments, potentially with dissimilar medium access control protocols, as if they were attached to a single LAN. Above the DLC layer, bridged LAN segments appear as a single logical LAN. At the LLC layer, the LAN is relatively hardware independent, and data presented to the LLC could in principle be transmitted on any underlying or any future MAC layer.

A router offers interconnection at the network layer in the OSI reference model. When LAN segments are interconnected using this technique, the protocols that

[78] The terminology used does tend to vary depending on the protocol environment being used. This is especially true of the term 'gateway', which in TCP/IP networks is used to describe the device that performs an IP routing function, and many IBM products use the term gateway to describe the device that provides LAN access to a System/370 host.

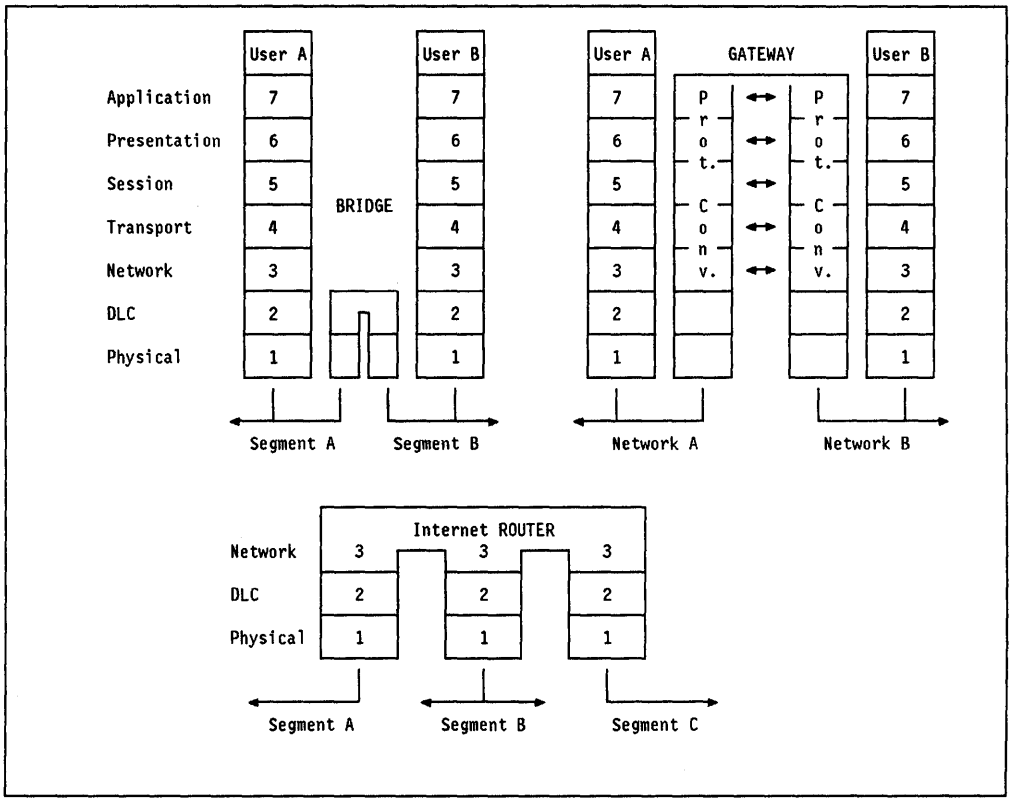


Figure 108. Network Interconnection Techniques (Based upon OSI Reference Model Layers 1 - 7)

traverse multiple segments must share a common network layer protocol, such as Internet Protocol (IP). The interconnection provided by this approach is not transparent to the LAN (data link) addressing structure. To access another segment across a router, the router must be explicitly addressed at the DLC layer.

Figure 108 shows a gateway as an interconnect device which supports more than one architecture or network addressing scheme to permit connectivity and interoperability between the devices and applications in the attached environments. Gateway products support mapping of addresses from one network to another, and may also provide transformation of data between the environments to support end-to-end application connectivity. Gateways usually link subnetworks at higher layers than bridges or routers. Gateways may operate at layers ranging from layer 3 to layer 7. The term *LAN gateway* is often used in a more general sense; it is used to describe a device that connects a LAN to another network using different protocols.

In a LAN environment, bridges are normally used to interconnect LAN segments, because of the low level at which this type of interconnection is established. There are however some LAN and protocol combinations that lead to other interconnection techniques being favored, for example, TCP/IP in a CSMA/CD environment lends itself to the use of routers as the interconnect device.

11.3 LAN Bridges

In summary bridges:

- Connect similar LAN segments into a single, larger logical LAN.
- They are application transparent.
- Are generally protocol-independent.

There are advantages in using bridges to interconnect LAN segments:

- They are simple compared to routers and so potentially have a low cost.
- They have potentially high performance because they have little or no requirement to interpret the data in a frame.
- They have a great deal of protocol independence.

11.3.1 MAC Layer Bridges

MAC layer bridges, as the name implies, relay messages from one LAN segment to another at the MAC sublayer level. MAC layer bridges consist of two (or more) physical and MAC layers (one for each LAN segment they interconnect). The MAC layer functions contained in the bridge are interconnected by a relay function which passes frames received from one MAC to the other MAC if certain forwarding conditions are satisfied. The relay function not only passes frames between the MACs but may also provide the protocol conversion necessary if the MAC protocols differ.

There are a number of different bridge methodologies. These include:

1. Source-Routing Bridges, described in section 11.4, "Source-Routing Bridges" on page 251.
2. Transparent Bridges, described in section 11.5, "Transparent Bridges" on page 260.
3. The proposed Source Routing Transparent Bridges, described in section 11.8, "Source Routing Transparent (SRT) Bridges" on page 271.

When there are multiple possible routes between segments, such as in the example shown in Figure 109 on page 251, the different bridging approaches lead to different *active* topologies.

Source-routing bridging, used by IBM Token-Ring Networks, allows **all** bridges in a multisegment LAN to be active concurrently. The different routes are distinguished by routing information in the MAC frame. Transparent bridging however, only allows a single route in the multisegment LAN to be forwarding frames at any one time. The single route is maintained by the bridges, using an algorithm known as the *spanning tree algorithm* which is described in more detail in section 11.6, "The Spanning Tree Algorithm" on page 264.

A possible *active* topology in Figure 109 on page 251 could be that Bridge 1, Bridge 2 and Bridge 3 are active and forwarding frames, while Bridge 4 and Bridge 5 are in an active standby state. The bridges that are in the standby state would only enter the forwarding state if another bridge became unavailable. Source-routing bridges also implement a single route through a multisegment LAN, also using the spanning tree algorithm. (This is *not* used by the bridge to determine the route for any data - that is determined by the sending station). The spanning tree is used by the bridge to allow it to send only one copy of a single route broadcast frame to each LAN segment.

In a source-routing LAN, the single route is used for a particular type of broadcast frame, known either as single-route broadcast frames or limited broadcast frames.

Source-routing and transparent bridges are described in more detail in the following sections.

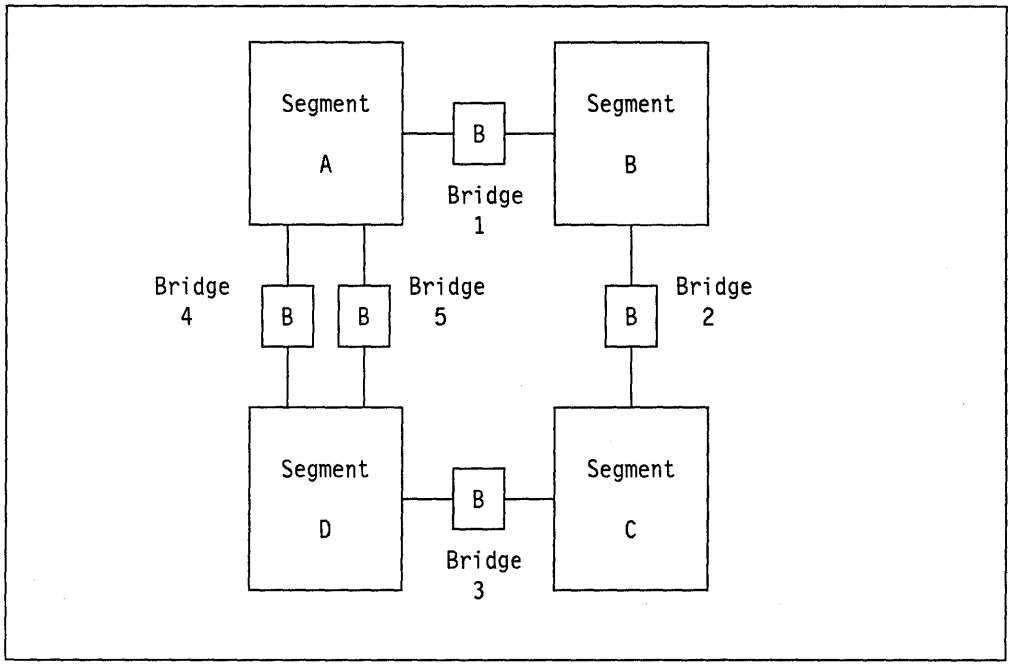


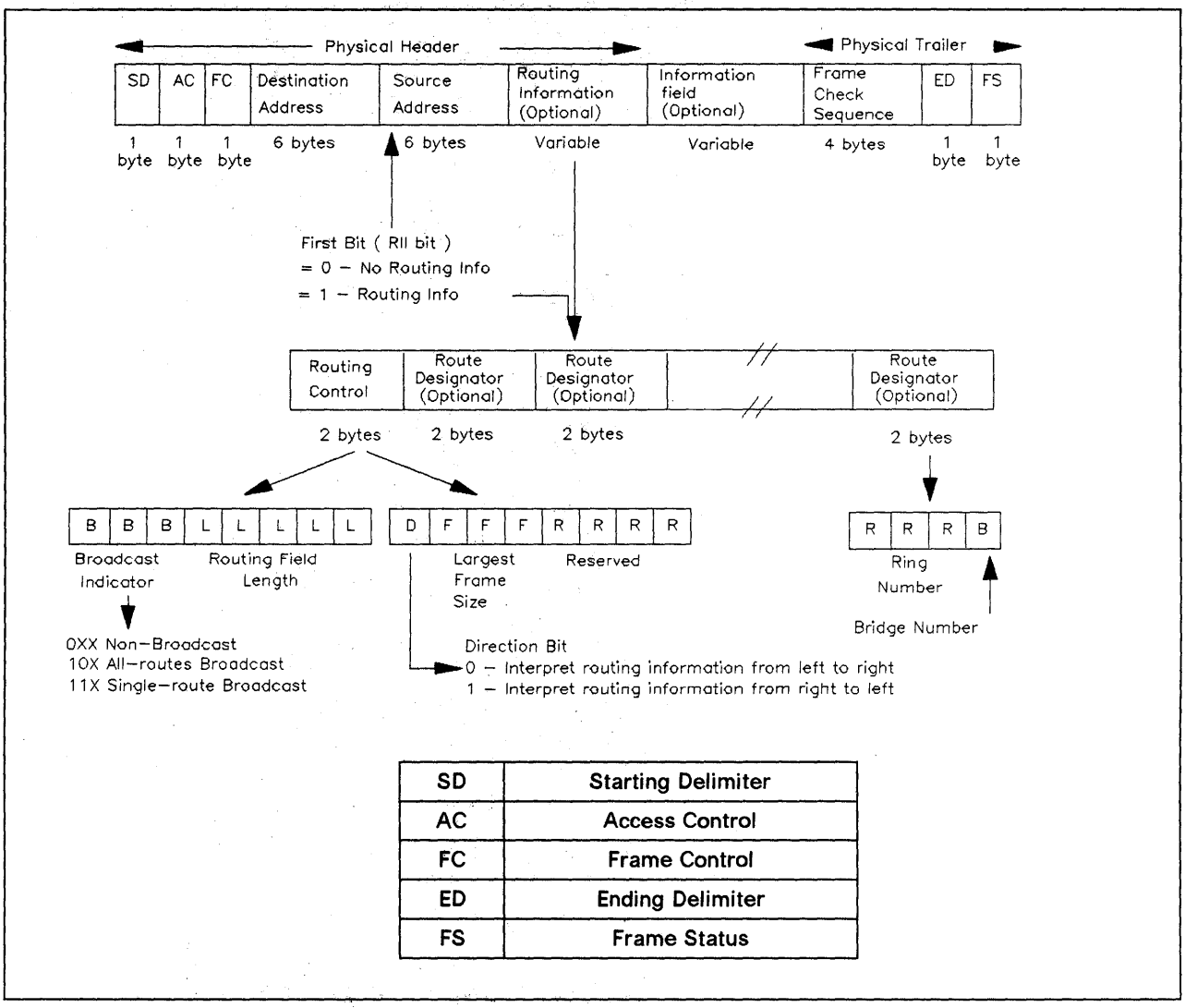
Figure 109. An Example of a Multisegment LAN

11.4 Source-Routing Bridges

Source routing is the method used in IBM Token-Ring Networks to control the route a frame travels through a multisegment LAN. Source-routing bridges put the responsibility of "navigating" through a multisegment LAN on the end stations. A route through a multisegment token-ring LAN is described by a sequence of ring and bridge numbers placed in the routing information field of a token-ring frame.[79] The routing information field in a MAC frame if present, is part of the MAC header of a token-ring frame. The presence of routing information means the end stations are aware of, and may make decisions based upon, the routes available in a multisegment LAN. By contrast, transparent bridges take the responsibility for routing frames through a multisegment LAN, which leads to more complex bridges, and end stations that are unaware of whether they are bridged, and if they are, what route they are taking.

Figure 110 on page 252 shows how the routing information is carried by a token-ring frame. If the Routing Information Indicator (RII) bit (the high order bit) of the **source** MAC address is set then the frame contains a routing information field.[80] The routing control field contains at minimum a 2-byte control field, and

[79] Each ring segment in a LAN has a ring number assigned when the bridges are configured. On an active ring it is held by the Ring Parameter Server management function. Bridges are also given numbers when they are configured. Together, the ring numbers and bridge numbers, known as route designators, are used to map a path through a multisegment LAN.



SD	**Starting Delimiter**
AC	**Access Control**
FC	**Frame Control**
ED	**Ending Delimiter**
FS	**Frame Status**

Figure 110. Routing Information of a Token-Ring Frame

optionally contains a number[81] of 2-byte route designator fields. Each route designator field contains a 3-digit ring number and a 1-digit bridge number. Together they map out a route through a multisegment token-ring LAN. Details on the other fields in Figure 110 may be found in the *IBM Token-Ring Architecture Reference*.

When a frame containing a routing information indicator is sent, all bridges on the LAN segment will examine the frame. The bridge will copy a broadcast frame as long as it hasn't already been on the bridged ring segment (the ring segment attached to the other half of the bridge) and it meets the adapter's single-route broadcast frame criteria. If the frame is a non-broadcast frame, the bridge interprets any routing information either left-to-right or right-to-left, depending on the value set in the direction bit and then forwards the frame if appropriate.

---

[80] The high order bit in the destination MAC address is used to indicate a group address. A group address field would be redundant in the source field because token-ring frames can only originate from a uniquely identified adapter.

[81] IBM implementation of the IEEE 802.5 standard limits the number of bridge hops to seven.

The broadcast indicators in the routing control field also control the way a bridge treats the frame. The types of broadcast frames are:

Non-Broadcast, also known as **Routed** frames. The frame will travel a specific route defined in the routing information field.

All-Routes Broadcast, also known as **General Broadcast** frames. The frame will be forwarded across the bridge provided certain conditions are met. These conditions are described later in this chapter. (See section 11.4.1.1, "All-Routes Broadcast Route Determination" on page 254.)

Single-Route Broadcast, also known as **Limited Broadcast** frames. The frame will be forwarded by all bridges that are configured to forward single-route broadcast frames. If the network is configured properly, a single-route broadcast frame will appear once on each LAN segment. (See section 11.4.1.2, "Single-Route Broadcast Route Determination" on page 254.)

Typically all-routes broadcast and single-route broadcast frames are used to discover a route during session setup. Once the route is established, non-broadcast frames are generally used.

11.4.1 Route Determination

Source-routing requires that the originating end station provides the routing information. When one station wishes to send data to another station, the sending station must first obtain a route to the destination MAC address, since a broadcast mechanism for all data transfer would affect the LAN performance. There are two common ways of determining the route between stations:

1. **All-Route** broadcast route determination
2. **Single-Route** broadcast route determination

These possible methods of route determination are usually two stage processes:

1. On-segment route determination
2. Off-segment route determination

Usually is used here because there is no formal method in IBM Token-Ring LANs of route determination. There are many ways of determining the route between two end stations in a source-routing LAN, but the on-segment/off-segment approach described is relatively common.

In the first stage:

- The source station sends a frame, usually a TEST or XID LLC protocol data unit (LPDU) onto the local LAN segment.[82] The frame either has no routing information field or is a non-broadcast frame without any route designator fields; in either case, the frame is not forwarded by any bridges on the local segment.
- The sending station then waits for some time (the time varies depending on the application) and if it does not obtain a response it goes to the second stage of route determination.

[82] This frame is usually sent to SAP 0, a null SAP which is opened automatically by adapters, that only responds to connectionless TEST/XID requests.

For the second stage the sending station resends the TEST or XID LPDU, this time with a stub routing information field with the broadcast bits set. This broadcast may either be an **All-Routes** broadcast or a **Single-Route** broadcast. An example all-routes broadcast flow is shown in Figure 111 on page 255 and an example single-route broadcast is shown in Figure 112 on page 256.

If no response is received from the target station during either the on-segment or the off-segment route determination stages, it is an application responsibility to retry or back out.

11.4.1.1 All-Routes Broadcast Route Determination

Figure 111 on page 255 is an example of a typical route determination scheme using all-routes broadcast frames. After receiving no responses from the on-segment route determination, the sending station issues an all-routes broadcast TEST or XID frame. The bridges forward the frame unless:

1. The frame has already been on the next segment.
2. Forwarding the frame would exceed the bridge all-route broadcast hop count limit in that direction. IBM limits the number of bridges in a path to seven. The hop count further limits how far a frame may travel in a network. In IBM source-routing bridges, hop counts are set (and may be different) in both directions.
3. If the bridge filter functions do not allow a frame to be forwarded.

The routing information field is built up as the frame crosses the bridges:

- The sending station provides the stub routing control field.
- The first bridge adds two route designator fields, the first is the starting ring/bridge combination, the second is the second segment number and a null bridge entry.
- Successive bridges then fill in their bridge number and add another two-byte designator field.

The routing information of the frame being forwarded through bridge A and bridge D in the example shown in Figure 111, would build up as follows:

After bridge A: The route designators would be 001A 0020.
After bridge D: The route designators would be 001A 002D 0030.

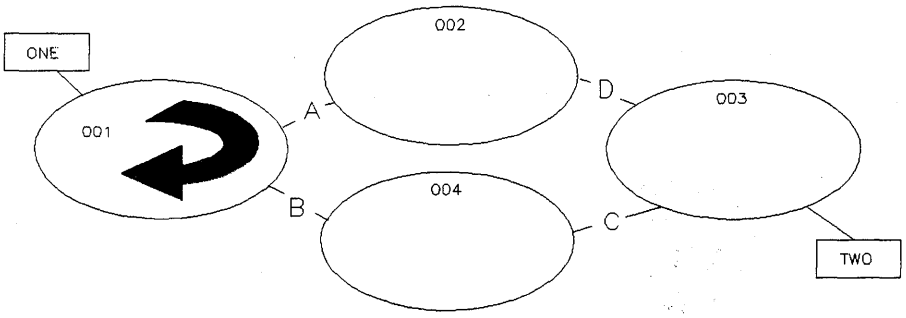
As many frames as there are routes will be received by the target machine. The target machine responds with non-broadcast frames, flipping the direction bit in the routing information field. The response frames then trace back through the network and arrive at the sending station. Usually the route chosen is the route contained in the first reply, although criteria such as the minimum number of hops or the supported frame sizes would be equally valid.

11.4.1.2 Single-Route Broadcast Route Determination

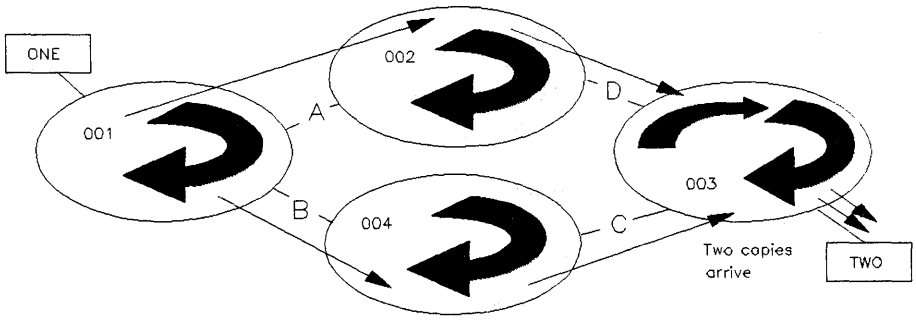
Some products implement single-route broadcast route determination for the second stage of route determination. The sending station resends an XID or TEST LPDU[83], with a stub routing information field. It also sets the single-route broadcast fields in the routing control field.

The primary aim of the single-route broadcast function of the IBM Token-Ring Network is to minimize the processing overhead of the target machine (or

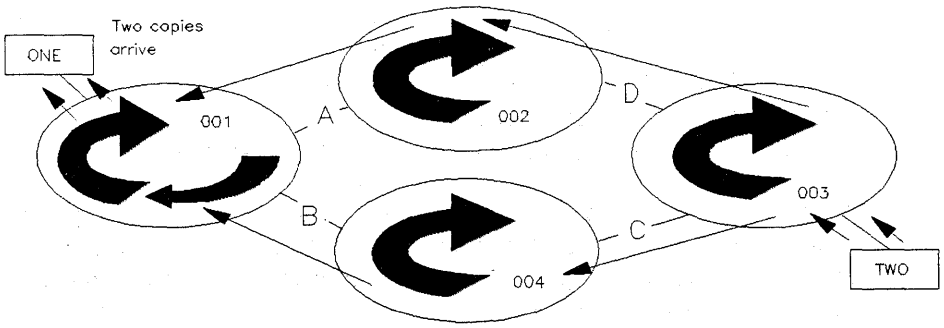
[83] Generally to the null SAP, SAP 00.



On-segment route determination. Station ONE issues a TEST or XID LPDU and waits for a response.



Off-segment route determination. An XID or TEST is issued with the all-routes broadcast bits set. Multiple copies reach the target station TWO.

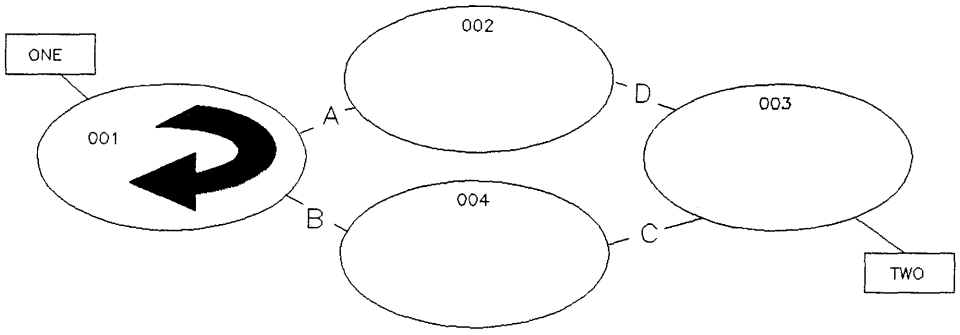


The target machine responds with non-broadcast frames that route back to the source. The routing information when the frames arrive at ONE is:

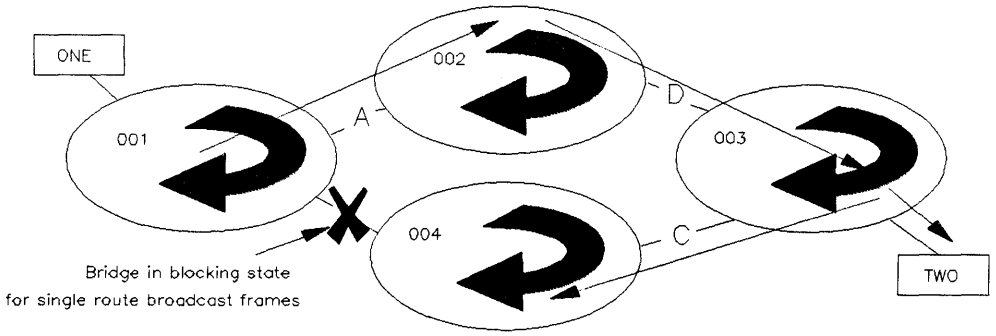
1. 001A-002D-0030
2. 001B-004C-0030

With the direction indicator set to 1 - right to left.

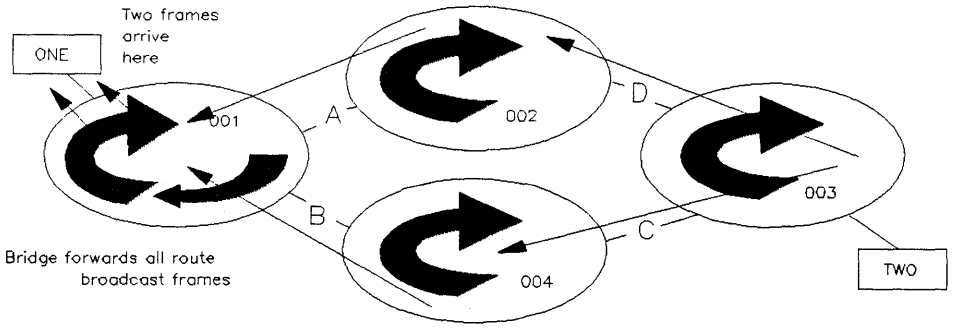
Figure 111. Example All-Routes Broadcast Route Determination. Station ONE is setting up communication with station TWO.



On-segment route determination. Station ONE issues a TEST or XID LPDU, and waits for a response.



Off-segment - A TEST or XID is issued as a single-route broadcast frame. One copy reaches each segment.



Receiving station TWO responds to the single frame received with an all-routes broadcast frame. Two copies are received by the sending station. The route information fields contain these routes:

1. 003D-002A-0010
2. 003C-004B-0010

With the direction bit set to 0 (that is, read left to right).

Figure 112. Example Single-Route Broadcast Route Determination. Station ONE is setting up communication with station TWO.

machines) by only allowing one copy of the broadcast frame on each segment in the LAN. An all-routes broadcast would cause as many frames as there are

possible routes to arrive at the target machine. The single-route broadcast function is particularly appropriate to LAN server functions, where the number of search requests is significant and where the processor overhead to service multiple broadcast frames could affect response times.

The propagation of single-route broadcast frames is limited by:

- Whether this bridge has been configured to forward single-route broadcast frames.
- Whether the frame has already been on the next segment, as noted in the route designator fields built by the bridges the frame has already passed through.
- Filters at the bridge.

The single route is derived from either:

1. Static definition resulting from bridge installation parameters.
2. Single-route broadcast routes derived dynamically from the automatic spanning tree algorithm (which in turn uses some bridge installation parameters). The spanning tree algorithm is described later in section 11.6, "The Spanning Tree Algorithm" on page 264.

On receipt of the single-route broadcast frame, the destination station usually issues an all-routes broadcast frame, directed at the source address. The original sending station receives as many frames as there are routes in the network, from which it usually chooses the route taken by the first frame for subsequent communication.

11.4.1.3 Largest Frame Size Supported by a Bridge

In a multisegment LAN, bridges are only capable of handling certain-sized frames. The largest frame size supported is implementation dependent, and may differ from the largest frame supported by the end stations and the largest frame supported by the LAN segments that are being interconnected. Source routing provides a mechanism where the end stations can learn the maximum frame size.

The routing control field contains an entry for the maximum frame size allowed. This field is initially set by the originator of the broadcast frame to B'111'. During the forwarding process, a bridge examines this value. If it is higher than that supported by a particular bridge, the bridge lowers the value in the field to the maximum it can support. As broadcast frames are forwarded across the bridges in a path, the maximum frame size allowed for the particular route is obtained. Table 3 shows the allowed frame length values.[84]

Table 3 (Page 1 of 2). Maximum Frame Sizes in the Routing Control Field

Largest Frame field	Size (Bytes)	Comments
B'000'	516	Minimum for 802.2 LLC Type 1 (connectionless) operation
B'001'	1500	Largest frame size supported by 802.3 LANs
B'010'	2052	Typical 24x80 full screen application

[84] Many applications and devices do take this field into account and adjust the frame sizes they use accordingly. These include applications and devices like OS/2* EE, Personal Communications/3270, NetBIOS, 3174 and 3745.

Table 3 (Page 2 of 2). Maximum Frame Sizes in the Routing Control Field		
Largest Frame field	**Size (Bytes)**	**Comments**
B'011'	4472	Largest frame size supported by FDDI
B'100'	8144	Largest frame size supported by 802.4 LANs
B'101'	11,407	
B'110'	17,800	
B'111'		Used by all-routes broadcast frames.

11.4.1.4 Other Route Determination Techniques

There are many other possible mechanisms for route determination. For example a DOS LAN requester of IBM OS/2 LAN Server V1.2 follows the sequence described below:

1. The DOS LAN Requester issues a number of NetBIOS ADD_NAME_QUERY commands using single-route broadcast frames to the NetBIOS functional address. The purpose of this frame is to ensure the DOS LAN Requester's NetBIOS name is unique throughout the network.
2. The DOS LAN Requester then issues a NetBIOS NAME_QUERY command, again using a single-route broadcast frame to the NetBIOS functional address. This frame is used to search for a target NetBIOS name. In this example it would be the name of the Domain Controller.
3. The OS/2 LAN Server responds with an all-routes broadcast NetBIOS NAME_RECOGNIZED frame.

Another example is provided by IBM OS/2 TCP/IP. The PING function issues an all-routes broadcast to the all-stations broadcast address (FFFFFFFFFFFF).

IBM OS/2 TCP/IP PING function is using the Address Resolution Protocol to resolve an Internet address into a MAC address. ARP packets are sent to the broadcast Internet address (the address on that network with a host part of all binary ones), in an IEEE 802 environment this maps onto the broadcast IEEE 802 address (of all binary ones), so all stations, TCP/IP or otherwise, receive the ARP requests. To quote from the RFC:

> The dynamic address discovery procedure is to broadcast an ARP request. To limit the number of all ring broadcasts to a minimum, it is desirable (though not required) that an ARP request first be sent as an all-stations broadcast, without a Routing Information Field (RIF). If the all-stations (local ring) broadcast is not supported or if the all-stations broadcast is unsuccessful after some reasonable time has elapsed, then send the ARP request as an all-routes or single-route broadcast with an empty RIF (no routing designators). An all-routes broadcast is preferable since it yields an amount of fault tolerance. In an environment with multiple redundant bridges, all-routes broadcast allows operation in spite of spanning-tree bridge failures. However, single-route broadcasts may be used if IP and ARP must use the same broadcast method.

The OS/2 TCP/IP ARP request traced during the compilation of this document showed a single all-routes, all-stations ARP request with a routed response, which is within the bounds specified in the RFC.

Another interesting aspect of RFC1042 is the flexibility it allows when multiple responses to an address resolution request are received, each having taken a different route through a multisegment token-ring LAN. An individual TCP/IP implementation could do one of the following:

1. Take the first response and ignore the rest.
2. Take the last response, (that is, always update the ARP cache with the latest ARP message).
3. Take the response with the shortest path, (that is, replace the ARP cache information with the latest ARP message data if it is a shorter route).

11.4.1.5 Source Routing Summary

With source routing, if a bridge should become unavailable while devices are communicating through it, it is the responsibility of the end stations to find another route. Implementation of this type of recovery is product dependent. For example, if there are parallel routes and one should fail:

- With IBM Personal Communications/3270, the SNA session will drop but a new route will be established.
- IBM LAN program 1.3 will reestablish the NetBIOS Sessions automatically, without operator intervention.
- NCP V5 will reestablish a new route with another NCP without loss of the INN sessions.

An understanding of route determination can aid the planning of filters in bridges. Filtering is a bridge capability that can:

- Reduce broadcast traffic across bridges.

  From a performance standpoint, especially in a remote bridge environment, an excessively large number of broadcast frames crossing LAN segments may be undesirable.

  For example, TCP/IP's use of all-routes all-stations broadcasts for route determination may cause unnecessary load on some network devices.

  Note: Clearly, in a large multisegment LAN, an application's use of broadcast techniques needs to be understood, with filters or LAN topology being designed accordingly. The TCP/IP example given here is just for the address resolution phase; however, there are LAN applications available that make a more general use of broadcast techniques.

- Reduce the processor load on machines with the NetBIOS functional addresses.

  NetBIOS opens a functional address when it becomes active on a machine. Applications use single-route broadcast frames addressed to this NetBIOS functional address for name resolution, and all machines with the NetBIOS functional address open process these frames. In a large network, the overhead of processing these frames from all other NetBIOS nodes may be unacceptable.

- Allow more flexible NetBIOS naming conventions.

  NetBIOS names are unique on the LAN, or at least unique on the part of a multisegment LAN a particular station may communicate with. In a LAN environment, users may prefer to have NetBIOS names like "PETE" or "MARY", rather than have unique 16-character names in a large network.

11.5 Transparent Bridges

Transparent bridging is the bridge routing scheme drafted by the IEEE 802.1 (Internetworking) committee.

The transparent bridging philosophy is that the end stations should be completely unaware of the fact that a LAN contains bridges, so the responsibility for routing frames through a multisegment LAN is passed completely to the bridge.

Transparent bridging assumes the use of a topology with a single active route. This does not preclude a multiple route physical topology. Transparent bridges maintain a single active route in a multisegment LAN by using the spanning tree algorithm. Figure 115 on page 265 shows an example of a possible physical topology and a possible active topology maintained using the spanning tree algorithm.

Transparent bridges do not use or build routing information fields in the frames. Instead they copy **all frames** on the segments they are attached to, and examine the source and destination addresses. Source-routing bridges by comparison, only copy frames with the RII (Routing Information Indicator) bit set. The source addresses are used by transparent bridges to build a two-sided table, in which the source address is on "*this*" side of the bridge, and the destination address is on "*that side somewhere*".

The decision to forward the frame is based on a routing table. This table is often referred to as the *filtering database*.

- If the destination MAC address appears on the **SAME** side of the routing table it is not forwarded.
- If the destination MAC address appears on the **OPPOSITE** side of the routing table, the frame is forwarded.
- If the frame does not appear in the routing table, forward the frame.[85]

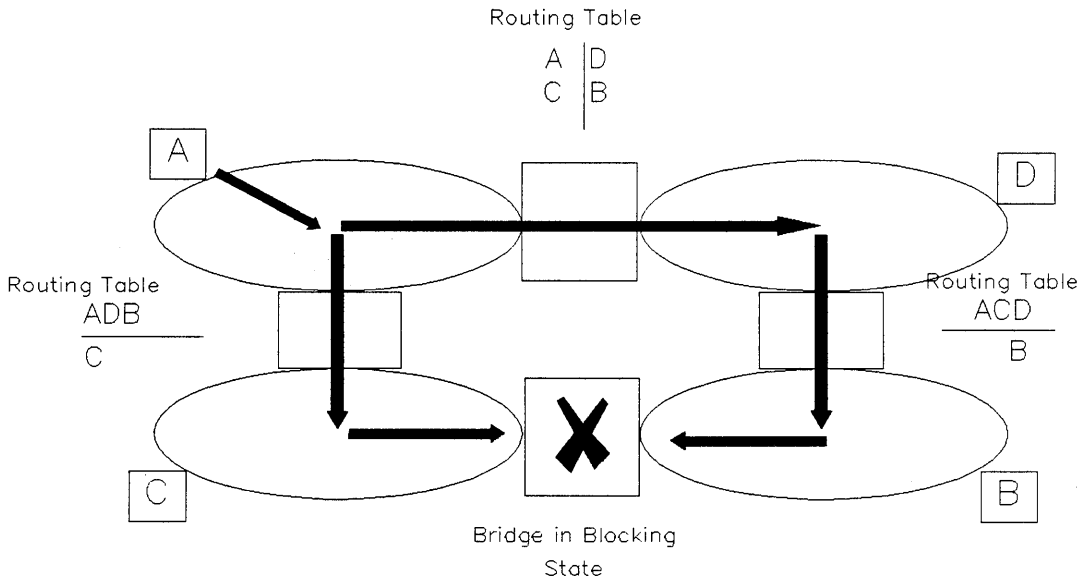
Transparent bridges copy all frames, and compare all frames against entries in their routing tables. Transparent bridges require fast processors, often implemented using parallel processing techniques, with the routing table lookup implemented in hardware rather than using software-based hashing techniques.

Other criteria may be used to make the decision to forward, in the same way as filters are used in IBM source-routing bridges.

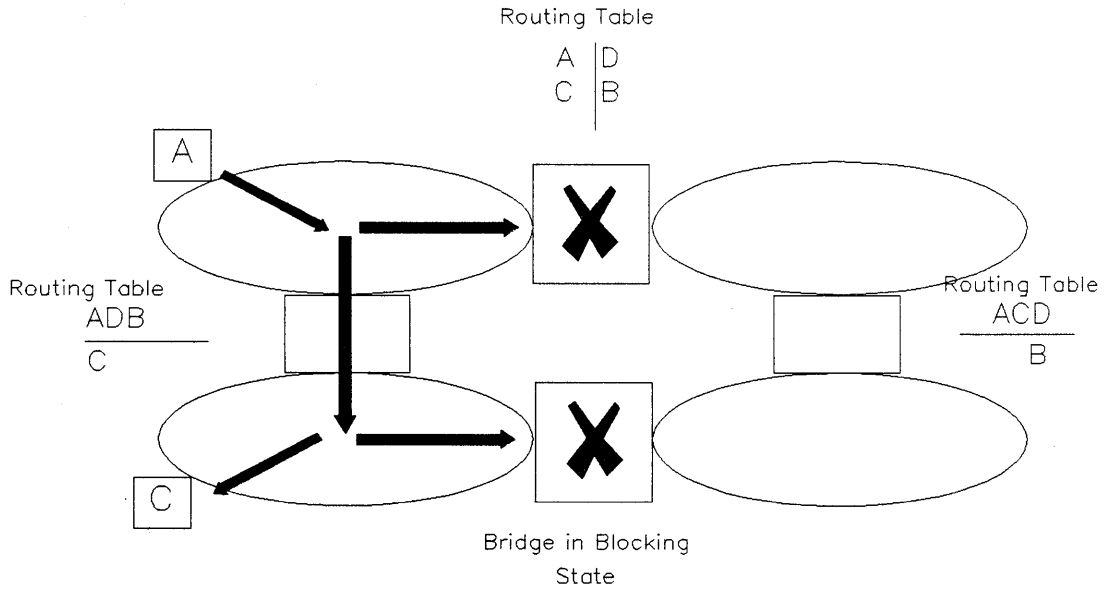
Unlike source routing, there is no mechanism to discover the largest frame size supported by the bridges on the active route. Transparent bridging is a technique used in IEEE 802.3 LANs, which support only 1500 byte frames.

When a frame arrives on a particular side of a bridge, the source address can be used to update "*this*" side of the routing table. The destination address is not used in this learning process because the bridge can only know which side of the bridge the destination address is on when it is the source of a transmitted

[85] Normally the frame is forwarded. This approach is used when the bridge learns the topology of the network. An alternative approach that is sometimes used in small networks, is to have a fixed filtering database. In this case, the routing table is manually configured, and frames for unknown destinations would be ignored.



The bridges "LISTEN TO SOURCES"; in this example, A is transitting to an unknown destination. The routing tables will be updated with the direction of A.



The bridges "FILTER FAMILIAR, FORWARD FOREIGN"; in this case, A is sending a message to C.

Figure 113. Routing in a Multisegment LAN Using Transparent Bridges

frame. When the destination station responds, the bridges in the network learn on which *side* the station is, modifying their routing tables accordingly.

Normally there is a combination of fixed and learned entries in the routing table (the filtering database), because fixed tables alone would become unwieldy in large networks. The dynamic routing table entries contain an aging timer, so that if a station hasn't been heard from for a certain time period, its entry can be deleted. This process minimizes the size of the routing table, thereby minimising the amount of memory required and the processing required to search for it.

Figure 113 on page 261 summarizes the transparent bridging process.

11.5.1 Why a Single Route in a Multisegment LAN?

Transparent bridging requires a single active route in a multisegment LAN. Some of the reasons for this apply to any LAN which uses source-routing or transparent bridging. The single route considerations include:

1. Unnecessary network load, together with additional coding logic at the end station to determine which is the "real" message so it may respond to that one alone. The premise of transparent bridging is that applications should be completely unchanged for a bridged environment; additional logic to handle multiple frames would constitute a change, unless the protocol already catered for this eventuality. TCP/IP is an example of a protocol that does cater for the receipt of multiple copies of data. In a transparent bridging environment the end stations have no mechanism to control, or know about, the route a frame took through the network, so if multiple copies of frames arrive the end stations would have to compensate for this.
2. If there are parallel routes in an 802.3 CSMA/CD network, collisions on the target LAN are almost inevitable, leading to the normal backoff/retry situation. Backoff/retry would cause queuing in the least desirable place, the bridge.
3. If there are closed loops in the network, stations can be regarded as being on both sides of the bridges in that loop. Without additional logic in the learning process, the new destination will always be on the "that" side of the bridge, as the source is on "this" side. In this case the discovery frame will circulate for ever. As there is no routing field, there is no hop count and no way to stop the circulating frame. The routing table would now think the original source is on both sides of the bridge, so if the destination sends a response to the source, or any other station sends a message, the frame would not be forwarded across the bridges in the parallel route, as the station would be viewed as being on the "same" side, even though it isn't. Figure 114 on page 263 shows an example of a simple closed loop.

In summary, in a transparent bridging LAN there must be a single route between any two stations. This requirement for a single route between any two stations is implemented as a single route for **ALL** stations for all network data. Compare this with a source-routing LAN where the requirement is that only some broadcast route set-up traffic crosses a single route. At any one time there can only be one active route through a LAN using transparent bridging though there is still a requirement for surplus bridges to provide backup paths. This leads to the requirement for some automatic single route selection mechanism for transparent bridged LANs. The process uses the "spanning tree algorithm". IBM Token-Ring LANs may also use the spanning tree algorithm to initialize a single-route broadcast route.

A Simple Closed Loop in a Transparent Bridged LAN

In this example, both X and Z are new stations on the LAN and BOTH bridges can forward frames. This example illustrates the importance of a single active route in a multisegment LAN using transparent bridges.

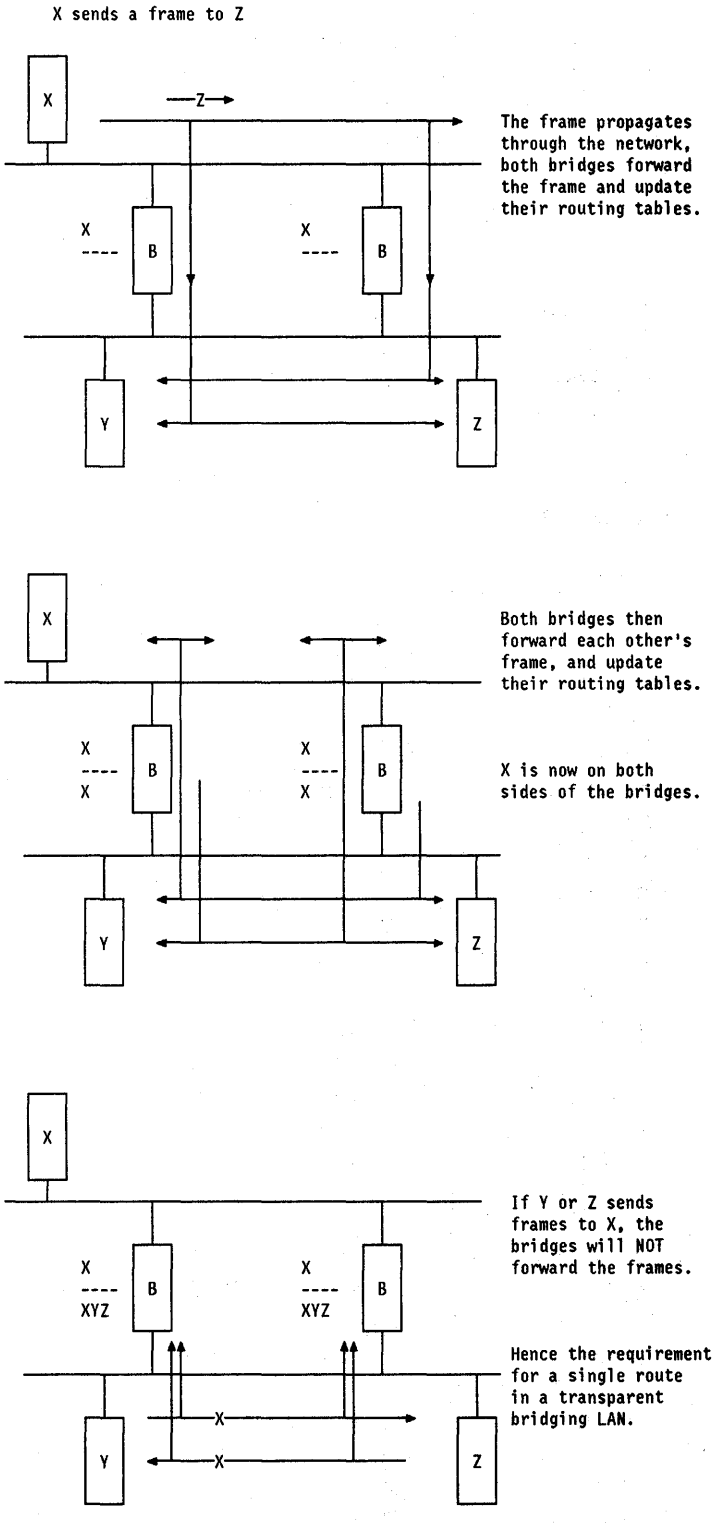


Figure 114. Closed Loops in a Transparent Bridging Environment

11.6 The Spanning Tree Algorithm

In a multisegment LAN, transparent bridges use the spanning tree algorithm to maintain a single active route. This algorithm, which is defined by the IEEE 802.1 committee, is also used by IBM source-routing bridges to maintain a single route for single-route broadcast frames. While both transparent bridges and source-routing bridges use the same spanning tree algorithm it is important to note that the functional address used by transparent bridges to communicate between one another and the functional address used by source-routing bridges is not the same. This means that in a network that contains both transparent and source-routing bridges, single route broadcast configuration will be done by all transparent bridges and separately by all source-routing bridges. This section describes the process whereby a single route is obtained and maintained in a multisegment LAN.

The process for providing a single route uses a spanning tree algorithm. A tree is a pattern of connections with no loops; "spanning" because it is the tree that spans (connects) all of the subnetworks. In transparent bridging there is a clear distinction between the active topology and the physical bridged topology, reflecting the use of the spanning tree algorithm. Figure 115 on page 265 shows the physical configuration of a sample LAN as well as a possible active LAN.

In an active topology, frames are forwarded through those bridge *ports*[86] that are in **forwarding** state. Other bridge ports that do not forward frames are held in **blocking** state. A bridge that is in blocking state may be put into forwarding state if the topology of the network changes. An application program will be unaware that the active topology of the network is changing, except if the device was communicating through a bridge that failed. Then applications would experience extended network delays.

The spanning tree topology is described in terms of:

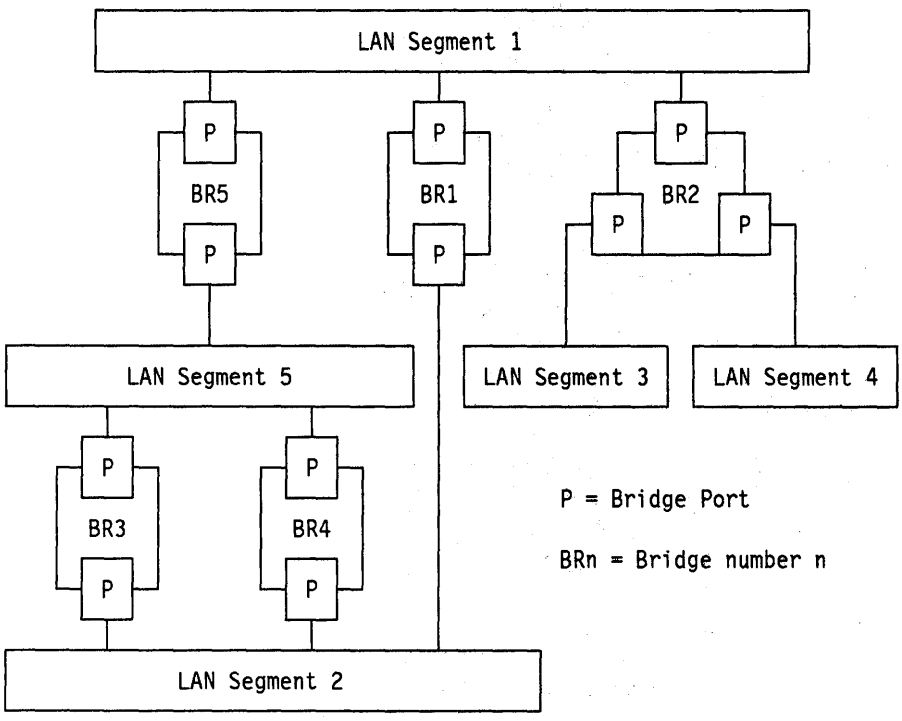
Unique bridge identifier: Each bridge in the network is assigned a unique identifier. This is a combination of the MAC address on the bridge's lowest port number and a two-byte bridge priority level, in essence a bridge number. The bridge priority is defined at installation.

Port identifier: Each adapter on the bridge is uniquely defined within the bridge with this two-byte value. The spanning tree is capable of obtaining routes in LANs with bridges with more than two ports.

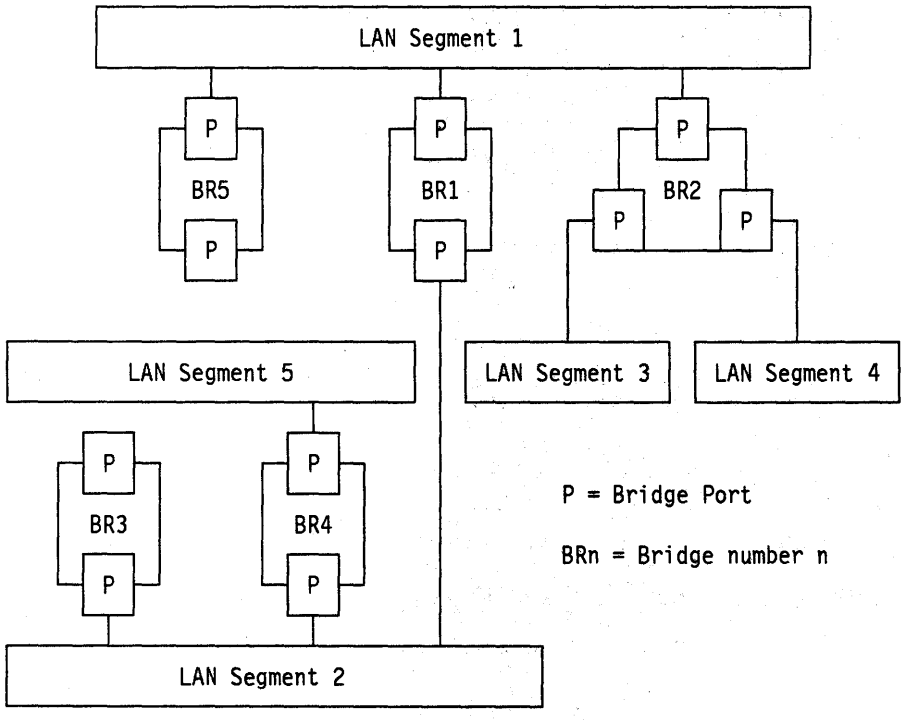
Root bridge: The bridge with the lowest value of the bridge identifier is assigned as the root (the "top") of the spanning tree. Potentially this bridge carries the greatest traffic, as it connects the two "halves" of the network together.

Path cost: In an interconnected network there are potentially preferred routes and less preferred routes. If there is an option to use a fast bridge or a slow bridge, the use of the faster would be preferable. If there is an option to use a heavily loaded LAN or a lightly loaded LAN, the lightly loaded route may be preferable. This leads to the idea of a cost being associated with each port of a bridge, the higher the value, the less preferred the route. At each bridge, the cost of transmission

[86] The spanning tree terminology refers to ports. A bridge port and a bridge LAN adapter are synonymous.



Bridged LAN - Physical Topology



Bridged LAN - Active Topology

Figure 115. Example of Physical Topology and a Possible Active Spanning Tree Topology

through each port is added to give the total cost of transmission for a particular path to any LAN from the root bridge.

Root port: Is the port in the direction of the least path cost to the root bridge.

Root path cost: From each bridge there are potentially many different paths to the root bridge; one of them has minimum cost - the root path cost.

Designated bridge: For each LAN segment only one bridge will be in the forwarding state. This is the designated bridge for that LAN. All other bridges on a particular segment will be in blocking state and will not forward frames or participate in the address learning process.

Designated port: The port attached to a LAN that provides a minimum cost path to the root bridge. All traffic from the LAN the bridge is attached to, and all the traffic from lower level LANs travel through this port.

The spanning tree is constructed by:

1. Determining the root bridge.
2. Determining the root port on all other bridges.
3. Determining the designated port on each LAN (the port with the minimum root path cost). If the root path cost is identical for more than one bridge on a segment, then the bridge with the highest priority is chosen.

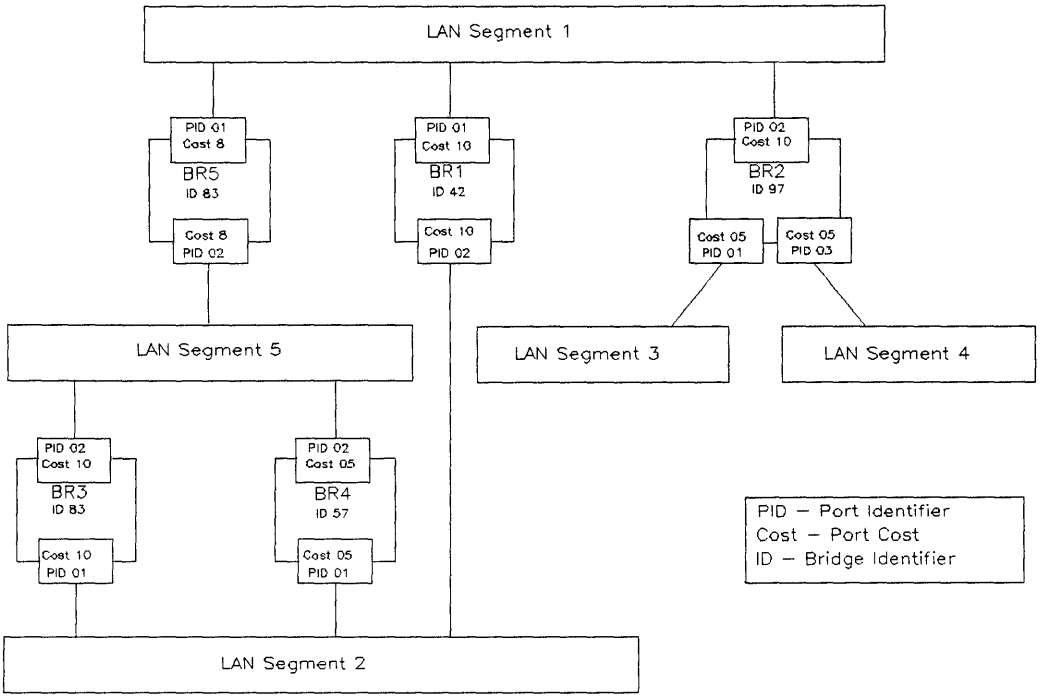
Figure 116 on page 267 shows an example of a physical topology and the corresponding logical tree topology.

In the transparent bridged network, after the active topology has been learned, each bridge assumes one of three roles:

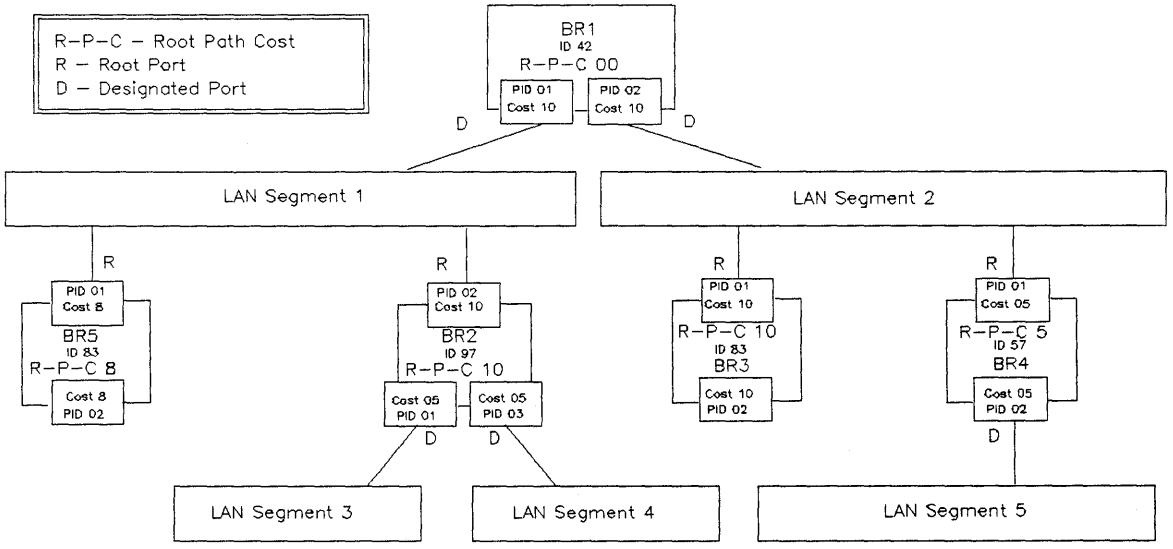
1. The root bridge:
   - The root bridge is the bridge with the lowest identifier.
   - The root bridge will forward frames.
   - The root bridge is responsible for periodically transmitting a "HELLO" Bridge Protocol Data Unit (BPDU) to all the LAN segments to which it is connected.
2. A designated bridge:
   - Will forward frames.
   - The responsibility of the designated bridge is to recognize and receive "HELLO" BPDUs from the root bridge, update the path cost and timing information in each message and forward the BPDU across the bridge.
3. A standby bridge:
   - Will **NOT** forward frames.
   - It is the responsibility of the standby bridge to monitor "HELLO" BPDUs but not to update or forward them. As bridges enter and leave the network, a standby bridge may be needed to assume the role of the root or designated bridge. The HELLO BPDUs will indicate when this is necessary.

The bridge topology is propagated with Bridge Protocol Data Units (BPDUs). These BPDUs are transmitted using connectionless protocols to a group MAC address 800143000000/SAP_42 combination that all the bridges recognize. These bridge protocol data units contain data including:

- The root identifier - the bridge that the transmitter of the BPDU believes to be the root of the spanning tree
- The root path cost - to the bridge believed to be the root bridge
- The bridge identifier - of the sending bridge
- Message age



Bridged LAN - Physical Topology



Bridged LAN - Logical Spanning Tree Topology

Figure 116. Spanning Tree Example

- A time-out parameter used to judge the failure of the spanning tree.

Using these frames the topology of the spanning tree is maintained.

When new bridges enter the network, they enter assuming they are the root, but in the listening state. There is an iterative process while the role of the new machine and the new topology is determined.

Similarly when a bridge in the spanning tree is removed, and after the bridge's maximum age time-out has expired, there is a similar iterative process to reconfigure the spanning tree.

Because of the propagation delays involved in the topology exchange protocol, a sharp change from one active topology to another is not possible. Therefore wait times are provided by the protocol as well as intermediate port states. The possible port states are:

- Blocking - no frame forwarding, no address learning, no protocol participation, except ensuring that another bridge has claimed responsibility for forwarding frames onto that segment.
- Listening - no frame forwarding, no address learning, but participating in the spanning tree algorithm.
- Learning - no frame forwarding, but building the address tables from passing frames and participating in the spanning tree algorithm.
- Forwarding - forwarding frames, building address tables, and participating in the spanning tree algorithm.

The root bridge generates BPDUs which are relayed by the bridges that are in a forwarding state. The root sets network-wide values for some timers. The bridge maximum age timer is used to detect the failure, and hence the requirement to reconfigure the spanning tree. The BPDU also contains the bridge forward delay timer, which is the time a bridge will stay in the listening state and the time spent in the learning state before moving to the forwarding state. In the IBM 8209 LAN Bridge, the maximum age timer defaults to 15 seconds, while the bridge forward delay defaults to 20 seconds.

11.6.1 Spanning Tree Algorithm Used on IBM Token-Ring

The spanning tree algorithm can be used by IBM Token-Ring bridges to allocate a **single-route broadcast** route through a multisegment token-ring LAN. The objective is to limit the number of broadcast messages received by a target machine to conserve its processor cycles.

The spanning tree algorithm used is identical to the spanning tree algorithm used in transparent bridging. Both comply with the spanning tree algorithm defined in the IEEE 802.1 Part D standard. While both transparent bridges and source-routing bridges use the same spanning tree algorithm it is important to note that the functional address used by transparent bridges to communicate between one another and the functional address used by source-routing bridges is not the same. This means that in a network that contains both transparent and source-routing bridges, single route broadcast configuration will be done by all transparent bridges and separately by all source-routing bridges. The learning stage, required by the standard, is very brief as there are no addresses to learn! The various states a bridge may be in are:

- Blocking
- Listening
- Forwarding.

When in **blocking** or **listening** state, the bridge will still forward all-routes broadcast frames as well as non-broadcast frames with routing information, which will be the majority of the network traffic.

The spanning tree protocols used are the same, but a different MAC address is used; with IBM Token-Ring bridges, the bridge functional address C00000000100 is used.

11.7 Parallel Routes

The major difference in LAN topology between a multisegment LAN using transparent bridges and a multisegment LAN using source-routing bridges is that source-routing bridges support active parallel paths.

- Active parallel paths between the two end stations allows:
  - Active redundancy for availability and load balancing.
  - Duplexed-backbone topology.
- Mesh configurations in general allow:
  - Shorter mean path lengths between station pairs.
  - Greater distribution of traffic and hence less congestion near the root of the spanning tree.
- Parallel bridges between the same two LANs allow:
  - Incremental additions of bridge throughput, if low-throughput bridges make good business sense or are the only possible technical solution (as between very high-speed LANs).
  - Far simpler implementation than in transparent bridging, which requires special (and non-standard) protocols between the bridges in parallel.
- Parallel remote bridges between the same two LANs allow:
  - Incremental additions of link throughput when low-speed links across public networks are used.

Explicit routing information provides:

- Better problem determination by end stations, bridges or LAN managers in case of a bridge or LAN outage.
- Potential end station selection of routes according to route characteristics such as:
  - Number of hops.
  - Maximum frame length allowed.
  - Security.
- The potential for end stations to tune operational parameters according to route characteristics such as:
  - Timers based on route length.
  - Maximum frame size.
- The potential for route servers, which via remote management could perform active load balancing.

Figure 117 on page 270 shows an example network with a number of different LAN media. Each different media type supports a different maximum frame size, added to which some of the bridge links may also limit the maximum frame size allowed. During the route discovery process in a source-routing LAN, the

maximum frame size supported on a particular route is obtained. Without source-routing support, the frame size would need to be set to the lowest common denominator in the entire multisegment LAN, in order to facilitate any-to-any communication.

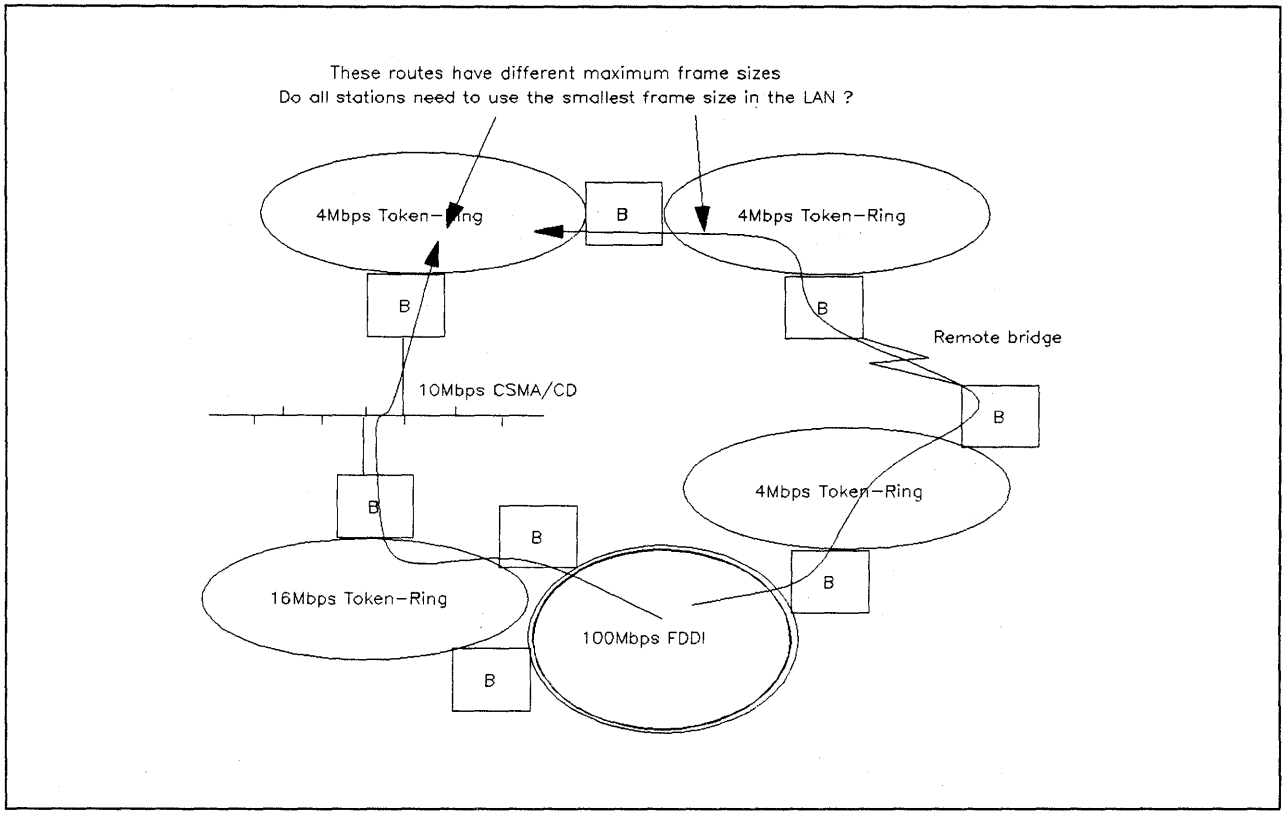


Figure 117. Maximum Frame Size Supported by Multiple Routes

11.8 Source Routing Transparent (SRT) Bridges

Source routing has the advantage of being able to support parallel routes across a multisegment LAN. This allows traffic load to be shared across bridges or shared across LAN segments bypassing heavily utilized components, as well as giving the end station the opportunity to make some decisions based on the current topology of the LAN.

The IEEE 802.1 (Internetworking) committee has a requirement that source-routing bridges/stations must be capable of interoperating with transparent bridges/stations on the same network. Proponents of source-routing and transparent bridging schemes have argued the benefits of both approaches to bridging, while the standards of their operation have been developed by two subcommittees within the IEEE.

- Source routing: IEEE 802.5, moving towards an International Standard as an addendum to ISO 8802.5.
- Transparent Bridging: IEEE 802.1 Part D, now a Draft International Standard, DIS 10038.

One of the problems of interoperability is the implications of the two techniques in the design of end stations as well as the design of the bridges. There may be occasions when two stations need to communicate, where one understands and uses routing information fields, and the other station does not. The goal of interoperability will not be achieved unless the source-routing station lowers its functionality to that of the transparent bridging station.

At a March 1990 meeting of the IEEE 802 committee, IBM proposed the idea of a Source-Routing Transparent (**SRT**) bridge, that would solve many of the interoperability problems. The SRT proposal is currently being evaluated, but no formal status has yet been granted.

An SRT bridge would be based on the concept of a transparent bridge but would include source routing as a tower on top of the base function. The transparent function would base its forwarding decision on a routing table while the tower would base the forwarding decision on a routing information field. So, if a routing information field exists - use it.

An SRT bridge would form a single spanning tree with all other SRT and transparent bridges. If the spanning tree allowed it, the bridge would forward frames that did not contain routing information fields. If the frame contained a routing information field, the frame would be forwarded in the same way as existing source-routing bridges, even if it were in blocking state from the point of view of the spanning tree.

SRT stations would be able to utilize the source-routing path if one existed, falling back to the spanning tree path if there were no alternatives.

For today's source-routing stations, minor changes in behavior would be required to make them SRT stations. During the route determination process a source-routing station may transmit a single-route broadcast frame that would traverse the active spanning tree, arriving at each segment once. The first frame from a transparent bridging station would arrive at the target segment once, as the frame would be forwarded over the spanning tree path to the destination. The result, the arrival of a single initial frame at the target segment, would be

the same. So instead of a two-stage on-segment/off-segment route determination, SRT stations would use a single on-segment *route explorer* frame.

The target SRT station would respond once to the route explorer frame, with a single-route broadcast frame without any routing information. The originating SRT station would then choose an appropriate route, or fall back to the spanning tree path. Transparent bridging stations would not respond to the frames with routing information present.

Single-route broadcast messages would be used in an SRT environment.

The SRT proposal also separates source routing from the token-ring definition, making it significantly easier to incorporate routing information into other standards, such as FDDI.

In summary:

- An SRT bridge
  - Shares a single spanning tree with all other transparent bridges
  - Forwards group addressed frames on the spanning tree
  - Is a transparent bridge for frames **without** routing information
  - Is a source-routing bridge for frames **with** routing information
- An SRT station
  - Sends group addressed frames **without** a routing information field
  - Sends a route explorer frame **without** a routing information field
  - Sends a route explorer response **once** without routing information, as an all-routes broadcast frame
  - Uses explicit routing information where available, falling back to using no routing information as would occur in a transparent bridging environment.

At the time of writing this document, the SRT idea was still a proposal. Significant changes may still occur before the concept proceeds further in the standards process.

11.9 Routers

Another way of connecting LAN segments is with a router. A router connects separate networks that use the same transport protocols. Bridges by contrast typically connect networks with the same LLC protocol.

Routers are commonly used in TCP/IP based LANs, where they are known as IP routers or Internet gateways. A RISC System/6000 or a PS/2* running OS/2 TCP/IP could provide the TCP/IP routing function between an IEEE 802.3 LAN segment and a token-ring LAN segment. Another example of a router is the IBM LAN-to-LAN Wide Area Network Program, where the higher-level protocol is NetBIOS, and two LANs are connected via an SNA or X.25 network.

Figure 118 shows the function of a router relative to the seven layer OSI model.

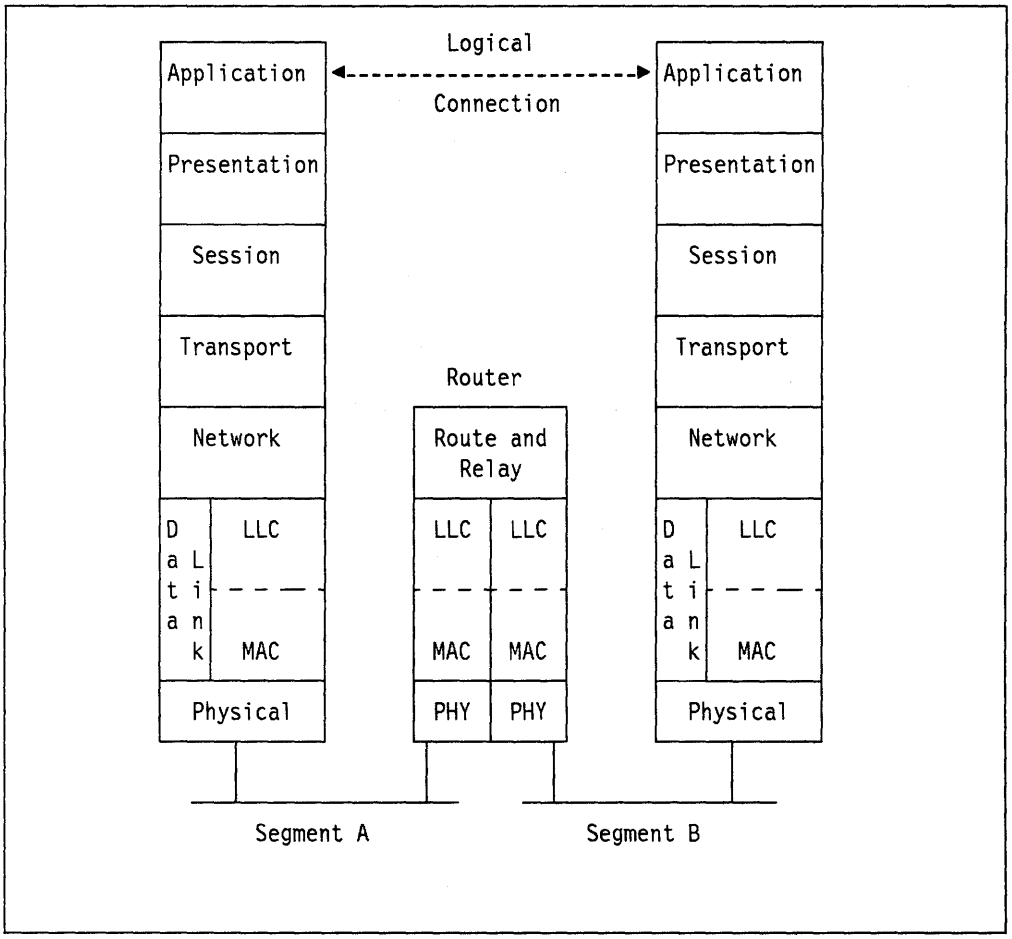


Figure 118. Model for a Router. Relationship between the OSI seven-layer model and a model for a router.

Routers determine how data frames are to be routed using information from within the frame. So in TCP/IP, the IP address would be used, while in NetBIOS the NetBIOS name would be used as a base for the routing decision.

Routers are application transparent[87], but protocol dependent. There are however a number of routers available that cater for a wide variety of protocols, so protocol dependence is often diminished.

A router maintains a table of destinations in a way similar to a transparent bridge. This allows a router to "know" where a destination is, without having to find it each time a new session is established. These tables are often dynamically exchanged with other routers in the network, using a router-to-router protocol. In TCP/IP, common protocols are:

GGP Gateway-to-gateway protocol

EGP Exterior gateway protocol

RIP Routing information protocol

Hello Hello protocol.

There is a derivative of a router, called a brouter, a bridge router. This device would behave like a router if it recognized the format of a frame and the frame contained routable information, and would behave like a bridge if it didn't.

11.10 Summary

LAN topology may require multiple segments for a variety of reasons. There are two major approaches to routing within such a multisegment local area network, each with advantages and disadvantages in some environments. Transparent bridging requires a single active route in a multisegment LAN and is closely aligned to CSMA/CD LANs. Source routing supports single and multiple route LANs, and in a token-ring-based LAN offers significant flexibility for both routing and traffic management. In a multisegment LAN, there are no fixed rules to the available topologies; therefore, design is very important.

[87] Application or protocol timers may need adjusting because of the use of routers.

Appendix A. Review of Basic Principles

The objective of a network is to provide connections between end users using shared facilities (lines, nodes etc.).[88]

There are several general techniques available for the sharing of a facility between different data connections and/or voice circuits. These are general techniques and apply (albeit with different levels of efficiency) to each element of the system separately, so they are described here first.

A.1 Available Techniques

A.1.1 Frequency Division Multiplexing

This technique is exactly the same as is used for radio or television broadcasting. A sender is allocated a range of frequencies between which a signal may be sent and information may be encoded on that signal using a range of "modulation techniques". The receiver must be able not only to receive that frequency but also to decode the modulation technique used. On a cable, or on a microwave carrier, the available band of frequencies is limited but the principle is still the same. The amount of information that can be carried within a frequency "band" is directly proportional to the width of that band and is also dependent on the modulation technique used. There are theoretical limits that cannot be avoided such that every frequency "band" has a finite limit. Because of the necessary imprecision of the equipment involved, there are "buffer" zones (guard bands) allowed between bands so that one band will not interfere with either of the adjacent ones. The size of these buffer zones is also determined by the modulation technique (you need a lot less for Frequency Modulation (FM) than for Amplitude Modulation (AM)) and by the precision (and hence cost) of the equipment involved.

Frequency division multiplexing has, in the past, found use in telephone systems for carrying multiple calls over (say) a microwave link. It is also the basis of cable TV systems where many TV signals (each with a bandwidth of 4 or 7 megahertz) are multiplexed over a single coaxial cable. It is also used in some types of computer Local Area Networks (LANs).

Frequency division multiplexing is also sometimes called "broadband multiplexing".

[88] This appendix is condensed from "A Plain Man's View of Voice and Data Integration" (IBM GG24-3029).

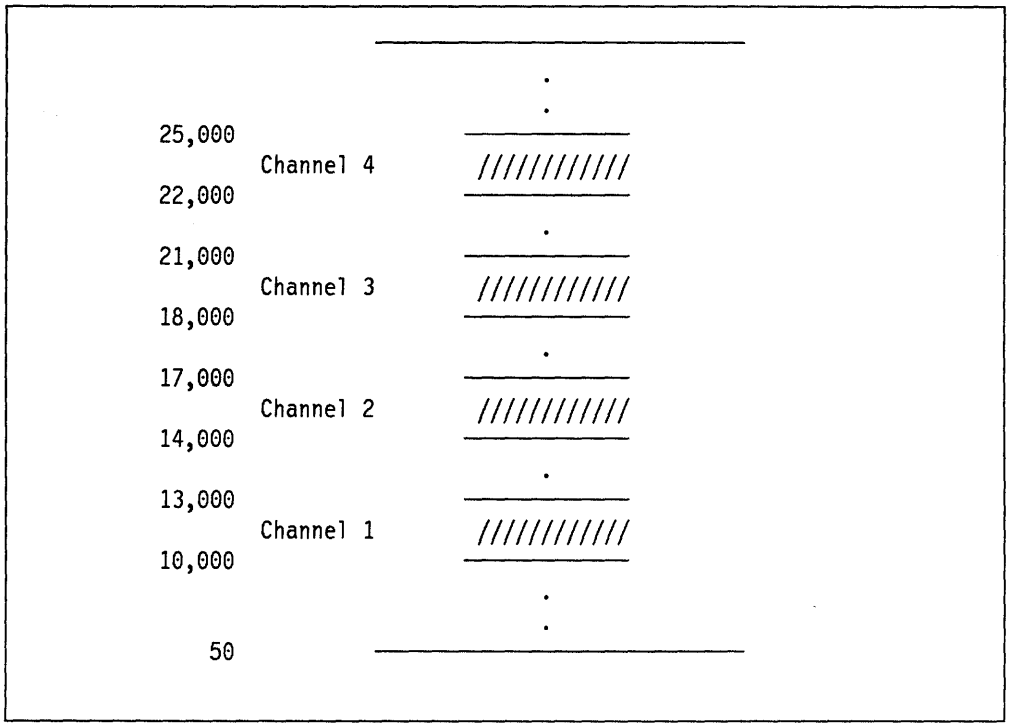


Figure 119. The Concept of Frequency Division Multiplexing. The physical carrier provides a range of frequencies called a "spectrum" within which many channels are able to co-exist. Notice the necessary "buffer zones" between frequency bands.

A.1.2 Time Division Multiplexing

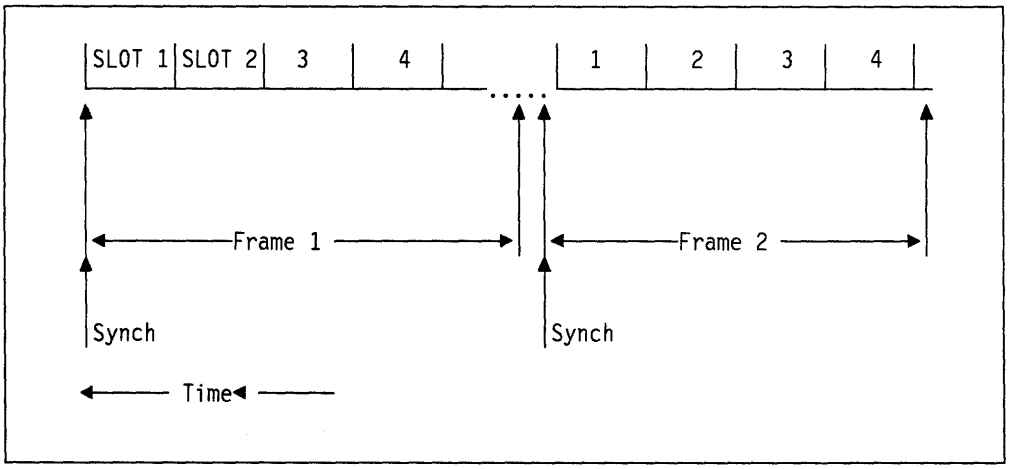


Figure 120. Time Division Multiplexing Principles

Consider the diagram above. "Frames" are transmitted over a single high speed channel. Within each frame there are many slots. A low speed channel is allocated one (or more) slots within a high speed frame. Thus a 2,048,000 bps channel can be subdivided into 32 subchannels of 64,000 bps. The start of each frame is signaled by some coding which allows the sender and the receiver to agree on where the beginning of the frame is. This synchronisation coding is sometimes a special (unique) bit stream (as when SDLC or BSC traditional data transmission is used) but with digital transmission is usually signaled by some

special state in the underlying PCM coding. (The common one is called a "code violation".)

Attaching equipment is able to insert data into any slot and to take data from any slot. Thus while the medium can run at a very high speed, each attachment operates at a much lower data rate.

A.1.3 Packetisation

This technique involves the breaking of incoming bit streams (voice or data) into short "packets". Different techniques use variable or fixed length packets. Packets have an identifier appended to the front of them which identifies the circuit or channel to which they belong.[89] In the TDM example above, a time slot was allocated for a low speed channel within every frame even if there was no data to be sent. In the packet technique blocks are sent only when a full block is available and "empty" packets are not sent. Thus utilisation of the high speed link can be dramatically improved.

Thus if a voice channel (without compression) is regarded as 64KB of data and a TDM approach gives 32 channels, then using packetisation the number of channels that can be handled will dramatically increase. If, on an average each (one-way) channel is only operational for half of the time (as in voice conversation) then perhaps 64 voice channels could be available. However, now there will be statistical variations and there will be a finite probability that all 64 channels will want the same direction at once. In this case some data will be lost but the probability is very small. The probability of 33 channels wanting to operate in the same direction simultaneously is quite high however. It is a matter for statisticians to decide how many channels can be safely allocated without too much chance of losing information (overrun). This will depend on the width of the carrier and the number of channels. The larger the number of channels the smaller the variation and the greater the safe utilisation. This is a similar situation to the queueing models of data communication but the characteristics are quite different. A good starting assumption is that the channel can be utilised to perhaps 70% of its capacity safely, (in a 2 Mbps circuit). In a 140 Mbps circuit (PCM fibre channel) then the 2,000 telephone calls will have a much more even distribution and, therefore, efficiency could perhaps approach 90%. That is, perhaps 3,500 calls could be handled. (Other things, like routing headers and the requirement for uniform transit times will probably reduce this somewhat.)

[89] Alternatively, they could have a routing header which identifies the source and destination of the packet.

A.1.4 Sub-Multiplexing

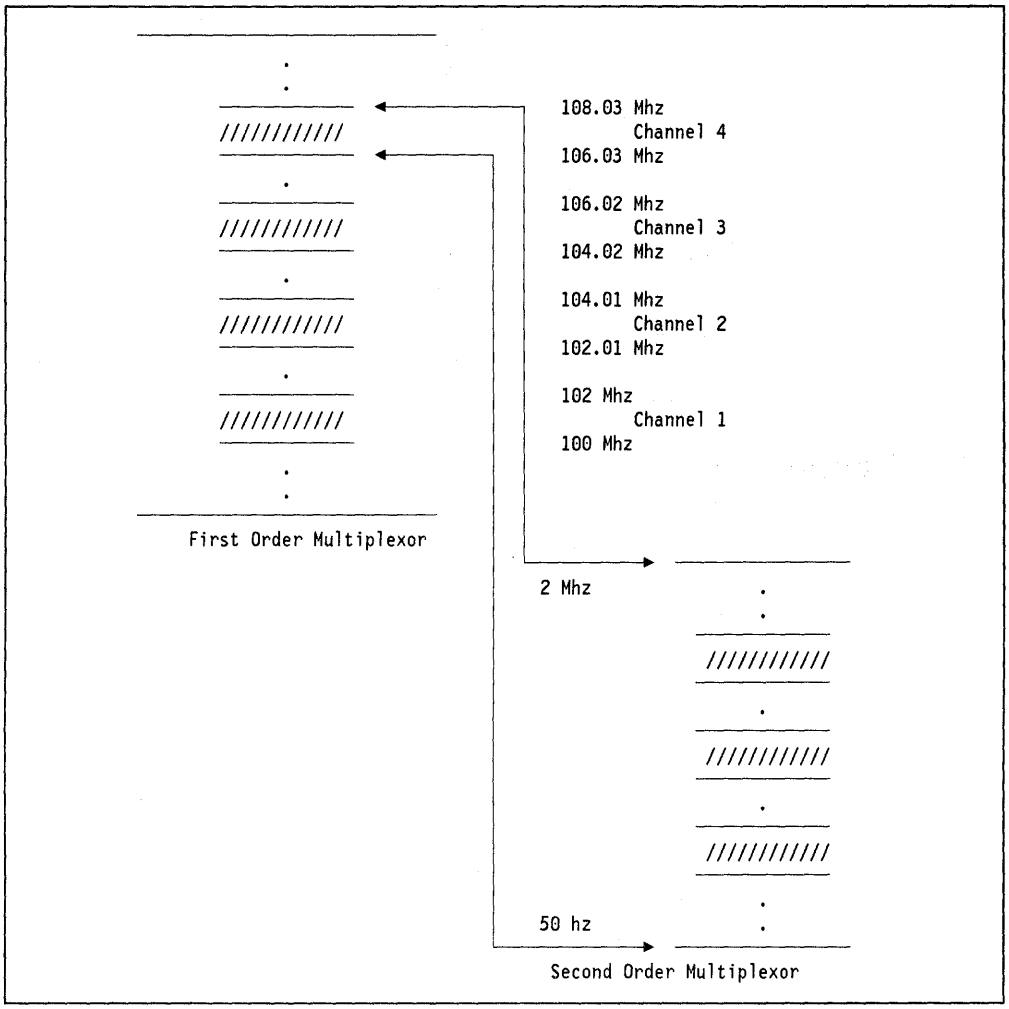


Figure 121. Sub-Multiplexing Concept. Frequency division technique used within frequency division derived channel.

It is quite possible, indeed usual, for multiplexors to be "cascaded" as suggested in Figure 121. A "high order" multiplexor is used to derive a number of lower speed channels that then are further reduced by other (lower order) multiplexors. This may then be reduced even further by lower and lower order multiplexors.

Since a derived channel is just like the original channel only "narrower", then different multiplexing techniques can be used within one another. For example, it is possible for a wide-band microwave channel to be frequency divided into a number of slower channels and then for one lower speed channel to be further divided by the frequency technique, another to be shared using the TDM digital technique and yet another to be shared using the packet technique. There are limitations, however. For example, a digital channel cannot be shared using frequency division multiplexing and a TDM technique would not be a very attractive way of subdividing a packet technique. Still, mixtures of techniques can be used relatively freely.

The hierarchies of multiplexors used in different parts of the world is shown below:

Table 4. PCM Hierarchies						
	North American		Japanese		CCITT-CEPT	
Order	Bit Rate (Mbps)	No. of Chan	Bit Rate (Mbps)	No. of Chan	Bit Rate (Mbps)	No. of Chan
Single Channel	64 Kbps	1	64 Kbps	1	64 Kbps	1
First Order	1.544	24	1.544	24	2.048	30
Second Order	6.312	96	7.876	120	8.448	120
Third Order	44.736	672	32.064	480	34.368	480
Fourth Order	274.176	4032	97.728	1440	139.268	1920
Fifth Order					564.992	7680

A.1.5 Statistical Multiplexing

Statistical multiplexing is the generic name for any method that aims to use channel capacity (and send information) only when there is information to send. This is in contrast to the techniques of allocating a channel and then not caring whether that channel is used or not. Packetisation is one form of statistical multiplexing.

The technique offers savings for voice in that the gaps in speech and the "half duplex" characteristic of speech can potentially be exploited for other conversations. Likewise, the gaps in traditional data traffic can be exploited. There are multiplexors available that allocate a large number of voice channels over a smaller number or real channels by this technique. Listening to any one of the voice channels would provide the listener with intermixed phrases and sentences from different conversations on the one real voice channel. In data communications, the use of "statmuxes" that derive (for example) 6 or 8 slow (2,400 bps) channels from a "standard" 9,600 bps line are in common use.

All of these have problems in that they require some technique to recognise "silence". That is, to determine what not to send. In voice, a delay buffer is needed so that when a word is spoken following a silence the need for a channel is recognised and the channel made available WITHOUT chopping off the beginning of the word.

In the past this technique has been used to improve utilisation of expensive undersea telephone cables but is not in common use for other situations because of the cost and the impact on quality caused by the additional delays and the interposition of yet another piece of equipment which degrades the signal quality.

A.1.6 "Block" Multiplexing

Block multiplexing is the usual method of operation of data networks. It is one form of statistical multiplexing.

A single channel is used to transmit "blocks" of varying lengths depending on the logical characteristics of the data to be transmitted and the physical characteristics of the devices involved. Often maximum limits are imposed on block lengths (though sometimes not). Blocks are usually queued for the link according to various criteria such as priority, length or message type. In IBM's

"Systems Network Architecture" (SNA), this method is used on all links but different characteristics apply to different kinds of link. For example, on links between IBM 3725s block lengths can be very long (4,000 bytes or more) and on links between 3725s and controllers, blocks are "segmented" (broken up) to fit into the I/O buffers of the receiving device.

Some types of local area network also use this technique.

A.2 Characteristics of Multiplexing Techniques

Frequency Division Multiplexing

- This is an analogue technique and applies to the kind of "interexchange carrier" systems still in use by many telephone companies (although it is rapidly being replaced by digital TDM techniques).
- Subchannel are separated from each other by "buffer zones&c dq. which are really wasted frequency space or wasted capacity, albeit necessary.
- The analogue equipment needed to make this work is extremely sensitive to "tuning" of frequencies and to the stability of filters.
- This equipment is also very expensive because of its analogue nature and its sensitivity to tuning, etc., and requires a large amount of labour to install and maintain.[90] It also requires retuning and maintenance whenever the physical channel changes (for example, is rerouted or repaired, etc.).
- Also, it is usual to use two channels per conversation, one in either direction. A reasonable estimate of "good" channel use by this technique (for voice traffic) is 10%.
- However, the equipment is extremely modular and a failure in one element most often does not affect the operation of the remainder of the system.

Time Division Multiplexing

- This method is quite simple and can be built in single chip hardware logic.
- Therefore, the hardware is low in cost (compared to other techniques).
- It will operate at very high speeds.
- While it gives sharing and channelisation of the link, it does not take into account, for example, the fact that telephone traffic is logically half-duplex (only one person talks at once) and though a channel is provided in each direction only one is in use at any one time. Nor does it take advantage of "gaps" in speech. There are intelligent multiplexing techniques (called statistical multiplexors) which do this. For these reasons "good" utilisation for telephone traffic is

[90] Another factor contributing to the expense is that it is difficult to apply large scale integration techniques to analogue systems. Analogue equipment tends to have many more separate components than comparable digital systems (digital ones have more circuits but many are packed together into a single component). This leads to a higher cost for the analogue alternative.

considered to be around 40%. This is a lot better than the analogue frequency division technique.

Packetisation

- The equipment required for packetisation is MUCH more complex and expensive.
- Operation at very high speeds increases the complexity of the required equipment.
- Use of the packetisation technique results in very much improved (optimal) use of the trunk. This is because when there is a silence no packet is transmitted. There are overheads inherent in the addressing technique which must be used to route and to identify the packet but nevertheless, an efficiency of 80% can perhaps be approached if the carrier is wide enough to permit a large number of simultaneous calls.
- In 1992 this technique is NOT operational in any serious voice system. However, this technique is the basis of the "ATM" (Asynchronous Transfer Mode) of operation and it is widely believed that this technique will ultimately replace TDM techniques as the basis of worldwide voice and data public communication networks.

  Packetisation is a normal method for operation of data networks and is in wide use, (there is an important cost/performance trade off discussed under "Block Multiplexing").

Block Multiplexing

This can be considered a special case of packetisation (in the sense of variable length packets). Alternatively, packetisation can be considered a special case of block multiplexing with fixed length blocks. In the data processing industry, the use of variable length blocks is usual. In the data networks constructed by telephone companies (generically called X.25 networks because of the "standard" they use to interface to their users), packetisation is the normal method.

In a network where data integrity is of supreme importance, a block must be fully received by an intermediate node BEFORE it is sent on to the next node.[91] Since the whole block must be received before it is sent on, the longer the block, the longer the delay in the intermediate node. If there are many nodes through which the data must pass, then the delay for each node adds to the total delay for the network transit. **So the shorter the block, the more quickly it can be sent through a network.** Thus if there are 2,000 bytes to be sent, the transit time will be much faster if this is sent as ten, 200-byte "packets".

There is another problem with widely varying block lengths. Queueing delays for links become uneven and parallel link groups tend to operate erratically. (In SNA for example, "Transmission Group" (TG) operation could be greatly degraded by the presence of widely varying block lengths.) Another problem is that varying block lengths create erratic demands on buffer pool utilisation in switching nodes.

[91] Whether the routing header in the beginning of the block is correct is not determined until the end of the block has been received and the Frame Check Sequence (FCS) is checked. So the block cannot be sent on until it is checked.

It would seem sensible, therefore, to keep data block lengths to a minimum in all networks. However, there is a very important characteristic of computer equipment that must be taken into account. **Processing a block (switching or sending or receiving) takes almost exactly the same amount of computer processing regardless of the length of the block.** This applies to computer mainframes, switching nodes, terminal controllers etc.[92] So, in the example of a 2,000-byte block, the processing time in each involved processor (including the sender and receiver) **will be multiplied by 10** if blocks are sent 200 bytes at a time.

Since the amount of processing involved is a cost factor, it will cost a lot more for processing hardware if short blocks or packets are sent. Hence the EDP industry tradition of sending variable, long blocks.

When considering voice transit, where network transit delay and uniformity of response are paramount, it would seem that mixing voice into such a data system would not be feasible.

However, the above applies to traditional systems where the block is "long" in time as well as in bytes. That is, the link speed is low compared to the length of the block. With high link speeds (for example, 4 Mbps on a local area network), block lengths that were considered "long" in the past become short enough not to be a problem (for example, 5,000 bytes takes 10 milliseconds at 4 Mbps).

Also, new equipment designs will, in the future, allow vastly reduced cost in the switching node so the cost of short packets may be less of a problem.

Sub-Multiplexing

This technique is popular because it allows for great flexibility and for modularity of equipment. For example, a first order TDM multiplexor might break up a 140 Mbps fibre circuit into 64, 2 Mbps channels. Some of these can then be further multiplexed into 30 (64 Kbps) voice channels or perhaps several 2 Mbps channels combined to provide a television circuit. The derived 64 Kbps voice channels can perhaps then be broken up further by (third order) multiplexors to provide many 4,800 or 9,600 bps data channels. The equipment modularity introduced also assists in reliability and serviceability management.

The ability to mix techniques is also very important. (There are some limits here; frequency division multiplexing cannot be used within a digital channel, for example.) A wideband analogue carrier can be frequency multiplexed into several high speed digital channels and then one of these could use packet mode techniques for carrying data and others could use TDM techniques for carrying voice traffic.

---

[92] In the IBM 3725, for example, switching an intermediate block takes (total including link control) around 3,200 instructions. Getting data into or out of the processor takes of the order of 100 nanoseconds per byte. So for INN (Intermediate Network Node - meaning between two 3725s) operation a 3725 can "switch" somewhere around 350-400, 200-byte blocks per second (70% utilisation) and maybe 360 to 410 blocks per second if blocks are 100 bytes. (These figures are approximate and depend heavily on environmental conditions.)

Appendix B. Queueing Theory

An understanding of the principles of queueing theory is basic to the understanding of computer networks and so a brief discussion is included here for reference.

When the earliest interactive (or "online") computer systems were built, the designers often did not understand that queueing theory would dictate the behavior of these new systems. Some serious and very costly mistakes were made because the way communications systems actually behave is **the opposite of what normal intuition would suggest**. That said, queueing theory can be applied to any situation where many users wish to obtain service from a finite resource in a serial fashion. Such situations as people queueing for the checkout at a supermarket or motor cars queueing at an intersection are much the same as data messages queueing for the use of a link within a data switching device.

Consider the diagram below:

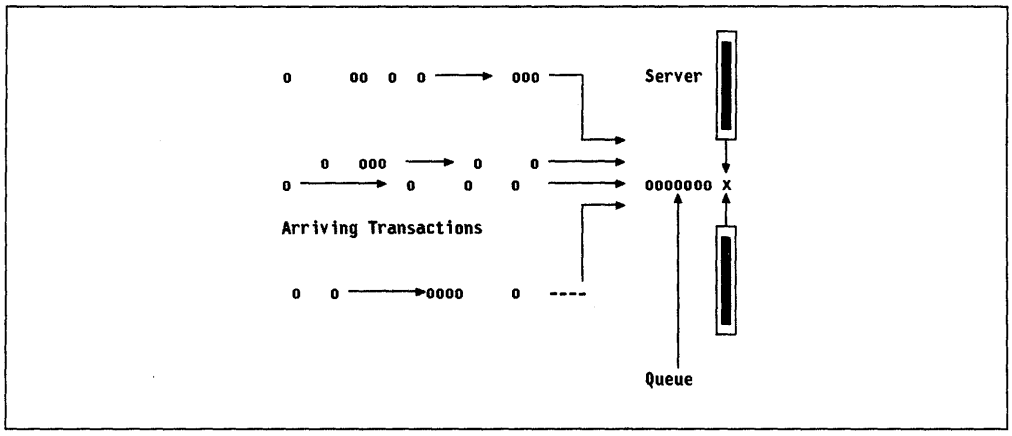


Figure 122. Schematic of a Single Server Queue

Imagine that this is a queue of people at a supermarket checkout. Most reasonable people would expect that if the average time it takes for the checkout clerk to serve one customer was two minutes then the rate at which customers will be served would be 30 customers per hour. At this point the checkout clerk would be busy all the time (100% utilised). It also seems reasonable to expect that if the number of people who arrive to join the queue is about 30 per hour then the length of the queue should be one or two and the time spent waiting for service would be quite short. Nothing could be further from the truth! In the case just described (over time) the queue will become very long indeed. In theory the length of the queue in this situation will approach infinity!

In order to discuss the way queues behave there are some technical terms that must be understood.

Service Time is the time taken by the checkout clerk to process a particular person. This will be different for each person depending on the number of items to be processed.

Average Service Time is the average over time of a number of people processed by the checkout.

Arrival Rate is the rate (people per hour) at which people arrive and join the queue.

Queue Length is the number of people waiting in the queue at a particular time.

Average Queue Length is the average length of the queue over a given period of time.

Server is the job of the checkout clerk.

Utilisation of the Server is the percentage of time that the server is busy.

This is the average service time multiplied by the arrival rate divided by the length of time in question. The utilisation value is something between zero and one but is often expressed as a percentage.

The graph below in Figure 123 shows queue length as a function of server utilisation for single server queues.

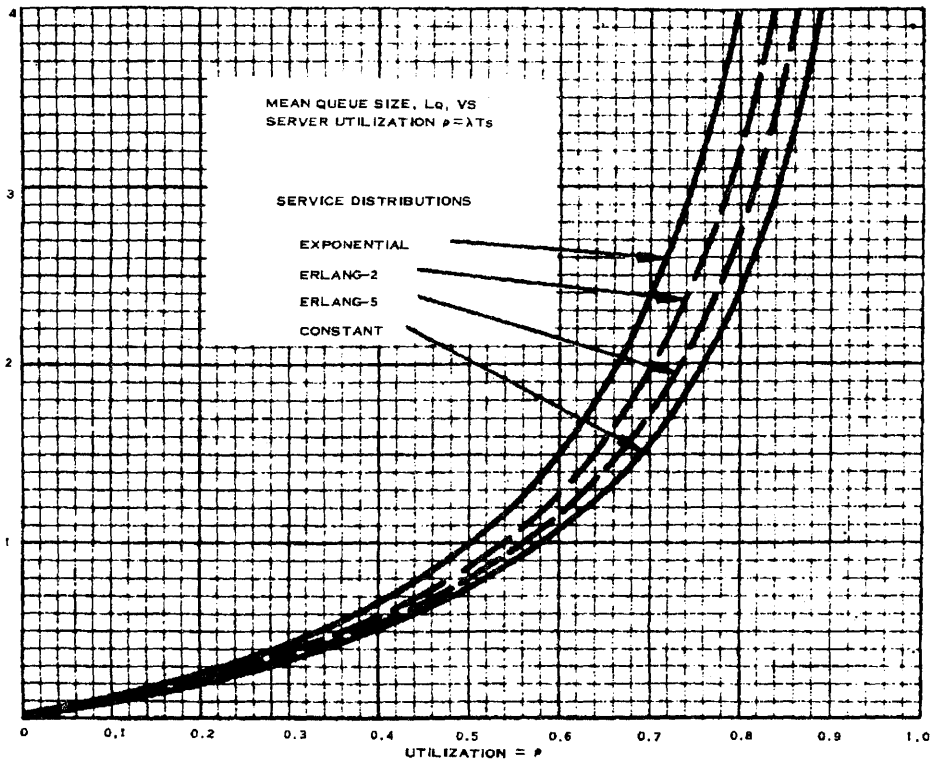


Figure 123. Behavior of a Hypothetical Single Server Queue. The curves show queue length as a function of server utilisation. As utilisation approaches 1 the length of the queue and hence the waiting time approaches infinity.

B.1.1 Fundamentals

In order to describe the behavior of a queue mathematically we need the following information:

Input Traffic

- Arrival Rate

  The average (over some specified period of time) rate at which transactions (messages, people etc.) arrive at a facility requiring service.

This is represented by the parameter λ.

- Distribution of Arrivals

  Quoting an average arrival rate or an average service time tells us nothing unless we know something about the overall pattern which these things follow. In some systems, transactions may arrive at exactly regular intervals. In other systems arrivals may occur in bursts. In most interactive data communications systems transactions arrive for processing in a completely random way.

  The usual distribution of arrivals used in a data communication system is called a Poisson distribution. This distribution describes random arrivals from an infinite population independent of past events.

The Waiting Queue

- The Queueing Time

  The total time spent by a transaction in the queue (including the service time) is called the queueing time.

  T_Q = mean (average) queueing time

- The Waiting Time

  The time spent waiting in the queue *not* including the service time is called the waiting time.

  T_W = mean (average) waiting time

- Number of Transactions in the Queue

  The length of the queue expressed in terms of the number of waiting transactions is called L_W.

The Service Facility

- Mean Service Time

  This is the average time it takes to process a transaction and is denoted by T_S.

- Server Utilisation

  This is the fraction of time the server is busy. It is symbolised by ρ. Busy time is (obviously) just the arrival rate multiplied by the average service time.

  $\rho = \lambda \times T_S$.

As λ, the rate of arrivals, increases then so does the utilisation of the server, the queue length and the average time spent waiting in the queue. At $\rho = 1$, the server becomes saturated, working 100% of the time. Therefore the maximum input rate to a single server queue is:

λ maximum $= 1/T_S = \mu$(mean service rate)

The utilisation coefficient is:

$\rho = \lambda/\mu$

Many of the numbers above are averages (means). If the arrivals are random then there will be a deviation around the mean values. For example, the 90th percentile is that value of time below which the random variable under study

occurs 90% of the time. Specific values for the 90th and 95th percentiles for Poisson distributed arrivals are:

90th percentile = 2.3 × (mean value)

95th percentile = 3 × (mean value)

The golden rule

For most queueing situations where arrivals approximate a random distribution queue size, the average delay increases exponentially with the utilisation rate. This is illustrated in Figure 123 on page 284.

B.1.2 Distributions

There are two things under discussion that are governed to some degree by uncertainty:

1. The pattern of arrivals
2. The time it takes to service a given transaction

Both of these distributions may be exactly regular or completely random or anything in between. Notice that the distribution of arrivals and the distribution of service times are quite independent of one another.[93]

The "in between" distributions are called "Erlang distributions with parameter m", or just "Erlang-m". The "Erlang parameter" is a measure of randomness. In Figure 123 on page 284 the different curves apply to different service time distributions. Arrivals are assumed to be random.

In typical data communications systems interactive data traffic arrives with a fully random (Poisson) distribution. For batch traffic, unless it is controlled, the traffic will arrive at the maximum rate of the link. Voice traffic typically originates at the rate of one byte every 125 µsec. This is one 32-byte packet every 4 milliseconds. Unless there is compression and/or removal of silences, the voice traffic will travel through the network at exactly one (32-byte) packet every 4 milliseconds.

In a typical data communications application service time is the time taken to send the transaction (frame) on a link and will vary in direct proportion to the length of the frame. If we are considering time taken to process a transaction (switch the block) in a packet switching computer typically this is the same amount of time regardless of the length of the block.

B.1.3 Some Formulae

The following are some simple formulae which can be used to analyse typical queueing situations. These are for exponential arrival and exponential service distributions.

[93] This is not quite true. The rate of arrival of frames at a switching node is dependent on their length and the link speed. The longer they are, the lower the rate of arrival. For the purpose of discussion this effect can be ignored.

B.1.3.1 Queue Size and Length

The mean queue size, including the transaction currently being serviced is:

$L_q = \rho/(1-\rho)$

The mean length of the queue is:

$L_w = \rho^2/(1-\rho)$

Then note that the queue length is shorter than the mean queue size by the quantity ρ; that is, the difference is, on the average, less than one transaction:

$L_q = L_w + \rho$

B.1.3.2 Queueing Time

The mean queueing time is:

$T_q = T_s/(1 - \rho)$

Queueing time 90th percentile = $2.3T_q$.

Queueing time 95th percentile = $3T_q$.

Mean waiting time:

$T_w = (\rho \times T_s)/(1 - \rho)$

B.1.4 Practical Systems

In a data communications processor (switch) there may be many queues. There is a queue (albeit a logically distributed one) for the processor resource. There are queues of data waiting for transmission on each outbound link. There may be many internal queues (such as for the I/O channel) depending on the structure of the processor.

The general rule is that no resource should be loaded more than 60% because of the effect of the waiting time on system responses. This is still a good rule if transactions arrive at truly random rates, are time sensitive and there are no congestion control mechanisms in place to control things. But there are a number of situations in which much higher utilisations are practical.

Regularity of Input Distribution and Service Time Distribution

If transactions arrive at a processing system (for example) at *exactly* one millisecond intervals *and* if processing takes *exactly* one millisecond then the facility will be utilised 100% and there will be no queue. All hinges on the word "exact".

If the distribution of arrivals and of service times is truly random then the "exponential" curve in Figure 123 on page 284 will hold true.

If arrivals are regular and service times are exponentially distributed then the "constant" curve in the figure is appropriate.

In summary, for a particular system, the less randomness involved the higher the safe level of utilisation.

Traffic Priorities

Traffic prioritisation is *very little use* unless there are congestion control mechanisms in place to allocate resources according to priority and to regulate low priority traffic.

In an SNA network for example, the extensive priority mechanisms can allow an internode link to be loaded in excess of 95% without significant degradation of the interactive traffic.

But the interactive traffic (the random traffic) itself had better not exceed about 60% utilisation of any facility. What happens is that interactive traffic uses the system as though the batch traffic was not there (well, almost). Control protocols allocate spare capacity to non-time-critical batch traffic.

The Law of Large Numbers

Consider a queue of data for a 9,600 bps intermediate node (INN) link in an SNA network. The sizes of frames carried on the link could vary from 29-byte control frames up to 4,000-byte blocks.[94]

The transmission time required to send a 4,000-byte block is a bit less than 4 seconds. For the 29-byte block it is about 35 milliseconds. Perhaps, for the sake of example, the average block length might be 400 bytes and take a transmission time of about .4 of a second.

If you select say 10 frames from the link at random then the average of the selected frames will be different from the true average over say one million frames. (There is a chance that all 10 frames will be 4,000 bytes long - or 29 bytes long). If you keep selecting samples of 10 frames each then the averages of these samples will be distributed about the true average with some variation (variance). If you select a sample of 100 frames then the chance is much better that it will be close to the true average. If you select a sample of 1000 frames then the chance is better still. The point here is **as the selected sample gets larger then the probability that its mean is close to the true mean of the population from which the sample is taken increases.**

Now consider a queue for a real link. If there are two (4000-byte) blocks waiting, then this represents around eight seconds of transmission time. If there are two (29-byte) blocks waiting then there is only about 65 milliseconds of transmission time. The example is deliberately extreme. It is obvious that transit delays through the link in question will be highly erratic. (A good case for breaking the data up into short packets perhaps). In this situation it would be dangerous to plan for a link utilisation of more than an average 30% because of the extreme variation that is possible in the service time (transmission time).

A faster link can handle more traffic. This means that there can be more blocks handled over any given time interval. This has the statistical effect that the selected sample of the population is larger and thus the likelihood of an atypical average is decreased.[95]

For example if the link speed is two megabits per second. A 4,000-byte block now takes 16 milliseconds and the 29-byte block takes about 120 microseconds. At 60% utilisation we can process about 40, 4,000 byte blocks per second or perhaps 5,000 29-byte control messages. If the average block length is 400 then the number of blocks per second for

---

94 SNA does not packetise or segment data for transport on INN links. The maximum length of blocks entering the network can be controlled however.

95 Another way of saying this is that if we sum n sources of bursty traffic then the total rate exhibits less burstiness as a result. As the number (n) of sources increases, the ratio of the standard deviation to the mean of a summed rate approaches zero.

60% utilisation is about 400. In this situation a group of 400 blocks will come very close to the average. The fact that some blocks are 4000 and others are 29 will have very little effect on the average. There will be very little variation.

Imagine what happens at 100 megabits per second!

This then is the effect: As the link speed increases the number of frames per second increases (unless we increase the block size). As the frame rate increases then the possible variation in a given sample of traffic (say one second) decreases. As the variance decreases one is able to use increasing link utilisations with safety. That is to say **the faster the link the higher the safe level of utilisation**.

High Speed in Itself

In practical networks of the past (with 9,600 bps links) a planned queueing delay for a link of say .4 of a second was considered acceptable. In a modern high speed network we are perhaps able to accept a queueing delay of 50 milliseconds. The link speed has increased by 1000 times but our delay objective has only become ten times more stringent.

This means that in the past, to achieve our .4 second delay objective, we could only plan on a queue length of perhaps 1 or 2, which means a link utilisation of about 60% . With a very high speed link we may be able to plan for a queue length of 60 to 100. Because of the stability associated with the very large number of frames per second, **a link utilisation of 80% may be approached in some cases**.

B.1.5 Practical Situations

In a practical networking situation there are number of points to note:

Time Interval

The time interval over which an average is expressed is absolutely critical. Traffic varies over time and expressions of arrival rates in "messages per hour" often do not help very much. On the other hand, extremely high peaks in message traffic over very short periods (say ten seconds) can be easily handled by buffering in the network.

Experience with conventional SNA networks suggests that an appropriate length of time over which to calculate peak message or transaction rates is five to ten minutes.

It is also important to realise that systems take a long time to reach a steady state after startup.

Arrival Patterns

Arrival patterns will not always be as the formulae assume. In any data communications system the population size of terminals is finite. In the case of interactive traffic there can only be a maximum of one transaction (perhaps many messages) per terminal in the system at any one time. Often a terminal device has a maximum rate at which transactions may be entered (for example, a bank cash dispenser). This could be because of mechanical functions that must be performed or because of the necessary actions by the human operator (like talking to a customer).

Service Time Distribution

This is almost always not random. In a communications node the time taken to transmit on a link is typically a linear function of the length of the data. The processor time taken by the node processor to receive the data on one link and enqueue it for another usually doesn't vary very much. Processing a transaction in a mainframe processor is usually reasonably constant *for each type of transaction* but things like disk access times tend to be random.

Multiple Servers

Instead of a single processor to handle transactions one could have a multiprocessor. Between nodes one can have parallel links. In this case we have a "multiserver queue". Statistics text books have many curves for the multiserver case but the principles are exactly the same as for a single server. In general, the curve shifts a bit to the right (meaning that slightly higher utilisations may be achieved). An important additional effect is that the queueing time becomes more uniform (less variance).

This is the situation such as in some banks where a single queue is used to supply multiple tellers. If one teller gets a customer requiring extensive time to service, the person in the queue immediately behind is serviced by the next free teller. If there were separate queues then this customer would have to wait much longer.

Appendix C. Getting the Language into Synch

When describing the timing relationship between two things there are a number of words that must be understood. Unfortunately, these words are commonly used in different contexts with quite different meanings.

In this document the strict engineering definitions have been used except where otherwise noted (such as in the description of FDDI. See section 9.3, "Fibre Distributed Data Interface (FDDI)" on page 183). When reading other documents it is necessary to try to understand exactly what the author means by the various terms.

In Communications Engineering when one bit stream is being compared with another the following terms are often used:

Synchronous

This literally means "locked together". When two bit streams are said to be synchronous it is meant that they are controlled by the same clock and are in the same phase.

Asynchronous

Any two events that are not tied together exactly in time are said to be asynchronous.

Isochronous

Literally "in the same time". An isochronous bit stream is one that goes at a constant rate. The term isochronous is often used colloquially to mean "digitally encoded voice". The term is not often used in the world of data communications but is a common term in the voice communications and engineering context.

Anisochronous

Literally, "not equal". Used in the communications engineering context to mean bit streams of unequal rates (this causes a problem in time division Multiplexing systems).

Mesochronous

The Greek prefix "meso" means "middle". If two isochronous bit streams have no precisely controlled relationship, but have exactly the same bit rate, they are called mesochronous.

If two bit streams start at some point of origin as synchronous streams and arrive at some destination by different routes or networking schemes they will be mesochronous at the point of arrival.

Plesiochronous

Literally "nearly the same". If two bit streams have nominally the same bit rate, but are controlled by different clocks, they will have bit rates that are nearly the same but not quite.

Heterochronous

If two bit streams have nominally different bit rates they are said to be heterochronous.

In data communications the common terms are "synchronous" and "asynchronous". Their meaning is usually determined by context.

The word "synchronous" when applied to a stream of bits *in itself* means that all the bits in the stream bear a strict timing relationship with one another. The same term can also be applied to a stream of characters.

The word "asynchronous" is usually used to mean **a data stream where the characters are not synchronised with one another but the bits within each character are**.

> This is the case of typical ASCII terminal operation. Each character consists of a start and a stop bit surrounding seven or eight data bits. Each data bit has a precise timing relationship with each other data bit within the character but characters are transmitted as they are typed and have no particular relationship with one another at all.

A synchronous link protocol such as SDLC (Synchronous Data Link Control) operates on blocks of data that are expected to be synchronous within themselves but there is no strict timing relationship between blocks of data.

Sometimes these concepts get mixed up as when SDLC data link protocol is used to transmit data over an asynchronous circuit or when ASCII protocols are used to send data through a synchronous channel.

In the FDDI standard the term synchronous is used in a completely different way: It means data traffic that has some real time or "guaranteed bandwidth" requirement.

All of the above terms describe a timing relationship between two things. When using these terms, it is essential that we know just what things are being related and in what aspect they are being compared. Care is essential.

Appendix D. An Introduction to X.25 Concepts

The CCITT recommendation X.25 describes an interface between a user and a packet-switched data network.[96] **It must be emphasized that the recommendation ONLY describes the interface between the user and the network. It describes the operation of the interface in great detail and it specifies what services should be available to a user device operating on such an interface. It DOES NOT say anything at all about how the network should operate internally.** Thus the phrase "X.25 network" cannot say anything meaningful about how the network operates, only that the network can support a certain type of connection between users.

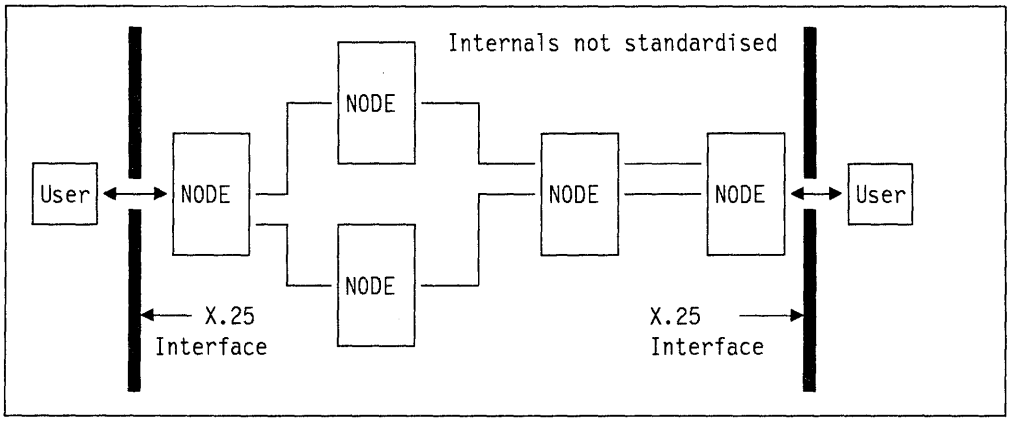


Figure 124. Schematic of a Packet Network. Each "node" (or "switch") receives a packet in full before routing it on towards it destination user. Not shown is the additional equipment necessary to manage the network.

Networks with X.25 capability are often represented as a "cloud". The "cloud" representation is useful since it emphasizes that the end users need have no concern about how the network operates internally.

The user presents data to the network in short blocks called packets. The function of the network is to deliver these packets to another user (destination) attached to the network. Packets are delivered, without change[97] to the data, in the same sequence as they were presented to the network and without being stored on any intermediate external storage medium (disk). Communication is synchronous in that both communicating users must be present at the same time for communication to take place.

The network is made up of nodes (also called "switches" or "Data Switching Exchanges" (DSEs)) that are connected together by data communication links (see Figure 124). In most networks each node is a specially designed computer but in addition there is almost always a larger network host (often a general purpose computer) to service the network nodes.

One of the parameters of network design in a packet network is the maximum length of a packet. Short packets make for fast transit times through the network. However they take considerably more resource both in the end user

[96] This appendix was abstracted from "Integrating X.25 Function into Systems Network Architecture Networks" (GG24-3052).

[97] It is possible, albeit inefficient, for the length of a packet to be different at different ends of the network, but the data is unchanged.

device and within the network. In X.25 the "universal compromise" packet size which every network must support is 128 bytes.

The central concept of X.25 is that of a "virtual circuit"; that is, a circuit is completed between two communicating end users in the same way as a circuit connects two people who use a telephone. The circuit is called "virtual" because it does not use dedicated resource within the network, but the logical path between a pair of end users is nevertheless dedicated to communication between this pair of users and no others. Virtual circuits (VCs) can be "switched" or "permanent". A switched virtual circuit (SVC) is sometimes named a "Virtual Call" (VC) in analogy with the telephone system and to confuse the innocent. The abbreviation for "permanent virtual circuit" is "PVC."

D.1.1 Components of the X.25 Interface

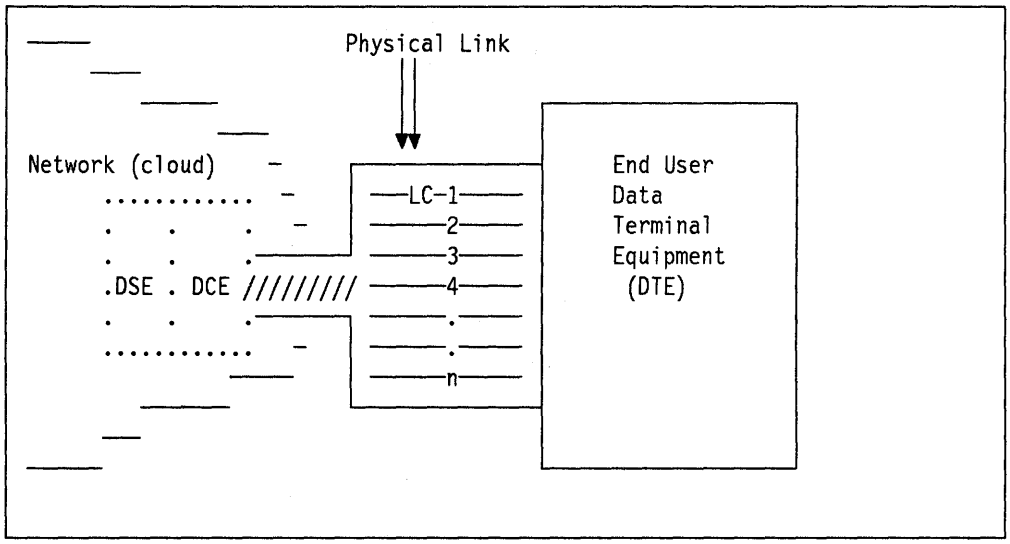


Figure 125. Elements of the X.25 Interface

The central concepts of the X.25 interface are illustrated in Figure 125. The more important of the concepts can be described as follows:

Packet

The packet is the unit of data sent between the end user (called the DTE) and the network. Its maximum size is called the "packet size" for this interface. A packet that is not full is sent as a short packet. **It is never "padded out" to the full packet length.**

Packet Header

Packets are transmitted with a three (or optionally four) byte header which is not included in the packet length. The header contains information about the packet type, a "more data" bit which allows for the grouping of packets into logical records and a 12-bit field which identifies which communication (virtual circuit) this packet belongs to.

Logical Channel

Within the one physical link to the network the end user (DTE) may have many communications (virtual circuits) with other end users in the network. Since the packet header does not contain the network address of the destination, there needs to be some mechanism of identifying

where this packet is to be sent. That method is the logical channel number (and the logical channel group number). A logical channel is simply a reference number to identify which virtual circuit this packet belongs to. Another way of saying it is that many logical communications can take place over a single physical circuit by multiplexing the physical circuit into many logical channels. The Logical Channel Number (LCN) is the identifier which is used to distinguish which virtual circuit a particular packet belongs to.

Virtual Circuit

A virtual circuit is simply an association between two logical channels, one at each communication endpoint. See Figure 126. A packet that is sent with a logical channel number of 3 by the user at interface A will be received at interface B with a logical channel number of 6 in its packet header. The detail of how the communication is achieved is left to the designers of the packet network.

DTE

Data Terminal Equipment. This means anything that is a user of the X.25 interface. It could be a simple terminal, or a protocol conversion device or a large mainframe CPU. The interface is the same and it is treated in the same way.

DCE

Data Circuit terminating Equipment. This is the network end of the link from the user. In various contexts it can be the modem interface or the interface at the network node. In the X.25 context it normally refers to the network node.

DSE

Data Switching Exchange. This term is not often used since the process of data switching is hidden from users of the network. It refers to the logical switching process within a node.

Physical Link

This is the link between the user and the network. In IBM "jargon" it is called the MCH (MultiChannel Link).

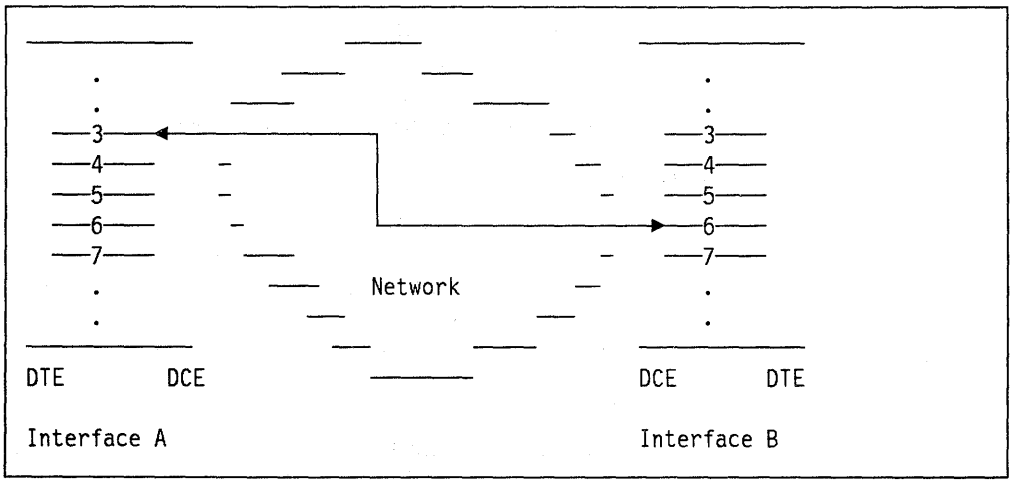


Figure 126. Virtual Circuit. Two logical channels (number 3 on interface A and number 6 on interface B) are communicating with one another. This pairing of logical channels is called a virtual circuit.

D.1.2 Logical Structure of the X.25 Interface

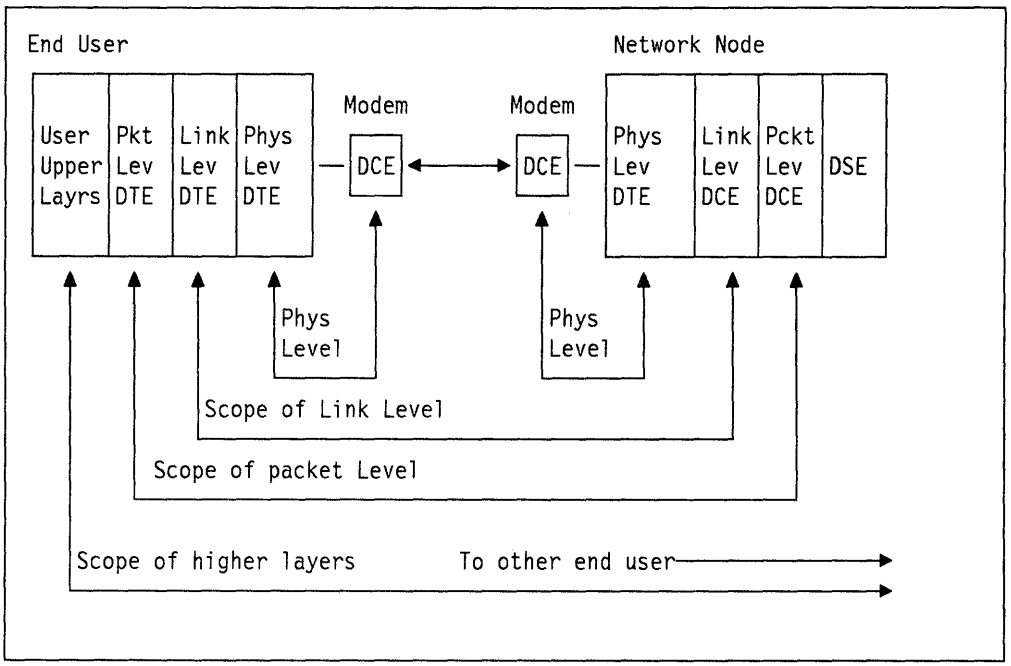


Figure 127. DTE and DCE Relationships. The scope of each layer is shown here. The relationship between DTE and DCE at each layer should be noted.

Logically, the interface operates in three "layers" called physical, link, and packet levels.

Physical Level

This is the lowest logical layer and provides electrical connection between the DTE and the link to the network. If a modem is used then the physical layer is specified by the CCITT recommendations V.24 or V.35.[98] If a "digital" link is to be used then recommendation X.21 is appropriate. This layer only provides for the sending and receiving of bits and for physical compatibility of plugs and cables and for voltage tolerances and signal timings etc.

Link Level

This is the "line control" or "link protocol" which passes frames of data to and from the network. Link control takes a queue of frames at each of the DTE and the DCE and is responsible for transferring them in as efficient a way as possible from one to the other. Link control takes frames from the packet level and delivers them to the packet level at the other side. Link level does not care about the content of the frames and does not know that the link is multiplexed among many virtual circuits. Link control is responsible for error detection and recovery (retransmission) between DTE and DCE. Inherent in the means of operation of the link control there is a "data flow control" enforced through a rotating "send window". However, it is not generally used for controlling flow except in conditions of extreme congestion because the DTE-DCE flow control is done at the packet level.

[98] V.24 and V.35 are also sometimes called "X.21bis". There are minor differences in technical detail between these V-series recommendations and X.21bis but the terms are often used interchangeably. "A rose by any other name..."

The link control protocol used in X.25 is a version (subset of options) of the international standard data link control "HDLC". The subset is the "Asynchronous Balanced Mode" and is called LAPB in the jargon of X.25.

Packet Level

The packet level protocol performs basically two functions. The first is to multiplex the physical channel provided by the link control into a number (up to 4096) of logical channels. The second is to provide a flow control between the DTE and DCE to provide an even delivery of data per logical connection (virtual circuit).

There is no error recovery in the packet level. (Though some network suppliers add one.) The scope of packet level is the same as for link level (from the nearest node in the network to the user) so error recovery is solely done at the link level.

D.1.3 Setting Up a Virtual Circuit

There are two kinds of virtual circuit:

Permanent virtual circuits are set up by the network administration and consist of a permanent relationship between a particular logical channel on one particular interface (or port) and another logical channel on a different interface somewhere else in the network. A permanent virtual circuit is always there (provided the network is operating) and the end user DTE requires nothing special to start using it.

Switched virtual circuits must be requested by the user and are then set up by the network (provided that the network has resources available). A special packet type (the "call" packet) is sent to the network on a logical channel that is not already in use. (There are strict rules for selection of the next logical channel to be used.) The call packet contains the address of the other user to which connection is requested. There is a standard format for addresses to be used by the network. The network then finds and sets up a path to the other user and selects a logical channel on which to notify the other user of the "Incoming Call". This incoming call (in reality the call packet sent by the other user with a different logical channel number and some fields changed) is presented to the other user which can then accept or reject the call. A "Call Accepted" packet is sent to accept the call or a "clear" packet is sent to refuse the call. "Clear" is the usual way of terminating an SVC connection (that is, hanging up).

D.1.4 Packet Types

There are surprisingly few different packet types in X.25.

Data Packets

These are the units in which data is sent through the network. Packets can be of different (maximum) sizes and in fact different logical channels on the same physical link can use different packet sizes. In the X.25 recommendation, packet sizes of 32, 64, 128, 256, 512, 1024, 2048 and 4096 bytes (octets) are allowed. However, every X.25 network must support a packet size of 128 in addition to the other sizes it may optionally support. For example, the network is capable of changing the packet size during transit so that a DTE at one end may have a packet size of 128 bytes and at the other end a packet size of 256.

Qualified Data Packets

These are the same as data packets but have the "Q" bit in the packet header set on. This is simply a way of identifying a logically different kind of data from what is carried in a data packet and can be used by the DTEs in any way they like.

Interrupt Packets

These packets carry only a small amount of data (in the CCITT 1980 version 1 byte) and are given priority in transit through the network. Whereas Data and Q packets are always delivered to the destination DTE in the order that they were presented to the network, interrupt packets are presented as soon as possible. It is up to the user to decide the meaning of data (if any) in an interrupt packet.

Control Packets

These are packets like "Call", "Reset" and "Clear" that are used to communicate information between the DTEs and the network.

D.1.5 The PAD Function

While the recommendation X.25 deals with the attachment of synchronous link-connected devices to a packet network, there are three other CCITT recommendations that are usually understood to be present when the phrase "X.25 network" is used. These relate to the attachment of "dumb" "ASCII TWX" devices to the packet network. Three recommendations are involved. These are X.3, X.28 and X.29. They are often referred to as "the triple-x pad" or just the "PAD."

These asynchronous terminals typically send a character on the communications line every time a key on the keyboard is depressed. That is, transmission is one character at a time and the assembly of characters into logical records and blocks is done by the program to which the terminal is communicating. These types of terminal often do not use error checking at all but it is increasingly common to use a method of error control called "echo-plexing". In "echo-plexing", a character sent from the device to the computer is sent back in the opposite direction and the terminal compares the returned character with the character sent in. If they are the same then there has been no error.

For efficiency reasons it is obvious that sending characters across the packet network as one character per packet will not be very attractive. There is then a need for a function that assembles characters from ASCII devices into groups to be sent through the network in packets. "PAD" stands for "Packet Assembler/Disassembler" and the PAD device performs this function.

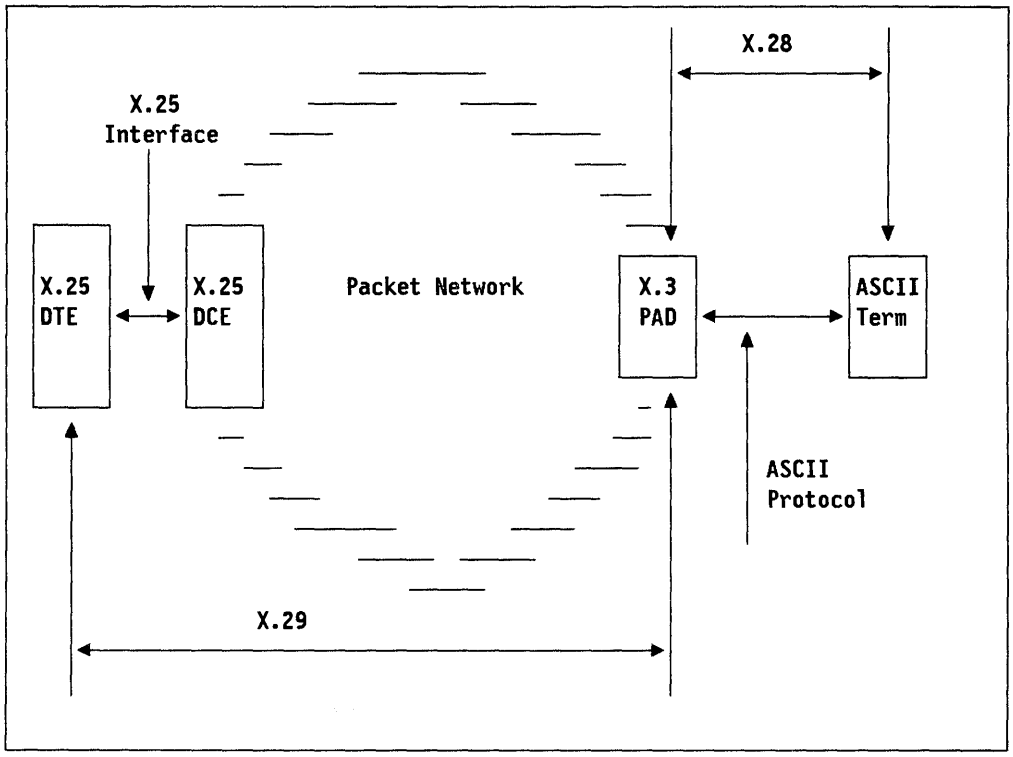


Figure 128. The ASCII "PAD". Scope of the three CCITT recommendations X.3, X.28 and X.29 is shown.

As characters are sent by the terminal, they are "echoed" (if needed) and assembled into a buffer. When certain criteria (such as a packet being filled or a predetermined control character being entered) are met, then the PAD will forward the new packet on a virtual circuit to the partner on the other side of the network. The three recommendations cover the following functions:

X.3 This covers the internal operation of the PAD, the control parameters that can be entered from the terminal to customize the operation of the PAD and things like echo and flow control etc.

X.28 This covers the data link interface between the terminal and the PAD. Access can be via the "PSTN" (Public Switched Telephone Network), or via leased line, or through TELEX etc.

X.29 This covers the exchange of control messages between the P-DTE (Packet Mode DTE) and the PAD. These messages are used for example to set up PAD parameters in order to either relieve the terminal operator from the chore of setting up PAD parameters or to prevent the terminal user from changing parameters without the permission of the host application.

In Figure 128 the PAD is shown as a function or a part of the network. This is not necessarily true. PADs are most often implemented as external devices as shown in Figure 129 on page 300.

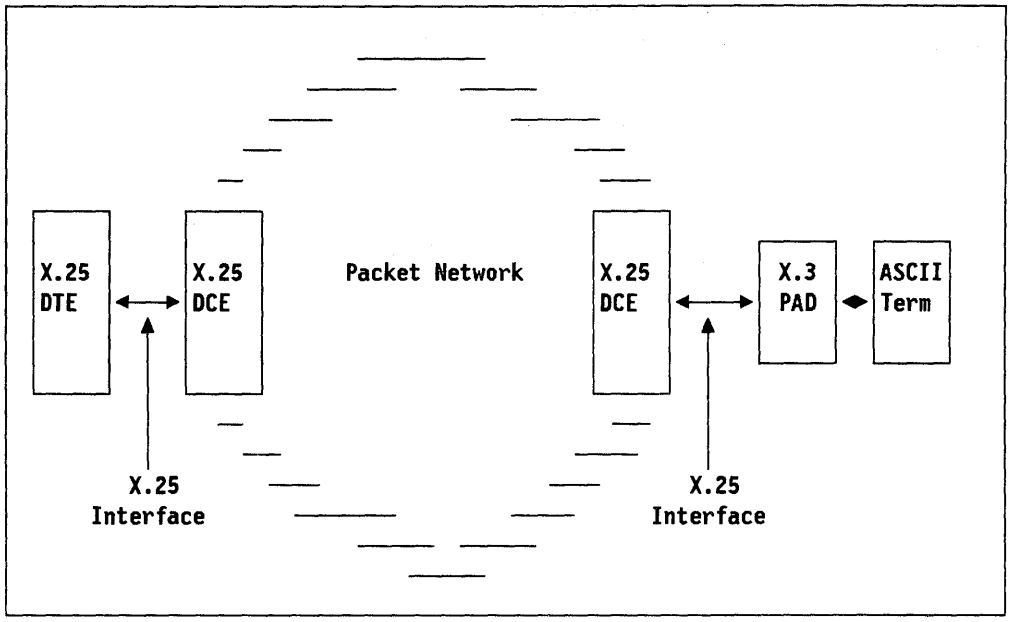


Figure 129. An "External" PAD. While described as a network function the PAD is most often implemented as an external piece of equipment.

Appendix E. Abbreviations

ABBREVIATION	MEANING
AAL	ATM Adaptation Layer
ACF	Access Control Field (DQDB)
AL	Access Link
ANSI	American National Standards Institute
APD	Avalanche Photodiode
ASCII	American (National) Standard Code for Information Interchange
ATM	Asynchronous Transfer Mode
BER	Bit Error Rate
bps	bits per second
BRI	Basic Rate Interface
B-ISDN	Broadband ISDN
CAU	Controlled Access Unit
CBR	Constant Bit Rate
CCITT	Comite Consultatif International Telegraphique et Telephonique (International Telegraph and Telephone Consultative Committee)
CMIP	Common Management Information Protocol (ISO)
CMIS	Common management Information Service (ISO)
CMT	Configuration Management
CPE	Customer Premises Equipment
CPN	Customer Premises Network
CRC	Cyclic Redundancy Check
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
dB	decibel
DFT	Distributed Function Terminal
DLC	Data Link Control
DPG	Dedicated Packet Group (FDDI)
DQDB	Distributed Queue Dual Bus
DSAP	Destination Service Access Point
DTE	Data Terminal Equipment
EBCDIC	Extended Binary Coded Decimal Interchange Code
FCS	Frame Check Sequence
FDDI	Fiber Distributed Data Interface
GFC	Generic Flow Control
HDLC	High-Level data Link Control
HDTV	High Definition Television
IEEE	Institute of Electrical and Electronic Engineers
ILD	Injection Laser Diode
IMPDU	Initial MAC Protocol Data Unit (DQDB)
INN	Intermediate Network Node
IO	Input/Output
IP	Internet Protocol
ISO	International Standards Organization
IWS	Intelligent Workstation
Kbps	Kilobits per second (Thousands of bits per second)
LAB	Latency Adjustment Buffer

LAM	Lobe Access Unit
LAN	Local Area Network
LAPB	Link Access Procedure Balanced (X.25)
LAPD	Link Access Procedure for the D_Channel (ISDN)
LE	Local Exchange
LED	Light Emitting Diode
LLC	Logical Link Control
LMI	Local Management Interface (Frame Relay)
LPDU	Logical Link Control Protocol Data Unit
LSAP	Logical Link Control Service Access Point
LT	Line Termination
LU	Logical Unit (SNA)
MAC	Medium Access Control
MAN	Metropolitan Area Network
MAP	Manufacturing Automation Protocol
MB	Megabytes
Mbps	Megabits per second (Million bits per second)
MFI	MainFrame Interactive
MMS	Manufacturing Messaging Services
NADN	Nearest Active Downstream Neighbor
NAUN	Nearest Active Upstream Neighbor
NNI	Network Node interface
NRZI	Non Return to Zero Inverted
NT	Network Termination
NTRI	NCP Token-Ring Interface
OC-n	Optical Carrier level n
OSI	Open Systems Interconnection
PABX	Private Automatic Branch Exchange
PBX	Private Branch Exchange
PC	Personal Computer
PDU	Protocol Data Unit
PHY	Physical Layer
PMD	Physical Medium Dependent
PON	Passive Optical Network
POH	Path Overhead
PRI	Primary Rate Interface (ISDN)
PRM	Protocol Reference Model
QOS	Quality of Service
RAM	Random Access Memory
RF	Radio Frequency
RFC	Request For Comment
RI	Routing Information
RR	Receive Ready
SABME	Set Asynchronous Balanced Mode Extended (Command)
SAP	Service Access Point
SAR	Segmentation and Reassembly
SDH	Synchronous Digital Hierarchy
SDLC	Synchronous Data Link Control
SIP	SMDS Interface Protocol
SMDS	Switched Multi-Megabit Data Service
SMT	Station Management
SNA	Systems Network Architecture
SNI	Subscriber-Network Interface (SMDS)
SNI	SNA Network Interconnection (SNA)
SOH	Section Overhead

SPE	Synchronous Payload Envelope (Sonet/SDH)
SPN	Subscriber Premises Network
SRPI	Server/Requester Programming Interface
SSAP	Source Service Access Point
STM	Synchronous Transfer mode
STP	Shielded Twisted Pair
TA	Terminal Adapter
TCP/IP	Transmission Control Protocol/Internet Protocol
TDM	Time Division Multiplexing
TE	Terminal Equipment
THT	Token Holding Timer (FDDI)
TIC	Token-Ring Interface Coupler
TR	Token-Ring
TRA	Token-Ring Adapter
TRM	Token-Ring Multiplexor
TRSS	Token-Ring SubSystem
TRT	Token Rotation Timer (FDDI)
TTP	Telephone Twisted Pair (Wiring)
TTP	Timed Token Protocol (FDDI)
TTRT	Target Token Rotation Time
UA	Unnumbered Acknowledgement
UNI	User to Network Interface
UTP	Unshielded Twisted Pair
VAD	Voice Activity Detector
VBR	Variable Bit Rate
VC	Virtual Circuit (X.25)
VC	Virtual Channel (ATM)
VCI	Virtual Channel Identifier (ATM,DQDB)
VP	Virtual Path
VPI	Virtual Path Identifier
WAN	Wide Area Network
WOMAN	Wideband Optical Metropolitan Area Network
XC	Cross Connect
XID	Exchange Identification

Bibliography

The following publications contain more information on related topics.

General References

Digital Telephony and Network Integration Bernhard E. Keiser and Eugene Strange Van Nostrand Reinhold Company Limited, New York. 1985.

Megabit Data Communications John T.Powers, Jr. and Henry H. Stair II. Prentice-Hall Inc., New Jersey. 1990.

Metropolitan Area Networks: Concepts, Standards and Services. Gary C. Kessler and David A. Train McGraw-Hill Inc. New York. 1991.

Digital Signaling Technology

A Tutorial on Two-Wire Digital Transmission in the Loop Plant Syed V. Ahamed, Peter P Bohn and N. L. Gottfried. IEEE Transactions on Communications, Vol. Com-29, No 11, November 1981.

Line Codes for Digital Subscriber Lines Joseph W. Lechleider. IEEE Communications Magazine, September 1989.

Digital Subscriber Line Technology Facilitates a Graceful Transition from Copper to Fiber David L Waring, Joseph W Lechleider and To Russell Hsing. IEEE Communications Magazine, March 1991.

Study of the Feasibility and Advisability of Digital Subscriber Lines Operating at rates substantially in excess of the Basic Access Rate. ANSII Committee T1E1.4, Technical Report 91-002R4.

A Discrete Multitone Transceiver System for HDSL Applications. Jacky S.Chow, Jerry C. Tu and John M. Cioffi. IEEE Journal on Selected Areas in Communication. August 1991.

High Bit Rate Digital Subscriber Lines: A Review of HDSL Progress Joseph W. Lechleider IEEE Journal on Selected Areas in Communication. August 1991.

Optical Networks

Wavelength Domain Optical Network Techniques Godfrey R. Hill Proceedings of the IEEE, January 1989.

Broadband Photonic Switching using Guided Wave Fabrics Narinder K. Ailawadi, Rod C. Alferness, Glen D Bergland and Richard A. Thompson IEEE LTS, May 1991.

ISDN

ISDN Data Link Control - Architecture Reference IBM Order Number SC31-6826.

ISDN Circuit Switched Signalin Control - Architecture Reference IBM Order Number SC31-6827.

SONET/SDH

SONET - Now It's the Standard Optical Network Ralph Ballart and Yau-Chau Ching. IEEE Communications Magazine, March 1989.

CCITT Recommendations G.707, G.708 and G.709 CCITT Blue Book 1988 - Synchronous Digital Hierarchy

Asynchronous Transfer Mode (ATM)

Broadband ISDN and Asynchronous Transfer Mode (ATM) by Steven E. Minzer. IEEE Communications Magazine, September 1989.

Layered ATM Systems and Architectural Concepts for Subscribers' Premises Networks by Jan P. Vorstermans and Andre P. De Vleeschouwer. IEEE Journal of Selected Areas in Communications, Vol 6, No 9, December 1988.

Broad-Band ATM network Architecture Based on Virtual Paths Ken-Ichi Sato, Satoru Ohta and Ikuo Tokizawa. IEEE Transactions on Communications, August 1990.

Voice Packetisation and Compression in Broadband ATM Networks Kotikalapudi Sriram, R. Scott McKinney and Mostafa Hasheim Sherif. IEEE J. on Selected Areas in Communications, April 1991.

Virtual Path and Link Capacity Design for ATM Networks Youichi Sato and Ken-Ichi Sato. IEEE Journal on Selected Areas in Communications, January 1991.

Draft Recommendations I.113, I.121, I.150, I.211, I.327, I.311, I.361, I.413, I.432 and I.610. CCITT Study Group 18. Geneva, January 1990.

Token Ring

IBM Token-Ring Network Technology IBM Order Number GA27-3732.

IBM Multisegment LAN Design Guidelines IBM Order Number GG24-3398.

FDDI

Performance Analysis of FDDI Token Ring Networks: Effect of Parameters and Guidelines for Setting TTRT Raj Jain. IEEE LTS Magazine. May 1991.

DQDB

Introduction to Switched Multi-Megabit Data Service (SMDS), An Early Broadband Service Christine F. Hemrick and Lawrence J. Lang. Proceedings of the International Switching Symposium, Session A3, Stockholm, June 1990.

The IEEE 802.6 Physical Layer Convergence Procedures Richard Brandwein, Tracy Cox and James Dahl. IEEE LCS Magazine, May 1989.

New Proposal Extends the Reach of Metro Area Nets John L. Hullett. Data Communications Magazine, February 1988.

MetaRing

MetaRing - A Full-Duplex Ring with Fairness and Spatial Reuse Israel Cidon and Yoram Ofek. IEEE Infocom 90, San Fransisco, CA. June 1990.

Integration of Connection and Connectionless Traffic on the Metaring Yoram Ofek. IEEE Workshop on Metropolitan Area Networks, 1990.

CRMA and CRMA-II

Cyclic-Reservation Multiple-Access Scheme for Gbit/s LANs and MANs based on Dual-Bus Configuration M. Mehdi Nassehi. EFOC/LAN 90.

CRMA: An Access Scheme for High Speed LANs and MANs M. Mehdi Nassehi. Supercomm/ICC '90, Atlanta, Georgia. April 1990.

CRMA-II: A Gbit/s MAC Protocol for Ring and Bus Networks with Immediate Access Capability H.R. van As, W. W. Lemppenau, P. Zafiropulo and E. A. Zurfluh EFOC/LAN 91.

Performance of a Gbps Reservation Based Ring with Fairness H.R Van As and P. Zafiropolo ANSI X3T9.5 FDDI Standardisation Meeting Fort Lauderdale Oct 91.

DQMA and CRMA: New Access Schemes for Gbit/s LANs and MANs Hans R Muller, M Mehdi Nassehi, Johnny W. Wong, Erwin Zurfluh, Werner Bux and Pitro Zafiropulo. Proceedings of IEE Infocom '90, San Fransisco, June 1990.

Configuration Control for Bus Networks Peter Heinzmann, Hans Müller and Ken Wilson. Proc. 15th Annual Conf. on Local Computer Networks, Minneapolis, MN. October 1990.

Fast Packet Switching

Paris: An Approach to Integrated High-Speed Private Networks Israel Cidon and Inder S. Gopal. International Journal of Digital and Analogue Cabled Systems., Vol. 1., 1988, pp 77-85.

Protocol Issues

Implementation of Physical and Media Access Protocols for High-Speed Networks by Morten Skov. IEEE Communications Magazine, June 1989.

Routing and Flow Control in High-Speed Wide-Area Networks by Nicholas F. Maxemchuk and Magda El Zarki. Proceedings of the IEEE, Vol 78, No.1, January 1990.

XTP Protocol Definition Revision 3.4, Protocol Engines Incorporated, 1900 State St., Suite D, Santa Barbara, California 93101, 1989.

The Xpress Transfer Protocol (XTP) - A Tutorial by Robert M Sanders. Computer Science Report No. TR-89-10, Department of Computer Science, University of Virginia.

A Survey of Light-Weight Transport Protocols for High-Speed Networks by Willibald Doeringer, Doug Dykeman, Matthias Kaisersworth, Bernd Meister, Harry Rudin and Robin Williamson. IEEE Transactions on Communications, 38-11, (1990).

Control Mechanisms for High Speed Networks by Israel Cidon and Inder S. Gopal. ICC '90.

Real-Time Packet Switching: A Performance Analysis by Israel Cidon, Inder Gopal, George Grover and Moshe Sidi. IEEE Journal of Selected Areas in Communications, Vol. 6. No. 9, December 1988.

Congestion Control for High Speed Packet Networks by Krishna Bala, Israel Cidon and Khosrow Sohraby. Infocom 90.

A Blind Voice Packet Synchronization Strategy by Joong Ma and Inder Gopal. IBM Research Report. RC 13893. 1988.

Linear Broadcast Routing Ching-Tsun Chou and Inder S. Gopal. Journal of Algorithms 10, 490-517 (1989).

Packet Video and Its Integration into the Network Architecture Gunnar Karlsson and Martin Vetterli. IEEE J. on Selected areas in Communication, June 1989.

Surveys

A Taxonomy of Broadband Integrated Switching Architectures George E. Daddis and H. C. Torng. IEEE Communications Magazine, May 1989.

Rationale, Directions and Issues Surrounding High Speed Networks Imrich Chlamtac and William R. Franta. Proceedings of the IEEE, Vol. 78, No. 1. January 1990.

Standards

Distributed Queue Dual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN) IEEE 802.6 Working Group. Unapproved draft, October 1st 1990.

Glossary

A

access control byte. In the IBM Token-Ring Network, the byte following the start delimiter of a token or frame that is used to control access to the ring.

access priority. The maximum priority that a token can have for the adapter to use it for transmission.

access unit. A unit that allows multiple attaching devices access to a token-ring network at a central point such as a wiring closet or in an open work area.

- *Access unit* refers to either IBM 8228s or IBM 8230s.
- *Multistation access unit* refers specifically to IBM 8228s.
- *Controlled access unit* refers specifically to IBM 8230s.

active monitor. A function in a single adapter on a token-ring network that initiates the transmission of tokens and provides token error recovery facilities. Any active adapter on the ring has the ability to provide the active monitor function if the current active monitor fails.

adapter. In a LAN, within a communicating device, a circuit card that, with its associated software and/or microcode, enables the device to communicate over the network.

adapter address. Twelve hexadecimal digits that identify a LAN adapter.

adjusted ring length (ARL). In a multiple-wiring-closet ring, the sum of all wiring closet-to-wiring closet cables in the main ring path less the length of the shortest of those cables.

all-routes broadcast frame. A frame that has bits in the routing information field set to indicate that the frame is to be sent to all LAN segments in the network (across all bridges, even if multiple paths allow multiple copies of the frame to arrive at some LAN segments). The destination address is not examined and plays no role in bridge routing.

all-stations broadcast frame. A frame whose destination address bits are set to all ones. All stations on any LAN segment on which the frame appears will copy it. The routing information, not the destination address, determines which LAN segments the frame appears on. All-stations broadcasting is independent of all-routes broadcasting; the two can be done simultaneously or one at a time.

application program interface (API). The formally defined programming language interface that is between an IBM system control program or a licensed program and the user of the program.

attenuation. A decrease in magnitude of current, voltage, or electrical or optical power of a signal in transmission between points. It may be expressed in decibels or nepers.

automatic single-route broadcast. A function used by some IBM bridge programs to determine the correct settings for, and set the bridge single-route broadcast configuration parameters dynamically, without operator intervention. As bridges enter and leave the network, the parameter settings may need to change to maintain a single path between any two LAN segments for single-route broadcast messages. See also *single-route broadcast.*

B

backbone LAN segment. In a LAN multiple segment configuration, a centrally located LAN segment to which other LAN segments are connected by means of bridges. In a hierarchical network, the LAN segment that is at the highest level of the hierarchy.

backup path. In an IBM Token-Ring Network, an alternative path for signal flow through access units and their main ring path cabling. The backup path allows recovery of the operational portion of the network while problem determination procedures are being performed.

bandwidth. In analogue communications this is difference, expressed in hertz, between the highest and the lowest frequencies of a range of frequencies. For example, analog transmission by recognizable voice telephone requires a bandwidth of about 3000 hertz (3 kHz).

In digital communications this is often used to mean the total available bit rate of a digital channel. At other times it can mean the symbol rate (baud rate) of a digital channel.

baseband. (1) A frequency band that uses the complete bandwidth of a transmission medium. Contrast with *broadband, carrierband.* (2) A method of data transmission that encodes, modulates, and impresses information on the transmission medium without shifting or altering the frequency of the information signal.

baseband local area network. A local area network in which information is encoded, multiplexed, and transmitted without modulation of a carrier.

beaconing. An error-indicating function of token-ring adapters that assists in locating a problem causing a hard error on a token-ring network.

bridge. (1) An attaching device that connects two LAN segments to allow the transfer of information from one LAN segment to the other. A bridge may connect the LAN segments directly by network adapters and software in a single device, or may connect network adapters in two separate devices through software and use of a telecommunications link between the two adapters. (2) A functional unit that connects two LANs that use the same logical link control (LLC) procedures but may use the same or different medium access control (MAC) procedures. Contrast with *gateway* and *router*.

bridge ID. The bridge label combined with the adapter address of the adapter connecting the bridge to the LAN segment with the lowest LAN segment number; it is used by the automatic single-route broadcast function in IBM bridge programs.

bridge label. A 2-byte hexadecimal number that the user can assign to each bridge. See *bridge ID*.

bridge number. The bridge identifier that the user specifies in the bridge program configuration file. The bridge number distinguishes among parallel bridges. Parallel bridges connect the same two LAN segments.

broadband. (1) A frequency band between any two non-zone frequencies. (2) A frequency band divisible into several narrower bands so that different kinds of transmissions such as voice, video, and data transmission can occur at the same time. Synonymous with *wideband*. Contrast with *baseband*, *carrierband*.

broadband local area network (LAN). A local area network (LAN) in which information is encoded, multiplexed, and transmitted through modulation of a carrier.

broadcast. Simultaneous transmission of data to more than one destination.

bus. (1) In a processor, a physical facility on which data is transferred to all destinations, but from which only addressed destinations may read in accordance with appropriate conventions. (2) A network configuration in which nodes are interconnected through a bidirectional transmission medium. (3) One or more conductors used for transmitting signals or power.

bus network. A network configuration that provides a bidirectional transmission facility to which all nodes are attached. A sending node transmits in both directions to the ends of the bus. All nodes in the path examine and may copy the message as it passes.

C

cable loss. The amount of radio frequency (RF) signal attenuation caused by a cable. See *attenuation*.

carrier. A wave or pulse train that may be varied by a signal bearing information to be transmitted over a communication system.

carrierband. A frequency band in which the modulated signal is superimposed on a carrier signal (as differentiated from baseband), but only one channel is present on the medium (as differentiated from broadband). Contrast with *baseband*, *broadband*.

carrier sense. In a local area network, an ongoing activity of a data station to detect whether another station is transmitting.

carrier sense multiple access with collision avoidance (CSMA/CA) network. A bus network in which the medium access control protocol requires carrier sense, and in which a station always starts transmission by sending a jam signal. If there is no collision with jam signals from other stations, it begins sending data; otherwise, it stops transmission and then tries again later.

carrier sense multiple access with collision detection (CSMA/CD) network. A bus network in which the medium access control protocol requires carrier sense, and in which exception conditions caused by collision are solved by retransmission.

coaxial (coax) cable. A cable consisting of one conductor, usually a small copper tube or wire, within and insulated from another conductor of a larger diameter, usually copper tubing or copper braid.

coaxial tap. A physical connection to a coaxial cable.

communication network management (CNM). The process of designing, installing, operating, and managing distribution of information and control among users of communication systems.

controller. A unit that controls input/output operations for one or more devices.

crosstalk. The disturbance caused in a circuit by an unwanted transfer of energy from another circuit.

cyclic redundancy check (CRC). Synonym for *frame check sequence (FCS)*.

D

datagram. A particular type of information encapsulation at the network layer of the adapter protocol. No explicit acknowledgment for the information is sent by the receiver. Instead, transmission relies on the "best effort" of the link layer.

data link control (DLC) layer. (1) In SNA or Open Systems Interconnection (OSI), the layer that schedules data transfer over a link between two nodes and performs error control for the link. Examples of DLC are synchronous data link control (SDLC) for serial-by-bit connection and DLC for the System/370 channel. (2) See *logical link control (LLC) sublayer*, *medium access control (MAC) sublayer*. **Note:** The DLC layer is usually independent of the physical transport mechanism and ensures the integrity of data that reach the higher layers.

designated bridge. In a LAN using automatic single-route broadcast, a bridge that forwards single-route broadcast frames. See also *root bridge*, *standby bridge*.

destination. Any point or location, such as a node, station, or particular terminal, to which information is to be sent.

destination address. A field in the medium access control (MAC) frame that identifies the physical location to which information is to be sent. Contrast with *source address*.

destination service access point (DSAP). The service access point for which a logical link control protocol data unit (LPDU) is intended.

destination service access point (DSAP) address. The address of the link service access point (LSAP) for which a link protocol data unit (LPDU) is intended. Also, a field in the LPDU.

differential Manchester encoding. A transmission encoding scheme in which each bit is encoded as a two-segment signal with a signal transition (polarity change) at either the bit time or half-bit time. Transition at a bit time represents a 0. No transition at a bit time indicates a 1.

Note: This coding scheme allows simpler receive/transmit and timing recovery circuitry and a smaller delay per station than achieved with block codes. It also allows the two wires of a twisted pair to be interchanged without causing data errors.

downstream physical unit (DSPU). A controller or a workstation downstream from a gateway that is attached to a host.

E

Early Token Release (ETR). In token-ring and Fiber Distributed Data Interface (FDDI) networks, a function that allows a transmitting adapter to release a new token as soon as it has completed frame transmission, whether or not the frame header has returned to that adapter.

EBCDIC. Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

electromagnetic interference (EMI). A disturbance in the transmission of data on a network resulting from the magnetism created by a current of electricity.

Ethernet network. A baseband LAN with a bus topology in which messages are broadcast on a coaxial cable using a carrier sense multiple access/collision detection (CSMA/CD) transmission method.

F

Federal Communications Commission (FCC). A board of commissioners appointed by the President under the Communications Act of 1934, having the power to regulate all interstate and foreign communications by wire and radio originating in the United States.

Fiber Distributed Data Interface (FDDI). A high-performance, general-purpose, multi-station network designed for efficient operation with a peak data transfer rate of 100 Mbps. It uses token-ring architecture with optical fiber as the transmission medium over distances of several kilometers.

filtered frames. Frames that arrive at a bridge adapter but are not forwarded across the bridge, because of criteria specified in a filter program used with the bridge program.

frame. The unit of transmission in some LANs, including the IBM Token-Ring Network and the IBM PC Network. It includes delimiters, control characters, information, and checking characters. On a token-ring network, a frame is created from a token when the token has data appended to it. On a token bus network (IBM PC Network), all frames including the token frame contain a preamble, start delimiter, control address, optional data and checking characters, end delimiter, and are followed by a minimum silence period.

frame check sequence (FCS). (1) A system of error checking performed at both the sending and receiving station after a block check character has been accumulated. (2) A numeric value derived from the bits in a message that is used to check for any bit errors in transmission. (3) A redundancy check in which the check key is generated by a cyclic

algorithm. Synonymous with *cyclic redundancy check (CRC)*.

frequency pair. In the broadband IBM PC Network, the two frequencies or channels used by an adapter: one to transmit data to the network, and one to receive data from the network.

functional address. In IBM network adapters, a special kind of group address in which the address is bit-significant, each "on" bit representing a function performed by the station (such as "Active Monitor," "Ring Error Monitor," "LAN Error Monitor," or "Configuration Report Server").

G

gateway. A device and its associated software that interconnect networks or systems of different architectures. The connection is usually made above the reference model network layer. Contrast with *bridge* and *router*.

group address. In a LAN, a locally administered address assigned to two or more adapters to allow the adapters to copy the same frame. Contrast *locally administered address* with *universally administered address*.

group SAP. A single address assigned to a group of service access points (SAPs). See also *group address*.

H

hard error. An error condition on a network that requires that the source of the error be removed or that the network be reconfigured before the network can resume reliable operation. See also *beaconing*. Contrast with *soft error*.

header. The portion of a message that contains control information for the message such as one or more destination fields, name of the originating station, input sequence number, character string indicating the type of message, and priority level for the message.

"hello" message. A message used by the automatic single-route broadcast function of IBM bridge programs to detect what bridges enter and leave the network and to cause single-route broadcast parameters to be reset accordingly. The root bridge sends a "hello" message on the network every 2 seconds.

hertz (Hz). A unit of frequency equal to one cycle per second.

hierarchical network. A multiple-segment network configuration providing only one path through intermediate segments between source segments and destination segments. Contrast with *mesh network*.

hop count. The number of bridges through which a frame has passed on the way to its destination.

Note: Hop count applies to all broadcast frames except single-route broadcast frames.

hop count limit. The maximum number of bridges through which a frame may pass on the way to its destination.

I

idles. Signals sent along a ring network when neither frames nor tokens are being transmitted.

impedance. The combined effect of resistance, inductance, and capacitance on a signal at a particular frequency.

individual address. An address that identifies a particular network adapter on a LAN. See also *locally administered address* and *universally administered address*.

International Organization for Standardization (ISO). An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

J

jitter. Undesirable variations in the arrival time of a transmitted digital signal.

L

LAN adapter. The circuit card within a communicating device (such as a personal computer) that, together with its associated software, enables the device to be attached to a LAN.

LAN multicast. The sending of a transmission frame intended to be accepted by a group of selected data stations on the same LAN.

LAN segment. (1) Any portion of a LAN (for example, a single bus or ring) that can operate independently but is connected to other parts of the establishment network via bridges. (2) An entire ring or bus network without bridges. See *ring segment*.

LAN segment number. The identifier that uniquely distinguishes a LAN segment in a multi-segment LAN.

latency. The time interval between the instant at which an instruction control unit initiates a call for

data and the instant at which the actual transfer of data begins. Synonymous with *waiting time*. See also *ring latency*.

layer. (1) One of the seven levels of the Open Systems Interconnection reference model. (2) In open systems architecture, a collection of related functions that comprise one level of hierarchy of functions. Each layer specifies its own functions and assumes that lower level functions are provided. (3) In SNA, a grouping of related functions that are logically separate from the functions of other layers. Implementation of the functions in one layer can be changed without affecting functions in other layers.

limited broadcast. Synonym for *single-route broadcast*.

link. (1) The logical connection between nodes including the end-to-end link control procedures. (2) The combination of physical media, protocols, and programming that connects devices on a network. (3) In computer programming, the part of a program, in some cases a single instruction or an address, that passes control and parameters between separate portions of the computer program. (4) To interconnect items of data or portions of one or more computer programs. (5) In SNA, the combination of the link connection and link stations joining network nodes.

link station. (1) A specific place in a service access point (SAP) that enables an adapter to communicate with another adapter. (2) A protocol machine in a node that manages the elements of procedure required for the exchange of data traffic with another communicating link station. (3) A logical point within a SAP that enables an adapter to establish connection-oriented communication with another adapter. (4) In SNA, the combination of hardware and software that allows a node to attach to and provide control for a link.

lobe. In the IBM Token-Ring Network, the section of cable (which may consist of several cable segments) that connects an attaching device to an access unit.

local area network (LAN). A computer network located on a user's premises within a limited geographical area.

Note: Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation.

local bridge function. Function of an IBM bridge program that allows a single bridge computer to connect two LAN segments (without using a telecommunication link). Contrast with *remote bridge function*.

locally administered address. An adapter address that the user can assign to override the universally administered address. Contrast with *universally administered address*.

logical connection. In a network, devices that can communicate or work with one another because they share the same protocol. See also *physical connection*.

logical link control protocol (LLC protocol). In a local area network, the protocol that governs the exchange of frames between data stations independently of how the transmission medium is shared.

logical link control protocol data unit (LPDU). The unit of information exchanged between network layer entities in different nodes. The LPDU consists of the destination service access point (DSAP) and source service access point (SSAP) address fields, the control field, and the information field (if present).

logical link control (LLC) sublayer. One of two sublayers of the ISO Open Systems Interconnection data link layer (which corresponds to the SNA data link control layer), proposed for LANs by the IEEE Project 802 Committee on Local Area Networks and the European Computer Manufacturers Association (ECMA). It includes those functions unique to the particular link control procedures that are associated with the attached node and are independent of the medium; this allows different logical link protocols to coexist on the same network without interfering with each other. The LLC sublayer uses services provided by the medium access control (MAC) sublayer and provides services to the network layer.

logical unit (LU). In SNA, a port through which an end user accesses the SNA network in order to communicate with another end user and through which the end user accesses the functions provided by system services control points (SSCPs). An LU can support at least two sessions, one with an SSCP and one with another LU, and may be capable of supporting many sessions with other logical units.

LU type 6.2. A type of logical unit that supports sessions between two application programs in a distributed data processing environment using the SNA general data stream, which is a structured-field data stream, between two type 5 nodes, a type 5 node and a type 2.1 node, and two type 2.1 nodes.

M

MAC frame. Frames used to carry information to maintain the ring protocol and for exchange of management information.

MAC protocol. (1) In a local area network, the protocol that governs communication on the transmission medium without concern for the physical

characteristics of the medium, but taking into account the topological aspects of the network, in order to enable the exchange of data between data stations. See also *logical link control protocol (LLC protocol)*. (2) The LAN protocol sublayer of data link control (DLC) protocol that includes functions for adapter address recognition, copying of message units from the physical network, and message unit format recognition, error detection, and routing within the processor.

MAC segment. An individual LAN communicating through the medium access control (MAC) layer within this network.

main ring path. In the IBM Token-Ring Network, the part of the ring made up of access units, repeaters, converters, and the cables connecting them. See also *backup path*.

Manchester encoding. See *differential Manchester encoding*.

Manufacturing Automation Protocol (MAP). A broadband LAN with a bus topology that passes tokens from adapter to adapter on a coaxial cable.

medium access control frame. See *MAC frame*.

medium access control (MAC) protocol. In a local area network, the part of the protocol that governs communication on the transmission medium without concern for the physical characteristics of the medium, but taking into account the topological aspects of the network, in order to enable the exchange of data between data stations.

medium access control sublayer (MAC sublayer). In a local area network, the part of the data link layer that applies medium access control and supports topology-dependent functions. The MAC sublayer uses the services of the physical layer to provide services to the logical link control sublayer and all higher layers.

mesh network. A multiple-segment network configuration providing more than one path through intermediate LAN segments between source and destination LAN segments. Contrast with *hierarchical network*.

Micro Channel. The architecture used by IBM Personal System/2 computers, Models 50 and above. This term is used to distinguish these computers from personal computers using a PC I/O channel, such as an IBM PC, XT, or an IBM Personal System/2 computer, Model 25 or 30.

N

NetView. A host-based IBM licensed program that provides communication network management (CNM) or communications and systems management (C&SM) services.

Network Basic Input/Output System (NetBIOS). A message interface used on LANs to provide message, print server, and file server functions. The IBM NetBIOS application program interface (API) provides a programming interface to the LAN so that an application program can have LAN communication without knowledge and responsibility of the data link control (DLC) interface.

network layer. (1) In the Open Systems Interconnection reference model, the layer that provides for the entities in the transport layer the means for routing and switching blocks of data through the network between the open systems in which those entities reside. (2) The layer that provides services to establish a path between systems with a predictable quality of service. See *Open Systems Interconnection (OSI)*.

network management. The conceptual control element of a station that interfaces with all of the architectural layers of that station and is responsible for the resetting and setting of control parameters, obtaining reports of error conditions, and determining if the station should be connected to or disconnected from the network.

noise. (1) A disturbance that affects a signal and that can distort the information carried by the signal. (2) Random variations of one or more characteristics of any entity, such as voltage, current, or data. (3) Loosely, any disturbance tending to interfere with normal operation of a device or system.

non-broadcast frame. A frame containing a specific destination address and that may contain routing information specifying which bridges are to forward it. A bridge will forward a non-broadcast frame only if that bridge is included in the frame's routing information.

O

observing link. The reporting link (or authorization level) between a bridge and a network management program that authorizes the network management program to perform all network management functions except those restricted to the controlling link. (The restricted functions include removing adapters from a ring, changing certain bridge configuration parameters, and enabling or disabling certain bridge functions.)

Open Systems Interconnection (OSI). (1) The interconnection of open systems in accordance with specific ISO standards. (2) The use of standardized procedures to enable the interconnection of data processing systems.

Note: OSI architecture establishes a framework for coordinating the development of current and future standards for the interconnection of computer systems. Network functions are divided into seven layers. Each layer represents a group of related data processing and communication functions that can be carried out in a standard way to support different applications.

Open Systems Interconnection (OSI) architecture. Network architecture that adheres to a particular set of ISO standards that relates to Open Systems Interconnection.

Open Systems Interconnection (OSI) reference model. A model that represents the hierarchical arrangement of the seven layers described by the Open Systems Interconnection architecture.

P

packet. In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. Synonymous with *data frame*.

parallel bridge. One of the two or more bridges that connect the same two LAN segments in a network.

path. (1) In a network, any route between any two nodes. (2) The route traversed by the information exchanged between two attaching devices in a network..

path cost. A value, maintained by each IBM bridge program that uses the automatic single-route bridge function, that indicates to the automatic single-route bridge function the relative length of the path between the root bridge and a designated or standby bridge.

PC Network. An IBM broadband or baseband LAN with a bus topology in which messages are broadcast from PC Network adapter to PC Network adapter.

physical connection. The ability of two connectors to mate and make electrical contact. In a network, devices that are physically connected can communicate only if they share the same protocol. See also *logical connection*.

physical layer. In the Open Systems Interconnection reference model, the layer that provides the mechanical, electrical, functional, and procedural means to establish, maintain, and release physical connections over the transmission medium.

physical unit (PU). In SNA, the component that manages and monitors the resources of a node, such as attached links and adjacent link stations, as requested by a system services control point (SSCP) via an SSCP-SSCP session.

port. (1) An access point for data entry or exit. (2) A connector on a device to which cables for other devices such as display stations and printers are attached. Synonymous with *socket*.

primary adapter. In a personal computer that is used on a LAN and that supports installation of two network adapters, the adapter that uses standard (or default) mapping between adapter-shared RAM, adapter ROM, and designated computer memory segments. The primary adapter is usually designated as adapter 0 in configuration parameters. Contrast with *alternate adapter*.

primary path. In the IBM Token-Ring Network, the normal flow of signals through access units and the main ring path cabling.

protocol. (1) A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication.(2) In SNA, the meanings of and the sequencing rules for requests and responses used for managing the network, transferring data, and synchronizing the states of network components. (3) A specification for the format and relative timing of information exchanged between communicating parties.

R

random access memory (RAM). A computer's or adapter's volatile storage area into which data may be entered and retrieved in a nonsequential manner.

read-only memory (ROM). A computer's or adapter's storage area whose contents cannot be modified by the user except under special circumstances.

remote bridge function. The function of some IBM bridge programs that allows two bridge computers to use a telecommunications link to connect two LAN segments. Contrast with *local bridge function*.

repeater. In a network, a device that amplifies or regenerates data signals in order to extend the distance between attaching devices.

ring error monitor (REM). A function that compiles error statistics reported by adapters on a network, analyzes the statistics to determine probable error cause, sends reports to network manager programs, and updates network status conditions. It assists in fault isolation and correction.

Request for Comment (RFC). The Internet Protocol suite is evolving through the mechanism of Request for Comments (RFC). Research ideas and new protocols (mostly application protocols) are brought to the attention of the internet community in the form of an RFC. Some protocols are so useful that they are recommended to be implemented in all future implementations of TCP/IP; that is, they become recommended protocols. Each RFC has a status attribute to indicate the acceptance and stage of evolution this idea has in the TCP/IP protocol suite. Software developers use RFCs as a reference to write TCP/IP software.

ring in (RI). In an IBM Token-Ring Network, the receive or input receptacle on an access unit or repeater.

ring latency. In an IBM Token-Ring Network, the time, measured in bit times at the data transmission rate, required for a signal to propagate once around the ring. Ring latency includes the signal propagation delay through the ring medium, including drop cables, plus the sum of propagation delays through each data station connected to the Token-Ring Network.

ring network. A network configuration in which a series of attaching devices is connected by unidirectional transmission links to form a closed path. A ring of an IBM Token-Ring Network is referred to as a LAN segment or as a Token-Ring Network segment.

ring out (RO). In an IBM Token-Ring Network, the transmit or output receptacle on an access unit or repeater.

ring segment. A ring segment is any section of a ring that can be isolated (by unplugging connectors) from the rest of the ring. A segment can consist of a single lobe, the cable between access units, or a combination of cables, lobes, and/or access units. See *cable segment*, *LAN segment*.

root bridge. In a LAN containing IBM bridges that use automatic single-route broadcast, the bridge that sends the "hello" message on the network every 2 seconds. Automatic single-route broadcast uses the message to detect when bridges enter and leave the network, and to change single-route broadcast parameters accordingly. See also *designated bridge*, *standby bridge*.

router. An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer. Contrast with *bridge* and *gateway*.

routing. (1) The assignment of the path by which a message will reach its destination. (2) The forwarding of a message unit along a particular path through a network, as determined by the parameters carried in the message unit, such as the destination network address in a transmission header.

S

segment. See *LAN segment*, *ring segment*.

server. (1) A device, program, or code module on a network dedicated to providing a specific service to a network. (2) On a LAN, a data station that provides facilities to other data stations. Examples are a file server, print server, and mail server.

service access point (SAP). (1) A logical point made available by an adapter where information can be received and transmitted. A single SAP can have many links terminating in it. (2) In Open Systems Interconnection (OSI) architecture, the logical point at which an n + 1-layer entity acquires the services of the n-layer. For LANs, the n-layer is assumed to be data link control (DLC). A single SAP can have many links terminating in it. These link "end-points" are represented in DLC by link stations.

session. (1) A connection between two application programs that allows them to communicate. (2) In SNA, a logical connection between two network addressable units that can be activated, tailored to provide various protocols, and deactivated as requested. (3) The data transport connection resulting from a call or link between two devices. (4) The period of time during which a user of a node can communicate with an interactive system, usually the elapsed time between log on and log off. (5) In network architecture, an association of facilities necessary for establishing, maintaining, and releasing connections for communication between stations.

single-route broadcast. The forwarding of specially designated broadcast frames only by bridges which have single-route broadcast enabled. If the network is configured correctly, a single-route broadcast frame will have exactly one copy delivered to every LAN segment in the network. Synonymous with *limited broadcast*. See also *automatic single-route broadcast*.

socket. Synonym for *port* .

soft error. An intermittent error on a network that causes data to have to be transmitted more than once to be received. A soft error affects the network's performance but does not, by itself, affect the network's overall reliability. If the number of soft errors becomes excessive, reliability is affected. Contrast with *hard error*.

source address. A field in the medium access control (MAC) frame that identifies the location from which information is sent. Contrast with *destination address*.

source service access point (SSAP). The service access point (SAP) from which a logical link control protocol data unit (LPDU) is originated.

source service access point (SSAP) address. The address of the link service access point (LSAP) from which a link protocol data unit (LPDU) is originated. Also, a field in the LPDU.

standby bridge. In a LAN using automatic single-route broadcast, a bridge that does not forward single-route broadcast frames. A standby bridge is a parallel bridge or is in a parallel path between two LAN segments. See also *designated bridge*, *root bridge*.

station. (1) A communication device attached to a network. The term used most often in LANs is an *attaching device* or *workstation*. (2) An input or output point of a system that uses telecommunication facilities; for example, one or more systems, computers, terminals, devices, and associated programs at a particular location that can send or receive data over a telecommunication line.

subsystem. A secondary or subordinate system, or programming support, usually capable of operating independently of or asynchronously with a controlling system.

symbolic name. In a LAN, a name that may be used instead of an adapter or bridge address to identify an adapter location.

T

telephone twisted pair. One or more twisted pairs of copper wire in the unshielded voice-grade cable commonly used to connect a telephone to its wall jack. Also referred to as "unshielded twisted pair"

throughput. (1) A measure of the amount of work performed by a computer system over a given period of time, for example, number of jobs per day. (2) A measure of the amount of information transmitted over a network in a given period of time. For example, a network's data transfer rate is usually measured in bits per second.

token. A sequence of bits passed from one device to another on the token-ring network that signifies permission to transmit over the network. It consists of a starting delimiter, an access control field, and an end delimiter. The access control field contains a bit that indicates to a receiving device that the token is ready to accept information. If a device has data to send along the network, it appends the data to the token. When data is appended, the token then becomes a frame. See *frame*.

token-bus network. A bus network in which a token-passing procedure is used.

token-passing. In a token-ring network, the process by which a node captures a token; inserts a message, addresses, and control information; changes the bit pattern of the token to the bit pattern of a frame; transmits the frame; removes the frame from the ring when it has made a complete circuit; generates another token; and transmits the token on the ring where it can be captured by the next node that is ready to transmit.

token-ring. A network with a ring topology that passes tokens from one attaching device (node) to another. A node that is ready to send can capture a token and insert data for transmission.

token-ring network. (1) A ring network that allows unidirectional data transmission between data stations by a token-passing procedure over one transmission medium so that the transmitted data returns to and is removed by the transmitting station. The IBM Token-Ring Network is a baseband LAN with a star-wired ring topology that passes tokens from network adapter to network adapter.(2) A network that uses a ring topology, in which tokens are passed in a sequence from node to node. A node that is ready to send can capture the token and insert data for transmission. (3) A group of interconnected token rings.

topology. The physical or logical arrangement of nodes in a computer network. Examples include ring topology and bus topology.

transceiver. Any device that can transmit and receive traffic.

translator. In broadband networks, an active device for converting an inbound channel to a higher frequency outbound channel. The conversion is done by removing the inbound carrier, adding the outbound carrier, and amplifying the signal. (A translator amplifies inbound errors and noise distortion.)

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of protocols that allow cooperating computers to share resources across a heterogeneous network.

twisted pair. A transmission medium that consists of two insulated conductors twisted together to reduce noise.

U

uninterruptible power supply (UPS). A buffer between utility power or other power source and a load that requires uninterrupted, precise power. It is usually battery powered.

universally administered address. The address permanently encoded in an adapter at the time of manufacture. All universally administered addresses

are unique. Contrast with *locally administered address*.

unnumbered acknowledgment. A data link control (DLC) command used in establishing a link and in answering receipt of logical link control (LLC) frames.

unshielded twisted pair (UTP). See *telephone twisted pair*.

Index

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

W

X

Readers' Comments

High-Speed Networking Technology
An Introductory Survey

Publication No. GG24-3816-00

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply.

Name _____ Address _____

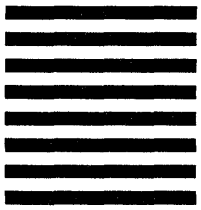
Company or Organization _____ _____

Phone No. _____ _____

Cu
Alc

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Center
Department 985, Building 657
P.O. BOX 12195
RESEARCH TRIANGLE PARK NC 27709-2195

Fold and Tape **Please do not staple** Fold and Tape

Cut
Alor

GG24-3816-00

High-Speed Networking Technology An Introductory Survey

GG24-3816-00

PRINTED IN THE U.S.A.

IBM®