

https://dspace.org/

The development of a method for comparing the operation and performance of metropolitan area networks

Dahl, Bradley

2015-02-20
http://hdl.handle.net/10539/17023

Downloaded from a DSpace repository. https://dspace.org/

THE DEVELOPMENT OF A METHOD FOR COMPARING THE OPERATION AND PERFORMANCE OF METROPOLITAN AREA NETWORKS

BRADLEY DAHL
88-00453/A

DEGREE AWARDED WITH DISTINCTION 24 June 1993

A DISSERTATION SUBMITTED TO THE FACULTY OF ENGINEERING, UNIVERSITY OF THE WITWATERSRAND, IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN ENGINEERING

JOHANNESBURG, 1993

DECLARATION

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Science in Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other University.

B Dahl

(Signature of candidate)

23 day of FEBRUARY 1993

ABSTRACT

A unique method for selection of a protocol for implementing Metropolitan Area Networks (MANs) is discussed in this dissertation report. Given a number of different protocols, this method enables the selection of the protocol which is most suited to the requirements of a given MAN topology and load. The method allows performance of a protocol to be compared from three different perspectives: (1) functional, (2) analytical and (3) load and configuration simulation. The application of the different methods of comparison gives insight into which MAN protocol meets the requirements of a network best. Since established and developing protocols can be compared against each other, the method allows short term and long term networking designs to be effectively planned.

ACKNOWLEDGEMENTS

I would like to acknowledge the asistance of :

Mr Mark Chodos, Project Manager of WEB Systems (Pty) Ltd, First Floor, Suite 103, Infotech Building, 1090, Arcadia. WEB systems allowed me to carry out the CACI LANNET II.5 release 4 simulations on their system.

Dr Ian Kennedy for supplying me with papers outlining MAN developments and research over previous years in the initial stages of the investigation.

Mr Neill Harris, my research supervisor, for his valuable references to papers throughout the course of the research.

CONTENTS

DECLARATION	**ii**
ABSTRACT	**iii**
ACKNOWLEDGEMENTS	**iv**
CONTENTS	**v**
LIST OF TABLES	**viii**
LIST OF FIGURES	**ix**
CHAPTER 1: INTRODUCTION	**1**
1.1 The Aim and Objectives of the Project	1
1.2 Background	1
1.2.1 Different Classes of Networks	2
1.2.2 Example Protocols Being Compared by Means of the Method	3
1.2.3 Application of MANs in Networking	4
1.3 Application of the Study of Methods in Network Engineering	5
1.4 Limitations and Assumptions of the Method	5
1.4.1 Limitations of the Comparison Method	5
1.4.2 Assumptions Made for the Comparison Method	6
1.5 Proposed Solution	6
1.5.1) Determining factors affecting performance in MANs	6
1.5.2) Determining criteria for comparison of MAN protocol performance	8
1.5.3) The derivation of the methods for comparing MAN protocols	9
1.6 Validating the Results and Methods Obtained	9
1.6.1 Validate the applicability of the comparison method	10
1.6.2 Validation of Simulator Protocol Models	10
1.7 Overview of Report Structure	10
CHAPTER 2: SURVEY OF CURRENT DEVELOPMENTS	**12**
2.1 Investigations into the Protocols Used to Illustrate the Method	12
2.2 Approaches to Designing and Planning of MANs	12
2.3 Trends in Network Simulators	13
CHAPTER 3: APPROACH TO COMPARING THE PERFORMANCE OF MANS	**15**
3.1 Bridging Functional Differences in the Example Protocols	15
3.2 Comparing the characteristics of network protocols	15
3.3 Criteria used for comparing network reliability and performance	17
3.4 Classifying criteria and indices for comparison of MAN performance	19
3.4.1) Functional Comparison Criteria	19
3.4.2) Analytical Comparison Criteria	20
3.4.3) Simulation Aided Comparison Criteria	20
CHAPTER 4: FUNCTIONAL COMPARISON OF MANS	**21**
4.1 Available Services	22
4.2 Reliability of the Protocol	23
4.2.1 Topology Strength and Error Recovery	23
4.2.2 Station Hard Error Recovery	24
4.2.3 Soft Error Recovery	25
4.3 Management of the Network	26
4.3.1 Planning and Implementations of the Network	26
4.3.2 Reconfiguration of the Network	27
4.3.3 Growth of the Network	29
4.4 Other Issues	31
4.4.1 Fairness of the Protocol Operation	32
4.4.2 Protocol Flow Control Mechanisms	34

4.4.3 Capacity Planning Constraints of the Network 35
4.5 Summary 35

CHAPTER 5: ANALYTICAL COMPARISON OF MANS 37
5.1 A Standard Configuration 37
5.2 Comparison of Performance 38
5.2.1 Absolute Maximum MAC layer Throughput 38
5.2.2 Bandwidth Allocation 39
5.2.3 Packet Transmission Delays 40

CHAPTER 6: MAN SIMULATOR REQUIREMENTS 42
6.1 Problem Identification 42
6.1.1 Basic Requirements for the Simulator 42
6.1.2 Special Techniques or Algorithms to be Employed 46
6.1.3 Assumptions for Practical Implementation and Operation of the Simulator 48
6.1.4 Tools and Methods for Implementation 49
6.1.5 Resulting Limitations or Restrictions 50
6.2 Identifying the System Components 50
6.2.1 User Interface 51
6.2.2 Simulator Environment 51
6.2.3 Protocol 52
6.3 The Design of the Simulator Environment 52
6.3.1 Components Constituting the Environment 52
6.3.2 Interaction of Environment Components 53
6.4 Design of Protocol objects 54
6.4.1 The functioning of the protocol object 54
6.4.2 Implementing FDDI 59

CHAPTER 7: MAN SIMULATOR IMPLEMENTATION 64
7.1 Implementation and Development of the Simulator 64
7.1.1 Features used for development of the program 64
7.1.2 Limitations imposed on the Environment by the Development System 65
7.1.3 Protocol Limitations 66
7.1.4 Extent of Program Testing 67
7.2 Testing the Operation of the Protocols 67
7.2.1 FDDI Tests 68
7.2.2 DQDB Tests 72
7.3 Application of the Simulator in Making Decisions 77

CHAPTER 8: APPLICATION OF COMPARISON METHOD 79
8.1 LAN Backbone Investigation 79
8.1.1 The Nature of the Problem 79
8.1.2 Comparing the Performance of FDDI, DQDB and Metroring as Backbones 80
8.1.2.1 Parameters for the Simulations 80
8.1.2.2 Performing the Simulations 84
8.1.2.3 Results and Conclusions 87

CHAPTER 9: CONCLUSION 90
9.1 Unique Features of the Study 90
9.2 Application of Methods for Solving Problems 90

CHAPTER 10: FURTHER RESEARCH 93

APPENDIX A: THE OPEN SYSTEMS INTERCONNECTION (OSI) REFERENCE MODEL 95

APPENDIX B: THE FIBRE DISTRIBUTED DATA INTERFACE STANDARD (FDDI) 97

APPENDIX C: THE DISTRIBUTED QUEUE DUAL BUS STANDARD (DQDB) 104

APPENDIX D: THE METRORING NETWORK 110

APPENDIX E: THE EXPRESS TRANSFER PROTOCOL (XTP) 118

APPENDIX F: DERIVATIONS OF INDICES USED IN ANALYTICAL COMPARISONS 125

APPENDIX G: CONSTRUCTING CONFIDENCE INTERVALS FOR STEADY-STATE SIMULATION 129

APPENDIX H: SIMULATOR TEST RESULTS 131

APPENDIX I: RESULTS FOR LAN BACKBONE INVESTIGATION 133

REFERENCES 137

LIST OF TABLES

1.1	Comparison of WAN/MAN/LAN characteristics	2
1.2	Factors affecting network performance	7
3.1	Comparison of FDDI, DQDB and Metroring characteristics	16
4.1	Criteria for functional comparison of MANs	21
4.2	Relative effectiveness of functionality criterion	36
5.1	Comparison of absolute maximum throughputs	39
5.2	Raw bit rate available per station	39
5.3	Delays experienced in MANs	41
9.1	Evaluation of resulting MAN comparison methods	97

LIST OF FIGURES

1.1 Relationships between different types of networks 3
3.1 Comparison of protocol layers against OSI model 15
4.1 Correcting link failures in FDDI and DQDB 23
4.2 Correcting link failures in Metroring 24
4.3 Reconfiguration of FDDI and Metroring - Case 1 28
4.4 Reconfiguration of FDDI - Case 2 29
4.5 Reconfiguration of Metroring - Case 2 29
4.6 Expanding a FDDI network 30
4.7 Expanding a Metroring network 31
4.8 Concurrent logical links on a Metroring network 34
6.1 Interpreting estimates and confidence intervals 45
6.2 An abstract depiction of pseudo-dynamic load generation 48
6.3 Components and their types in the simulator 51
6.4 Information flow in the simulator system 54
6.5 Token rotation representation in FDDI 60
6.6 Packet transmission representation in FDDI 60
6.7 Multiple packet transmission representation in FDDI 62
6.8 Priority class threshold times for FDDI's TTR protocol 62
7.1 Physical layer throughput vs offered load (FDDI tests) 69
7.2 Average end-to-end user delay vs offered load (FDDI tests) 70
7.3 Average delays for MANSIM (DQDB tests) 74
7.4 Average access delays vs offered load (DQDB tests) 75
7.5 Average access delays vs offered load(normalized DQDB tests) 77
8.1 The given CSIR backbone topology 79
8.2 Representating buildings on the backbone in a ring topology 81
8.3 Collision domain representation on a ring topology 82
8.4 Representating buildings on the backbone for Metroring 83
8.5 Collision domain representation on Metroring 84
8.6 User layer throughput vs offered load (CSIR backbone equiv.) 87
8.7 Ave user layer delay vs offered load (CSIR backbone equiv.) 88
B1 Component entities of a FDDI station 97
B2 Layout of fields of a FDDI token 99
B3 Layout of fields of a FDDI information frame 99
B4 Reconfiguration of FDDI to eliminate a failed component 103
C1 Component entities of a DQDB station 104
C2 Structure of a DQDB network 105
C3 DQDB slot structure 106
D1 The component entities of Metroring 110
D2 Interconnected rings made from ring segments in Metroring 111
D3 Information and Management frame composition 112
E1 XTP three packet connection mode handshake 118
E2 XTP packet types and formats 119
F1 Components of a data frame 125
F2 Symbolic representation of FDDI, DQDB and Metroring 127

CHAPTER 1: INTRODUCTION

It is now common to find Local Area Networks, LANs, in use in many organizations throughout the world. A LAN provides computer users with shared access to common resources, such as storage (harddisks) and input/output devices (printers, digital scanners, etc.) [Seifert 88]. It also enables users to exchange information rapidly and easily. LANs generally cover geographical areas the size of a building or campus.

In a similar way, wide area networks, WANs, allow many users to exchange information over much larger distances like those found between cities and countries. WANs make use of existing links, such as Post, Telephone and Telecommunication (PTT) service lines, whereas LANs utilize a network medium intended solely for the LAN. An example of a WAN is the telephone system which effectively spans the whole world. An example of a LAN is computers linked together in a building.

In the early 1980's a new type of network was developed which had the potential to increase network performance considerably: Metropolitan Area Networks (MANs).

As MANs are relatively new, research and development are on-going expanding this area of network technology. New tools and techniques are being developed to manage this new class of network while several network protocols have been proposed to implement MANs. A method is required to determine which of these protocols is best suited to implementing a MAN.

1.1 The Aim and Objectives of the Project

Aim: To develop a method which would enable the significant characteristics of Metropolitan Area Networks (MAN) protocols to be identified. This would enable protocols to be compared relative to one another to determine which MAN protocol is best suited to the requirements of a given network configuration and load.

Objectives:

1) Determine factors affecting performance in MANs.
2) Determine criteria for comparison of MAN protocols' relative suitability for meeting performance requirements stipulated by network design requirements.
3) Derive methods for comparing MAN protocols in terms of the criteria determined from objective (2).

The study of comparison methods would be validated by examining and comparing the Distributed Queue Dual Bus (DQDB), Fibre Distributed Data Interface (FDDI) and Metroring protocols for specific network configurations and loads.

1.2 Background

LANs and WANs can also be distinguished by several other criteria besides physical area coverage and physical link usage:

- data transfer rates,
- type of physical links,
- protocols and
- topologies.

Essentially a WAN provides physical point-to-point connections, while a LAN provides one-to-many physical connections. Technically, this implies routing mechanisms are needed for WAN operation but not for LAN operation. A station on a LAN transfers information by first selecting which of the other stations on the network the data must be sent to. The station then sets the destination address accordingly and transmits the data onto the local network. WANs on the other hand use addresses for two purposes to implement the routing. Addresses are used in WANs to distinguish the destination station from other stations on the network and to select the correct point-to-point link on which to transmit the information to the destination station. As the data reaches one station, the destination address determines if this station is the intended station. If not, the destination address determines the next link to transmit the data on to eventually reach the destination. Sometimes networks are designed so that this destination station resolution requirement is made easy to fulfil eg. in Ethernet every station first reads a passing packet and then determines if the packet is intended for itself.

1.2.1 Different Classes of Networks

LANs can also be linked together to form larger networks resulting in larger communities of users. WANs are often used to achieve this, ie. interconnect several LANs in locations widely separated from each other. However, LANs and WANs were designed to meet different communication requirements. LANs meet a small group of localized user's connection needs, while WANs meet several distributed groups of user's needs.

MANs are a class of network designed to cover a metropolis and combine features of LANs (ie. multiple access and packet switching techniques) and WANs (ie. long distance connections) with fibre optic technology. MANs can allocate bandwidth as required by users, which is something useful for both LAN and WAN services. It is now becoming part of the trend in MAN development to also provide sufficient security measures for individual users as many organisations could end up using the service of just one MAN [Cranfill 90].

Table 1.1 [Smythe 91] shows how the characteristics of a MAN compare with those of LANs and WANs. The main distinguishing factor is geographic size, while the other characteristics give an idea of how the classes of network's compare. It is apparent that in some cases some characteristics will not differ. This is important as it is one of the difficulties encountered in classification of networks.

Issue	WANs	MANs	LANs
Geographic size (km)	1000's	1-100	0-4
No. of nodes	10000's	1-500	1-200
Data rate (bit/sec)	100-100k	1-100M	1-100's M
Probability of bit error	10^{-3}-10^{-6}	$<10^{-9}$	$<10^{-9}$
Delay (msec)	>500	100-100's	1-100

Table 1.1 Comparison of WAN/MAN/LAN characteristics

Figure 1.1 illustrates the relationship between the WAN, MAN and LAN classes of networks by considering reliability and the range of services provided by each class and type of network. The superiority of a type of network's architecture in terms of reliability and range of services is represented by the area of a patch. The network types shown (POTS, PSTN, PABX and ISDN) encompass only part of the three network class' superiority. Eg. WANs are represented as being more superior than any other class or type of network; LANs are observed to encompass only part of MAN service and reliability superiority.

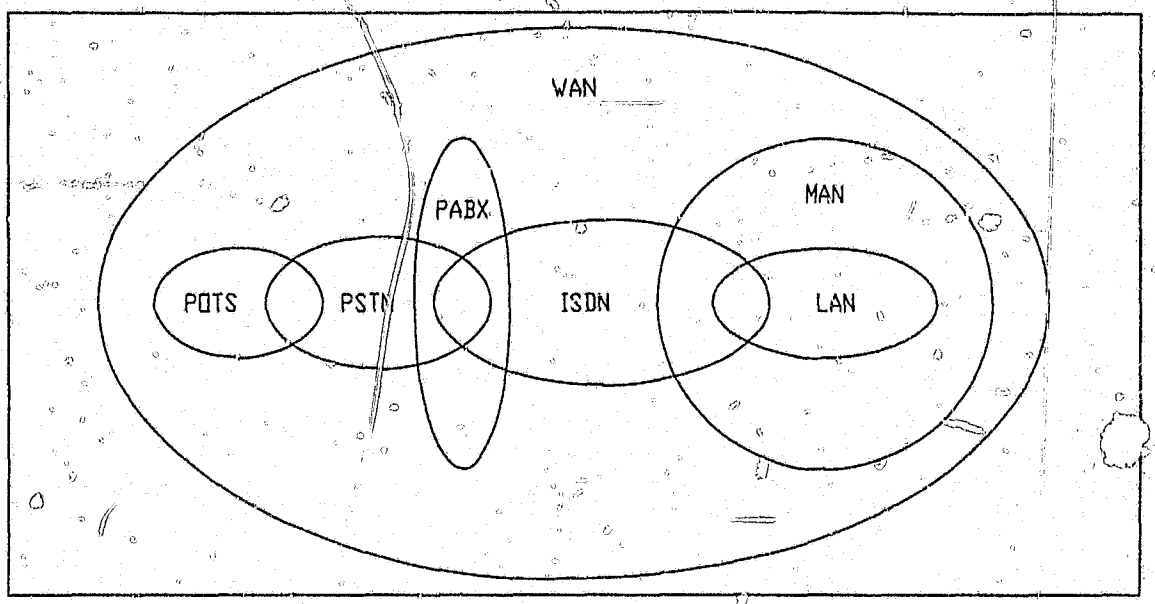


Figure 1.1 Relationships between different types of networks (derived from [Smythe 91])

The types of networks used in this illustration are:

POTS - The 'Plain old telephone service'.

PSTN - The Public Switched Telephone Network more generally describes the telephone system as a hierarchy of switching exchanges. These switching exchanges allow two telephones to be linked via a dedicated closed circuit. The difference between PSTN and POTS is that PSTN provides a better service than POTS. For example, PSTN allows digital communications whereas POTS does not.

PABX - Private Automatic Branch Exchanges are mini PSTNs allowing a local environment (eg. a company) exclusive communication service.

ISDN - The Integrated Services Digital Network is the standard being developed for a single WAN, which will allow all types of networks and their services to interconnect.

In distinguishing different network classes quantitatively, it is advantageous to interpret table 1.1 as the trends for classification of networks. However, in individual protocol specifications, the type or class of network the protocol characterizes is often explicitly stated.

1.2.2 Example Protocols Being Compared by Means of the Method

As with LANs and WANs, several protocols have been devised for MANs. Organisations such as the Institute for Electronic and Electrical Engineers, IEEE, and the International Standards Organisation, ISO, regulate the standards for each class of network (LAN, MAN and WAN). They also specify protocols devised to meet each network class' requirements. Currently the IEEE 802.6 subcommittee is responsible for the development of MAN standards. This subcommittee's intent is to provide a city-wide network protocol [Mollenauer 88].

The three protocols which are used to validate and illustrate the application of the method are:

1.2.2.1 The Distributed Queue Dual Bus Protocol, DQDB

Originally the Queued Packet and Synchronous Exchange (QPSX) was adopted by IEEE 802.6 as the protocol to meet MAN requirements. It has evolved to become known as the Distributed Queue Dual Bus (DQDB) protocol [Clemens 92]. DQDB is configured as a dual optical-bus with a special distributed queuing mechanism [Smythe 91]. As DQDB's first trials commenced in 1990 [Neef 92], and the first intercontinental trials were performed in 1992 [Clemens 92], it is a relatively new protocol. A summary of the protocol is found in appendix C. The particular version referred to in this study of methods is Version D.10 of IEEE 802.6 DQDB - MAN.

1.2.2.2 The Fibre Distributed Data Interface, FDDI

FDDI is constructed from two contra-directional rings and uses a token passing access method. FDDI is a well established standard as it is presently being used by many organizations around the world [Midkiff 92]. The protocol standard, documents ANSI/IEEE 802.5 - 1985, is summarised in appendix B [Ross 89].

Although it is basically an extension of Token-Ring running at 100Mbps instead of 4 or 16Mbps, it is capable of meeting the requirements of a MAN. However, FDDI is classified as a LAN protocol by ISO and IEEE [Halsall 88]. It was not meant to be a MAN protocol, but for the purpose of this study it is considered one, as its circumference can be as large as 100km and it operates at 100 Mbps using fibre optic cable. Cranfill [Cranfill 90] also identified FDDI as a means of implementing a MAN. This ambiguity of FDDI's intended class illustrates how it can be difficult to classify networks.

1.2.2.3 The Metroring Network

Metroring has been proposed to meet the same MAN requirements by linking a maximum of 16777216 stations together in a mesh like topology. The network mesh is formed in a special manner:

- Stations are physically linked in series to form a *segment*.
- Segments may then only be linked in such a manner that interconnected *rings* of stations result.

Metroring uses full-duplex point-to-point connections between adjacent stations, like an internet, to allow adjacent stations to communicate concurrently. The first version specifications are not complete yet, but the necessary information available from Harris [Harris 92] and Tjirkali [Tjirkali 92], is summarized in appendix D.

1.2.3 Application of MANs in Networking

MANs can be used to construct networks in various ways:

1.2.3.1 Backbones

A MAN gateway may be used to connect several LANs together instead of using several bridges or routers to form an internetwork. In this case the MAN acts as a backbone for linking the LANs together. The backbone configuration eliminates the problems encountered with large bridged internetworks, such as:

(1) routing table maintenance in routing devices on an internet and
(2) store and forward delays experienced at every routing device the data passes through (hops on the internet).

1.2.2.1 The Distributed Queue Dual Bus Protocol, DQDB

Originally the Queued Packet and Synchronous Exchange (QPSX) was adopted by IEEE 802.6 as the protocol to meet MAN requirements. It has evolved to become known as the Distributed Queue Dual Bus (DQDB) protocol [Clemens 92]. DQDB is configured as a dual optical-bus with a special distributed queuing mechanism [Smythe 91]. As DQDB's first trials commenced in 1990 [Neef 92], and the first intercontinental trials were performed in 1992 [Clemens 92], it is a relatively new protocol. A summary of the protocol is found in appendix C. The particular version referred to in this study of methods is Version D.10 of IEEE 802.6 DQDB - MAN.

1.2.2.2 The Fibre Distributed Data Interface, FDDI

FDDI is constructed from two contra-directional rings and uses a token passing access method. FDDI is a well established standard as it is presently being used by many organizations around the world [Midkiff 92]. The protocol standard, documents ANSI/IEEE 802.5 - 1985, is summarised in appendix B [Ross 89].

Although it is basically an extension of Token-Ring running at 100Mbps instead of 4 or 16Mbps, it is capable of meeting the requirements of a MAN. However, FDDI is classified as a LAN protocol by ISO and IEEE [Halsall 88]. It was not meant to be a MAN protocol, but for the purpose of this study it is considered one, as its circumference can be as large as 100km and it operates at 100 Mbps using fibre optic cable. Cranfill [Cranfill 90] also identified FDDI as a means of implementing a MAN. This ambiguity of FDDI's intended class illustrates how it can be difficult to classify networks.

1.2.2.3 The Metroring Network

Metroring has been proposed to meet the same MAN requirements by linking a maximum of 16777216 stations together in a mesh like topology. The network mesh is formed in a special manner:

- Stations are physically linked in series to form a *segment*.
- Segments may then only be linked in such a manner that interconnected *rings* of stations result.

Metroring uses full-duplex point-to-point connections between adjacent stations, like an internet, to allow adjacent stations to communicate concurrently. The first version specifications are not complete yet, but the necessary information available from Harris [Harris 92] and Tjirkali [Tjirkali 92], is summarized in appendix D.

1.2.3 Application of MANs in Networking

MANs can be used to construct networks in various ways:

1.2.3.1 Backbones

A MAN gateway may be used to connect several LANs together instead of using several bridges or routers to form an internetwork. In this case the MAN acts as a backbone for linking the LANs together. The backbone configuration eliminates the problems encountered with large bridged internetworks, such as:

(1) routing table maintenance in routing devices on an internet and
(2) store and forward delays experienced at every routing device the data passes through (hops on the internet).

1.2.3.2 Bridged MANs

MANs can be bridged together with routing bridges. Initially a network may start off as one MAN which then eventually expands into an internet of MANs. When compared with bridged LANs their main advantage would be the increase in overall speed. The speed increase will be due to the faster operating speed of MAN technology along with the larger physical area coverage of one network.

1.2.3.3 Asynchronous Transfer Mode (ATM) Switching

ATM switches are very high speed switches which are still undergoing development. Due to their technique of construction, they are also known as batcher-banyan switches [Heywood 89]. They will be able to switch packets at rates comparable with that of circuit-switches. The routing mechanism's store and forward delay, found in bridges and routers connected between networks, would effectively be eliminated. These switches would be ideal for MAN interconnection when they become available in about 1995 [Heywood 89].

1.3 Application of the Study of Methods in Network Engineering

The comparison method developed in this project provides a method to determine which type of MAN protocol is most suitable for meeting a specific set of configuration and loading requirements.

MANs support the services which PTTs supply, like LAN interconnection and digitized voice services. PTTs world-wide are thus spending tremendous effort in developing and implementing MANs [Heywood 89]. Results from this dissertation contribute a tool to aid in the selection process.

The comparison method is not limited to the three example protocols described. Provision is made for other MAN protocols to be modelled for the comparison of protocols. This will allow other MAN protocols to be included in the performance analysis and comparison. The comparison method has the potential of also being applied to LANs and internets (bridged LANs).

1.4 Limitations and Assumptions of the Method

Consideration of available resources for deriving the comparison method lead to several limitations and assumptions as to the scope of the MAN selection method.

1.4.1 Limitations of the Comparison Method

Four important limitations in the MAN selection method are as follows:

1) No implementation dependant factors are taken into account, eg. protocol processing delays. These factors are not always examinable as they cannot always be determined. Reasons for this indeterminability are:
   - Conditions under which protocols are analysed are, normally, highly idealised. Specific implementation factors in protocols, like processing delays, cannot then be considered in protocol assessment.
   - Measurement functions required for evaluating implementation characteristics are not available in commercial protocol software entities.
   - Protocol entity manufacturers do not always provide sufficient information on pro-

tocol entity performance.
- No implementation factors are predictable for protocols that are still undergoing development.

2) Only **asynchronous network traffic's** (as described in section 1.5.1) **effect on performance** is considered. The other form of network traffic is isochronous and has easily predictable effects on performance of networks. However, as an available service, isochronous traffic is considered.

3) Only the **physical layer** and the operation of stations on the network up to the top of the **MAC layer** is considered for protocol performance analysis. Section 3.1 mentions further functionality that is considered, but as far as the selection of performance comparison criteria is concerned only effects of the MAC layer and physical network are considered. Further functionality details become necessary in only a few areas of protocol comparison - see sections 4.2.3, 4.4 and 6.1.3.1.

4) The **steady-state network operation**'s performance is analyzed. Steady-state operation requires that no hard errors (station or link malfunctions) are present and no stations are attached or removed during network operation. There is also no shifting in the levels of the load applied to the network so that performed measurements represent the steady-state behaviour of the network.

1.4.2 Assumptions Made for the Comparison Method

Two assumptions in the MAN selection method are required due to the above limitations:

1) Only the effects on one network are analysed. No external networks are connected to the network under consideration ie. **no inter-networking traffic** is present on the network. A mixture of internetworking and local traffic does not reflect the true nature of the load of the network considered. Traffic local to the network must be used to determine the performance of the protocol.

2) Management functionality implemented via software is not considered. Thus **no logical management** part of the protocol, as defined below in section 1.5.1, is considered. The management functions excluded are
-initialization of network,
-allocation of network resources to users,
-security of data,
-automatic optimizing of configuration for capacity allocation and
-routing overheads.

1.5 Proposed Solution

1.5.1) Determining factors affecting performance in MANs

Table 1.2 lists the key factors distinguishing one protocol from another. It is important to identify such factors, and select the relevant ones in order to compare the protocol characteristics and overall performances from a common perspective. Table 1.2 indicates which distinguishing factors are considered relevant to the comparison method under the assumptions and limitations listed in section 1.4 above.

The several distinct factors, used in defining network protocols are:

Factor	Relevant
Loading of Network Nodes	Yes
Physical Topology	Yes
Configuration of Network Nodes	Yes
Available Services	Yes
Medium Access Method	Yes
Soft Error Recovery Method	Yes
Hard Error Recovery Method	Yes
Station Initialization Method	No
Physical Management of Network	Yes
Logical Management of Network	No
Protocol Structuring	Yes
Internetworking effects	No

Table 1.2 Factors affecting network performance and their relevance to the comparison method

a) Loading of network

Network loading deals with the pattern in which stations place data on the network. The loading pattern or profile is defined for each station in terms of the message lengths and the time of message transmissions.

b) Configuration of network

Configuration defines the exact way in which stations are connected together and the physical link distances between them for an operating network.

c) Physical topology and medium access

These define the architecture of the network in terms of
(1) the valid connection of stations in forming the network and
(2) the mechanism which controls station access to the physical network, called the Medium Access Control (MAC).

d) Available data services

Two types of data services can be provided by data networks:
1) *Asynchronous service*: data packets are transmitted at random points in time. This is fundamental for allowing LANs, with their typically bursty traffic, to communicate. An easy example where this type of service is used is a MAN that interconnects multiple LANs together.
2) *Isochronous service*: data (packets or bytes) are transmitted at regular intervals across the network, meaning a portion of the network bandwidth must be reserved. This allows data like voice and video signals to be carried on the network [Mollenauer 89].

e) Error recovery

The method, or procedures, used for hard error (hardware and protocol entity software failures) and soft error (bit and packet errors) handling determines how the protocol recovers from errors. Frame acknowledgement procedures between stations can also be used in aiding error recovery.

f) Management

Management of the network can take place at two levels: The physical organisation of the network and the management of users and resources on the network. We distinguish between these two "levels" of management as, respectively, physical management and logical management.

Physical management will refer to the physical planning, set-up, maintenance and future growth of the network. Here we consider the ease with which each of these tasks can be performed for different types of MAN topologies.

Logical management of the network (eg. controlling user access to network resources) is normally performed using special software. The functionality of this part of the network management is largely dependant on the implementation of the management software. As this is a high level function which is implementation dependant, it is not considered in the comparison method.

g) Protocol structuring

This refers to the degree to which protocols conform to the ISO reference model, the variation used by the IEEE (using MAC and LLC sublayers) or some unique protocol architecture.

1.5.2) Determining criteria for comparison of MAN protocol performance

Example indices for evaluating overall network performance for comparison of protocols are:

- *User data throughput*
  The transmission rate of useful data between processes. Typically this would be measured at the top of the MAC layer.
- *Line Utilization*
  This is the fraction of the total time data is carried by the medium.
- *MAC Data Frames per User Information Frame*
  The ratio of transmitted frames to the number of user data frames indicates how efficiently a protocol utilizes frames transmitted on the network.
- *Error recovery times*
  When a frame is lost or damaged, the protocol must detect the error and recover from it. This recovery time is an overhead.
- *End-to-End delay*
  This is the delay encountered between the time from starting to transmit a packet to the time when the packet is correctly received. This may include the time it takes an acknowledgment packet to return to the sender of the data packet.
- *Line efficiency*
  This index designates the ratio of the useful traffic to the total traffic on a link. The useful traffic refers to the amount of data sent by the user to the MAC layer for transmission to another station.

Proper consideration of indices necessary for comparing protocol performances is found in section 3.3 and 3.4.

1.5.3) The derivation of the methods for comparing MAN protocols

Using the criteria from section 1.5.2 the MAN protocols are compared against the background of the problem requirements from three different perspectives. These perspectives allow one to assess protocols at different levels of complexity. They form a hierarchy of methods for comparing protocols against several requirements or criteria that the user of the MAN comparison method will choose.

The first most fundamental way of comparing protocols is in terms of the **functionality** each protocol provides. The most elaborate way is when the eventual topology and load is taken into account and performance is assessed by means of **simulation**.

1.5.3.1) Functional Comparison

Here the basic requirements of the problem are taken into account to choose a MAN. The qualitative requirements of services, management and reliability of the protocol are examined without considering the required performance of the network (a quantitative requirement).

1.5.3.2) Analytical Comparison

Here performances are compared from a common perspective ie. a standard configuration is defined from which the indices for comparison are derived. They are derived by considering a special representation of the logical MAN topologies to be analysed and compared. The load and topology of the network is not considered as the emphasis is on the combined behaviour of the physical structure of the network and the medium access protocol working together.

1.5.3.3) Simulation Aided Comparison

As the effects of configuration and loading on a network are difficult to theoretically predict, simulation is used to analyse performance of the network when these two factors are taken into account. This method of comparison also allows the MAN comparison method user to custom design a network and find the best configuration and load distribution for the stations.

These three methods of MAN protocol comparison can then be used individually or collectively for comparing protocols. The user of these methods has discretion as to which method (or methods) of protocol comparison best suites the network problem requirements at hand.

Each protocol that needs to be compared against others will need to be evaluated in terms of the criteria distinguishing the method of comparison. The evaluated protocol can then be compared against others in terms of the results of the evaluation criteria. The protocol evaluation procedures take on different forms according to each method of comparison.

1.6 Validating the Results and Methods Obtained

Results from operating networks are not available for comparison purposes. The results of the research will be validated from two perspectives:

1.6.1 Validate the applicability of the comparison method

The study of comparison methods is applied to the DQDB, FDDI and Metroring protocols. These particular protocols illustrate the method's applicability to

- an existing protocol, FDDI,
- a protocol well into its development phase, DQDB and
- a protocol early in its development phase, Metroring.

The ease of applicability of the procedure of comparison along with its relevance to the comparison of these three protocols will illustrate the validity of the comparison method.

1.6.2 Validation of Simulator Protocol Models

A FDDI simulator is accessible. Results from parameter driven simulations of FDDI will be available. Comparison with results from papers is made for FDDI and DQDB only. The first Metroring simulation results will only be available at a later stage of 1993.

For FDDI:
- LANNET II.5 provides a FDDI simulation model (see section 7.2).
- Johnson [Johnson 88] gives many general results of FDDI simulation - frame delays, queue lengths, throughput.

For DQDB:
- Conti et al [Conti et al 91a] shows delays, packet loss and throughput against node positions.

1.7 Overview of Report Structure

Chapter 2 outlines the background material to this dissertation.

Chapter 3 presents the preliminary requirements before comparing MAN protocols. The problems addressed are the functional differences in example protocols, the classes of protocol characteristics required for comparisons and criteria used for comparing protocols.

Chapter 4 details the functional comparison method.

Chapter 5 explains the analytical comparison method.

Chapter 6 and 7 contain a summary of the development of the simulator. Initial requirements through to the validity of the protocol software models are presented.

Chapter 8 describes the application of the study of comparison methods to an actual network problem. The problem is outlined and the simulation method of comparison is fully illustrated here.

Chapter 9 concludes the report.

Appendix A through E present the protocols referred to in this dissertation. Appendix A summarizes the OSI 7 layer communication model. Appendices B through D describe only the details required of the three example protocols referred to throughout the dissertation.

Appendix F and G detail mathematical procedures used in the comparison method. Appendix F gives the definitions and derivation procedures for analytically comparing protocols. Appendix G summarizes a sequential procedure used for controlling the length of simulation for the constructed simulator.

Appendix H and I tabularize the results from the protocol model tests and the case study analysis respectively.

CHAPTER 2: SURVEY OF CURRENT DEVEL

The methodologies, methods and procedures reported to date in l discussed here. Current network simulators were also studied to f requirements of the simulation part of the study of comparison met

2.1 Investigations into the Protocols Used to Illust Method

FDDI is a well defined protocol in use today. It has been subject studies. Tangemann and Sauer [Tangemann et al 91], Huber, Sauer al 88], and Johnson [Johnson 88] give details of performance of the authors draw the following conclusions:

- FDDI follows the ideal utilization curve (as shown in figure 7.1).
- Average frame delay is low until the ring nears saturation
- Average transmission queue lengths remain small until the ring ne
- Under light and medium load both throughput and waiting times of the target token rotation time (TTRT).

An analytical model of a FDDI station was proposed by LaMaire [LaM could provide analytically derived delay results for design of FDDI cor These analytical results could then be used in place of simulation res stochastic M/G/1 vacation model of a single FDDI station.

The model predicted how the mean waiting time of frames in the static on the limit on the number of frames that could be transmitted durin station However, several assumptions in this dissertation, like frames a and the user inserts frames at a Poisson frame arrival rate, limit the appl

Conti et al [Conti et al 91a] examines the behaviour of the asynchronol performance and fairness, with and without the Bandwidth Balancing M investigated. The BWB mechanism controls fairness of station access c conclude that disabling the BWB mechanism results in unfairness no quality of service when the network is in underload. In underload the than the medium capacity. However, it does cause unpredictable network is overloaded.

- Security of data for service users
- Network management of the MAN for service provider and user
- Compatibility with existing networks
- Transition from current to future technology for MANs

Better defined technical requirements would ease the implementation of the network.

Cranfill took the future world-wide MAN, the broadband integrated services digital network - BISDN, to be an eventual goal for all MAN users. Thus no rigourous well defined approach was given for an independent MAN based communication system.

With far more applicability to the required MAN comparison method, Capurro et al [Capurro et al 91] maintain that correctly executing a broadband network project is not a simple task. They realise that a **rigourous and systematic planning methodology** is necessary. A planning methodology for public communication structures of large dimensions was outlined for MANs. The factors that were taken into account to identify choices and steps for the plan were:

- The **users** and the evolution of their needs in the region of interest.
- The **services required** by the users in the considered region.
- **Technical and environmental factors and constraints** in the geographical area of interest.
- The **technology and communication systems requirements** for the region.

From these factors they developed a rigourous methodology for planning MANs. The methodology covered a wide scope of network planning and development that does not apply to the intended research for this MAN comparison dissertation. Since their planning methodology did include network performance evaluation components, the four above factors were considered in this MAN comparison method. The analysis of the requirements for a MAN, as well as the required topology were outlined by Capurro et al. These two steps in solving MAN protocol selection problems are left to the user for this study of MAN comparison methods as they are very general and diverse topics beyond the scope of this project.

2.3 Trends in Network Simulators

Conti et al [Conti et al 91a] simulates the asynchronous part of DQDB with one priority class. They investigate the protocol's fairness of bandwidth allocation to nodes with and without the Bandwidth Balancing Mechanism enabled. The simulator requirements identified here for the DQDB investigation were:

- The aspects of the DQDB MAC protocol relevant to the study, ie. modelling of queues, event synchronization and resource contention.
- A flexible model allowing the incorporation of the new DQDB protocol modifications as DQDB evolves.
- Statistical reliability of performance measurements.

Instead of developing a new package, Conti et al created a model using the Research Queueing, RESQ, package. The following assumptions were made in the creation of this simulator:

- The interarrival time between segments is exponentially distributed.
- Each node consists of (1) a DQDB MAC, (2) a local node queue and (3) a traffic generator.
- A user packet coincides with a segment, ie. no fragmentation is required.

- When the local node queue is full, new packets arriving are discarded.
- No overhead from the control parts of segments is taken into account.
- Traffic for this network and external networks is modelled at the user layer.

These assumptions aid in establishing a context on which to base the simulator for this dissertation. Not all of the assumptions for the simulator would apply to the MAN simulator as only behavioural characteristics specific to DQDB was studied. However, the procedure used in this DQDB fairness study highlighted the need to identify metrics for studying the investigated phenomenon for the research, viz, fairness metrics for the DQDB fairness study and performance metrics applying to the comparison of MANs.

A performance study of the operation of FDDI was carried out by Johnson [Johnson 88] using a simulator. Johnson's simulator, called the Local Area Network Simulator (LANES), was used for analysing standard performance measures such as delay, channel utilization and transmission queue length as well as characteristics of the protocol's ring behaviour. The assumptions made in this simulation study were:

- A homogeneous network is studied with each node generating frames at the same mean arrival rate.
- The interarrival times for each node are exponentially distributed.
- As soon as frames are generated at each node, they are placed into the transmission queue.

These assumptions illustrated further considerations for the simulator used for comparing MAN protocol performance.

The most useful simulator studied, that illustrated many features required by a simulator, was documented by Frost et al [Frost et al 90]. This simulator, the General LAN Analysis System (GLAS), took into account

- LAN sensitivity to topology,
- traffic assumptions and
- processing delays.

It allowed for the direct comparison of LAN protocols under a common set of modelling assumptions. The Logical Link Control Protocol (LLC), the Token Ring, Token Bus, CSMA/CD and FDDI protocols could be studied under a variety of conditions by using GLAS. The features of GLAS were:

- Each of the MAC protocols could be analysed individually or in combination with the Logical Link Control protocol.
- The priority mechanisms of the MAC protocols were also incorporated into the models.
- Any topology could be used with all link distances being of arbitrary value.
- The traffic load was modelled for each node. The specification of interarrival time and message length distributions would characterise each node's load.
- Frame processing latencies are modelled allowing effects of specific protocol implementations to be included.
- A menu driven user friendly interface is used for setting all the parameters for the simulations.
- Analytical and simulation techniques are used for respectively predicting approximate and precise performances of networks.

CHAPTER 3: APPROACH TO COMPARING THE PERFORMANCE OF MANS

3.1 Bridging Functional Differences in the Example Protocols

Metroring, DQDB and FDDI standards are designed to meet varying levels of functionality. This is represented in terms of the ISO reference model (refer to appendix A) in figure 3.1. FDDI and DQDB specify the physical and MAC (medium access control) layers only, while Metroring specifies all lower layers up to the bottom of the Transport layer representing the end-to-end connection management function.

To compare the three protocols correctly, this functionality difference needs to be augmented. The functionality difference can be met by using other protocol layers above the LLC layer in FDDI and DQDB. The express transfer protocol (XTP) is introduced above FDDI and DQDB to allow proper comparison of the three protocols.

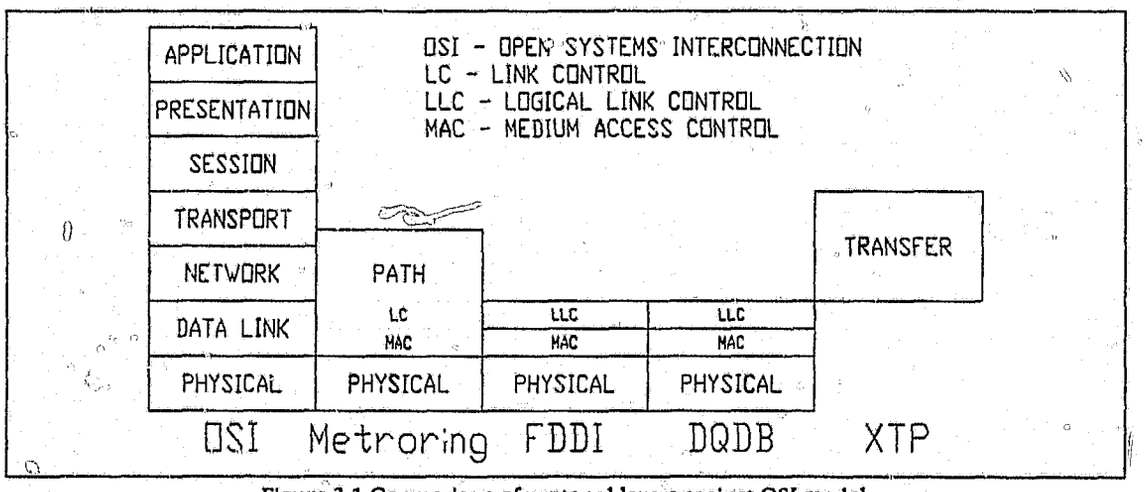


Figure 3.1 Comparison of protocol layers against OSI model

XTP is a real-time transfer layer protocol. The transfer layer consists of the network and transport layers of the OSI model. As the full transport layer does not need to be represented to make the FDDI and DQDB protocol stacks equivalent to the Metroring protocol stack, unnecessary functions, such as fragmentation and error control, would be introduced by XTP. These unnecessary functions of XTP will simply be ignored for comparisons of the three example protocols.

Only when information on the network and link control layers' characteristics is necessary in a comparison, will XTP be referred to for FDDI and DQDB. An overview of XTP and its functions relevant to this dissertation is given in Appendix E.

3.2 Comparing the characteristics of network protocols

The characteristics of network protocols can be classified into three general categories:
- physical attributes eg. cable type and maximum number of nodes on a network,
- network protocol reliability eg. bit error rate and topology strength and
- network protocol performance eg. frame delays and maximum throughput.

Distinguishing between these categories is important when selecting criteria for comparing networks. In this dissertation MAN performance, in particular, is examined. We need to highlight which category of protocol characteristics needs to be analysed and which categories of protocol characteristics need to be taken as independent variables in the analysis. To explain which categories of characteristics act as the independent variables, table 3.1 sets out a physical attributes comparison of the three example protocols.

ATTRIBUTE	FIBRE DISTRIBUTED DATA INTERFACE	DISTRIBUTED QUEUE DUAL BUS	Metroring
Medium	Optical Fibre	Optical Fibre	Optical Fibre
Physical Topology	Dual Counter-Rotating Rings	Two contra-directional buses	Full duplex point to point links
Maximum Number of Nodes	500	65534	16777216
Maximum Network Extent	100 km circumference	50 km bus length	200 km circumference
Encoding method	4B/5B symbol Manchester encoded	Manchester encoded	4B/5B symbol Manchester encoded
Medium Bit Transfer Rate	100 Mbps	150 Mbps x 2 busses	100 Mbps per link
Station Interface Latency	10 bits (2 symbols)	<1 bit (no data stored, but electronics delays)	1 bit delay in Fast Forward mode
Data Unit & Size	Frame of Maximum 4500 octets	Slot of fixed 53 octets	Frame of 4096 octets
Logical Topology	Ring	Bus	Rings
Media Access Method	Permission Token	Slot Requests on the reverse bus and Slot Utilization on the forward bus	Autonomous link control
Priority Control	Multiple Token Rotation Timers in the Timed Token Rotation Protocol	Multiple priority request bits	Multiple priority request bits and queue holding times
Fairness Mechanism	Timed Token Rotation Protocol	Estimation of System Activity via Slot Request and Free Slot counters, and a Bandwidth Balancing Mechanism	N/A

Table 3.1 Comparison of FDDI, DQDB and Metroring characteristics

The criteria are based on the attributes inherent in each protocol, ie. no special analysis is needed for the physical attributes to be compared. Some strengths are immediately obvious when looking at data transfer rates, operating distances, number of nodes and similar criteria. But subtle complications can arise as these limits are attained, for example:

- Physically large networks imply long frame propagation delays.
- Maintaining user information and routing information for over 60000 nodes in DQDB and Metroring is impractical for operating networks.
- When the cumulative load provided by the station users of the network becomes larger than the network's capacity, contention between the stations for use of the medium can arise.

Thus the implications behind these physical attributes need to be addressed. This is where protocol reliability and performance come in. Anyone wishing to choose a network protocol for solving a design problem will need to examine criteria relating to reliability and performance under the requirements of the design problem. The physical attributes of the protocols would be independent variables for the analysis of reliability and performance. The reliability characteristics of network protocols are taken into account in the functional method of comparing MAN protocols. The physical characteristics are independent factors used in determining the performances in the analytical and simulation methods of protocol comparison. Variable attributes like loading and network configuration would also be independent factors in the case of the simulation analysis.

3.3 Criteria used for comparing network reliability and performance

Section 1.5.1 gives a description of the factors affecting a network's performance. These factors are used to identify some indices for comparing MANs as in section 1.5.2. The criteria and indices that can be used for comparing MANs are then selected by considering these factors. These criteria and indices are grouped into three areas:

3.3.1 Protocol Definition Criteria

Protocol Structuring and Conformance addresses the issue of whether the protocols for a network are conformant to the ISO reference model, the variation used by the IEEE (using MAC and LLC sublayers) or some unique protocol layer architecture employed. Large structural variation from a common standard can imply poor compatibility with present or future protocol layers that would lie above or below the protocol layer under consideration.

The number of logical functions a protocol requires for its operation can be an index for **protocol complexity**. Protocol complexity can then be used as a measure for predicting the relative costs of different protocols. The costs would increase as the functionality (the number of logical functions) of the protocol increases. The protocol complexity could also be used to predict the operating speed of the protocol's implementations. The operating speed will generally decrease as the number of functions in the protocol increases.

New protocols do not always meet their initial requirements with the first implementation. Thus, if a protocol is new, it is liable to undergo development at some stage in the future. This implies current implementations of the protocol can become obsolete. To avoid this, the **maturity of the standard** should be considered.

3.3.2 Topology Comparison Criteria

These criteria deal with the physical network and measures for comparing networks in terms of structuring and reliability.

Medium utilization gives a measure of how well a protocol's data encoding scheme uses the capacity of the medium. In other words it indicates how much of the available medium bandwidth is used by the transmitter/ receiver when data is carried over the medium.

Medium reliability qualifies the resilience of the medium and physical operation of the network. The resilience to physical breaks in the medium is analysed for the protocol's operation. If one link breaks, the extent of corrective action can vary depending on the type of medium and method of operation used.

The medium reliability is also very dependent on the **topology strength**. This is a measure that examines a protocol's topology strength. It is used for determining which protocol topology is more reliable than others. The medium reliability is different from topology strength in that it also takes the protocol error recovery mechanism into account.

3.3.3 Network Performance Criteria

Although performance of a network is very dependant on both the network load from the application(s) being run and the configuration of the stations, performance criteria can also be compared under ideal conditions. Although the ideal conditions have little bearing on the practicality of the protocol under varying conditions, they do provide an adequate means of analytically comparing different types of network protocols. Analytical comparison of MAN protocol performance is presented in chapter 5 where idealized performance indices are derived.

Data throughput is the rate of data transfer between peer communicating entities ie. stations on a network. This can be measured at various points in a protocol stack, eg. Physical Layer, MAC Layer and Application Layer. Throughput can also be measured in bits per second (bps) or packets per second (pps). This data throughput is related to the **protocol efficiency** for each layer in a protocol stack. Protocol efficiency indicates how well the protocol transfers data while taking the effects of frame overheads into account. Typical overheads considered are header and trailer lengths. The overhead data ratio (length wise) to the data being transmitted gives the protocol efficiency. The more efficient the protocol is, the less the throughput is affected.

Every packet in a network experiences **frame delays** between its source and destination stations. This delay may be due to:

- **transmission delays** - the time taken to place a whole packet onto the network medium,
- **propogation delays** - the time taken for the end point of a packet to move from its source station to its destination stations,
- **protocol processing delays** - the time taken up by protocol software processing or
- **medium contention** - several stations trying to take control of the medium at the same time.

Generally frame delay degrades performance by expending time that could be used in communicating between two stations.

Delays can be measured relative to various paths in a network. A path may cover:

In section 3.2 a basic comparison of physical attributes is shown. The criteria used here are not difficult to select because they are normally listed in a protocol's specification document. The methods of comparison to be fully defined later, as introduced in section 1.5.3, also need criteria for means of comparing protocols. The attributes from section 3.3 will apply to these categories. The relevant criteria and attributes were identified by taking the limitations of the MAN comparison method, of section 1.4, into account in order to limit the scope of criteria used in the comparison of protocols.

3.4.1) Functional Comparison Criteria

The functional requirements dictated by a network design problem can lead to a choice of the best MAN protocol to meet the requirements quickly. Good **topology strength** and **medium reliability**, which are interpreted as functional requirements, are often specified in the set of requirements for a network. Furthermore, functionality of a protocol implies other factors come into play:

(1) the types of **data services provided**,
(2) data **error handling** abilities and
(3) **manageability** of the physical network.

Although data error handling abilities are present in all protocols, some protocols do handle errors better than others. This quality or efficiency of error handling can be used for comparison purposes. Management of a network is a very general criterion, but was elaborated on in section 1.4.

3.4.2) Analytical Comparison Criteria

Here, the load and topology of the network are not considered in assessing MAN protocol performance. Instead the nature of the physical network and the medium access protocol operation are taken as the factors affecting performance. General criteria of throughput and delays are used for assessing network performance as they are two criteria that apply to all medium-access-controlled digital networks. These two criteria provide the essential information in communication systems as to

(1) the time to send information to its destination and
(2) the rate at which information can be sent across the network.

The specific indices that are analysed without taking a load or network configuration into account are:

maximum throughput,
bandwidth allocation per station and
packet transmission delays.

These measures are idealized due to assumptions as to the structure of the network (chapter 5).

3.4.3) Simulation Aided Comparison Criteria

The effects of configuration and loading on a network are taken into account here. Again **resulting throughput** and **delays** will relatively determine which MAN standard best fulfils a design requirement. The variable factors of applied load and network configuration will need to be specified entirely by the user.

CHAPTER 4: FUNCTIONAL COMPARISON OF MANS

The type of MAN that meets the set of functional requirements for the problem specifications, is determined by a functional comparison of the man protocols under consideration. This final comparison method allows the user to determine which MAN protocols meet the functional requirements for a data network's operation. The question of MAN protocol functionality is addressed before performance of the specific network under consideration. This can result in a considerable portion of further numerical analysis (analytic and simulation comparisons) being excluded if MAN protocols unsuitable to the application are eliminated with functional comparisons.

Protocol analysis is subject to MAN protocol definitions. This analysis is subjective as protocol rules need to be examined and criteria for comparing MANs be identified from the protocol rules. The functional characteristics of MAN protocols are then compared. Section 3.4.1 describes the protocol description criteria that are used in this method of MAN comparison.

The criteria a network designer takes into account when a specific problem is to be solved are listed in table 4.1.

Availability of Isochronous and Asynchronous Services
Reliability of the Physical Topology
Reliability of Soft and Hard Error Recovery
Ease of Planning and Installation
Ease of Physical Network Reconfiguration
Ease of Expansibility of Network
Effectiveness of Fairness Control
Effectiveness of Flow Control
Capacity Planning Ability

Table 4.1 Criteria for functional comparison of MANs

Two of the criteria not directly addressed are cost and ease of operation. They are regarded as too subjective for this analysis because:

- When looking at costs, the cheapest available network will normally be chosen. The long term problems and the meeting of important functional specifications will then, not take priority in the decision process. Cost would then be the most important criterion on the list and it would outweigh many important functional factors. When important design issues are overshadowed by the consideration of initial costs of the network, future needs and changes in the required network's operation can easily result in costs far greater than initially anticipated. The design issue behind long term costs of a network is rather addressed ie. the manageability of the physical network.
- The ease of a network's operation is a very implementation dependant issue, as it depends on management software functionality. As stated in the introduction, implementation issues are not considered in the study of MAN comparison methods.

It is important to differentiate between the principle of a criterion of MAN functionality and its application to protocols considered. The principle of a criterion applied to the three example protocols considered, to illustrate the application of the principles behind the criteria. A comparison of the three example protocols is then compiled to show the validity of this MAN comparison method.

4.1 Available Services

The two types of data services considered here, asynchronous and isochronous, were explained in section 1.5.1.

Principle

Each particular protocol needs to be studied to see if it can meet each service requirement. In cases where several protocols do meet one or both of these requirements, each protocol needs to be assessed as to how accurately or efficiently it meets the service criterion.

Application

Asynchronous service is efficiently provided by all three protocols but isochronous service is not provided effectively by each protocol.

FDDI

FDDI stations are allocated isochronous bandwidth, which they use for purely isochronous data every time they receive the token. Only when data transmitting time is still available after isochronous data transmission, does asynchronous data get transmitted.

DQDB

DQDB was explicitly designed with isochronous transmissions in mind. The stations at the heads of the busses transmit slot "shells" for both asynchronous and isochronous data transmission. The heads of the busses allocate the slots used for isochronous transmissions by marking groups of slots by using the Type bit in the Access Control Field.

Metroring

Maximum permissible delays in data transmissions are required for the isochronous data services. Metroring has no way of ensuring this maximum permissible delay. Mechanisms do not exist for this purpose.

FDDI and DQDB reserve portions of their bandwidths for isochronous service purposes. For this function, special bandwidth allocation mechanisms are required in DQDB stations, although only two stations on the network (the heads of the busses) will use the mechanism at a point in time. Special parameters in the protocol software can be controlled to regulate the allocation of this isochronous bandwidth.

If isochronous services are of great importance in the network, DQDB should be chosen. FDDI's isochronous operation is found to be non-ideal when the offered asynchronous load increases to make the combined load reach or exceed the maximum available bandwidth. In one FDDI trial simulation by Johnson [Johnson 88], when the load was 120% of the network capacity, approximately 3,2% of the synchronous frames experienced unacceptable delays. These synchronous frames with unacceptable delays could not be used effectively for isochronous service. Furthermore, FDDI generally produces lower limit delays too large for typical isochronous traffic latencies. Isochronous latencies of about 125μsec may not be exceeded in the case of voice communications.

Even though isochronous data shares the same bandwidth as asynchronous data in the DQDB and FDDI protocols, the asynchronous service's operation is not significantly effected. A fraction of the possible asynchronous data bandwidth is reserved for isochronous data.

4.2 Reliability of the Protocol

Network reliability can be dependant on several factors. Reliability is examined in light of error correction in networks. Three factors are examined relating to reliability:

(1) The handling of physical link failures
(2) The handling of station failures
(3) The handling of frame errors

4.2.1 Topology Strength and Error Recovery

This examines the efficiency with which a protocol utilizes the physical topology and its characteristics.

4.2.1.1 Principle

The efficiency of physical link failure correction is determined for each protocol. This link failure correction efficiency allows comparison as to which protocol's link failure correction method is most effective. The two aspects of this efficiency are examined:

(1) The least changes in configuration a protocol can make to overcome a link failure.
(2) The number of link failures a protocol can withstand.

4.2.1.2 Application

The three example MAN protocol topologies are configured differently making their relative topology strengths distinct:

FDDI and DQDB

FDDI and DQDB are both configured physically as rings. Both of these protocols can handle one link failure as shown in figure 4.1. In FDDI, wrapping of the primary fibre loop onto the secondary standby loop corrects the failure. When DQDB fails, repositioning of the Logical Break with the physical break overcomes the failure. The natural break will coincide with the DQDB logical break implemented by the protocol.

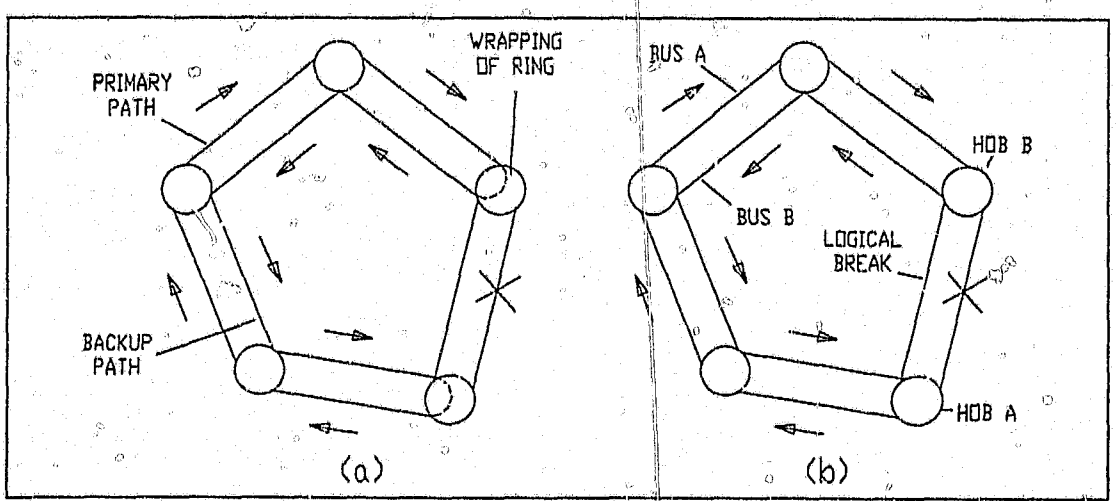


Figure 4.1 Correcting link failures (a) in FDDI and (b) DQDB.

DQDB can, however, be physically configured as a bus, in which case, this principle will not apply.

Metroring

Metroring generally has a more reliable physical configuration than DQDB and FDDI. In the case of a ring topology, one link failure in Metroring can be tolerated, as all links are bidirectional. Bidirectional links allow stations attached to the failed link to use the remaining paths of the network as the route (figure 4.2a). Figure 4.2b shows a slightly more complicated configuration for a Metroring network. Depending on their location, two link failures could be dealt with successfully. If failures 1 and 2 occurred, all stations could still communicate with each other. If failures 2 and 3 occurred, station F would be isolated from the rest of the network.

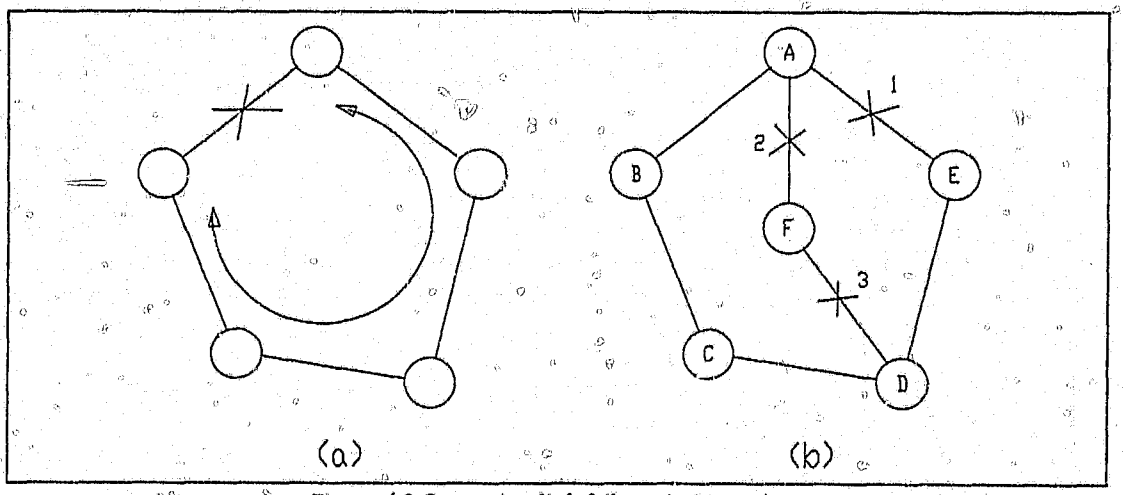


Figure 4.2 Correcting link failures in Metroring.

Thus we see Metroring provides the best resilience to failures in the network. However this resilience is dependent on how well the network's segments are arranged. Generally, the more physical communication paths, or segments, in a Metroring network the better the resilience to failures.

4.2.2 Station Hard Error Recovery

Station errors are those errors that result from actual hardware or software malfunctions. Along with link failures, these types of errors are classed as hard errors - ie. errors occurring in the operation of hardware controlling network behaviour.

4.2.2.1 Principle

Examination of the protocol must take place to determine if these errors are isolated. The effectiveness of the isolation must also be determined for those protocols where station hard error recovery is implemented ie. what degree of possible errors are dealt with by the mechanism(s) of the protocols.

4.2.2.2 Application

All three of the example protocols handle hard errors but all do not do the task equally effectively.

FDDI

The SMT (station management) and CMP (configuration management protocol) entities in FDDI are used in monitoring and controlling errors at the hardware level. The CMP actually controls reconfiguring of connections within the station. If station malfunction is detected by the SMT, the CMP can simply switch the station out of the ring via by-pass connections. If power failures occur, an optical by-pass switch can relay the signals through the station.

DQDB

With DQDB, station malfunctions are also constantly monitored. A faulty station can be isolated either at the physical layer by using an optical by-pass switch, or by simply suppressing any transmissions from the faulty station to the busses. The latter case works as the taps to the busses are logical OR gates. If the fault is with the network interfacing circuitry the optical by-pass switch is used.

Metroring

In Metroring, link and station faults are detected and dealt with at the MAC layer. When hard errors are present, CMP (Configuration Management Protocol) frames are used in redetermining the operating paths in the network. Every station also takes note of failed nodes or links and modifies its routing table accordingly. Metroring stations are thus not by-passed when they malfunction.

All three protocols are capable of handling station failures effectively, but FDDI and DQDB are more effective than Metroring. Metroring leaves gaps in its segments where station failures occur. Multiple station failures present a scenario similar to that of multiple link failures as discussed above in section 4.2.1). In a Metroring, multiple station failures make it possible for one or more stations to be isolated from the rest.

One argument in favour of Metroring is that with DQDB and FDDI only a maximum of three adjacent stations can be switched to the optical by-pass mode before the next active station. More than three consecutive stations in Metroring would be tolerated as their presence will simply be ignored when the failures are detected. As concentrators can by-pass one or several stations at a time in DQDB or FDDI, this argument is not a strong one in favour or Metroring. The use of concentrators allows several inactive stations to be by-passed via one optical by-pass switch. As stations, inactive due to power failures, are normally located on one site, concentrators easily solve the problem. Thus as a rule of thumb, choose DQDB and FDDI over Metroring when it comes to resilience to station failures.

4.2.3 Soft Error Recovery

Soft errors occur when a bit is transmitted, propagated or read incorrectly on the network. This normally leads to the whole packet (or data unit) being interpreted as incorrect. The procedures for handling this situation are present in all protocols and it is not generally considered an issue in selecting a protocol. This usually becomes important when selecting or designing a higher layer protocol, ie. whether the higher layer protocol incorporates packet error checking or not.

The packet sequence error check is not part of the MAC layer operation. As MAN protocols can be designed by incorporating several protocol layers at once (like Metroring), the packet sequence error check is considered.

4.2.3.1 Principle

In a case where soft error checking and recovery is required, the type of checking, packet checksum-error or packet sequence-error checking, must be determined. Some methods of implementing soft error recovery functions can work more efficiently than others. This efficiency of soft error recovery can also become an issue when selecting a MAN for implementing a highly efficient network. This information is obtained from a protocol's specifications.

4.2.3.2 Application

FDDI

At the MAC layer, FDDI uses three fields in the frame status field to monitor detected errors. The transmitting station's MAC responds to these fields. When a packet is corrupted, either one field in the frame status field or a checksum calculation indicates the data corruption to the sending station. The sending station then retransmits this packet. When a token is lost or corrupted, the ring enters a token claiming procedure (see section B2.4.1) to determine which station will retransmit the lost token.

At the XTP level lost packets are handled more effectively in terms of bandwidth used for packet retransmission. Retransmissions of lost packets are not immediate, but when several packets are missing, the receiver notifies the transmitter. The transmitter then only retransmits lost packets. Time-outs at the sending station are also used in this recovery procedure.

DQDB

DQDB has no soft error detection in its MAC layer, and XTP is necessary to handle all soft errors. The checksums and sequence numbers, present in XTP frame common header and trailer fields, are used for detecting soft errors. The checksums and sequence numbers ensure the necessary frame discarding and retransmissions, as discussed for FDDI above, occur.

Metroring

In Metroring, frame errors lead to the packet being discarded. Either a timeout-mechanism or an out-of-sequence frame will then indicate a frame error has occurred at some point in the transmission. An acknowledgement frame is then received by the sender indicating which packets are missing at the receiver. Retransmission of the missing frames then take place.

4.3 Management of the Network

Section 1.5.1 describes the two forms of management applying to networks. Here we consider the ease with which each of the physical management tasks can be performed for a particular MAN protocol.

4.3.1 Planning and Implementations of the Network

Planning and implementation of networks are two separate tasks but as they are part of the same phase of a network's life cycle they are discussed together.

4.3.1.1 Principle

The effort involved in planning and setting up a MAN depends on the complexity of the configuration required.

Complexity, for example, can be defined as a measure of randomness with which stations are located on a geographical area. Quantifying this measurement of complexity as an index is a difficult task. For example, a line or circle of station locations can be viewed as simple (low complexity), while a matrix is complex or relatively simple depending on the matrix dimensions. To generalize and properly define this complexity measure is an immense task, as geometric shape and randomness of station location would need to be considered. Instead, the ease of planning and installation is considered for a basic physical configuration - a ring or bus. The ease of planning and installation is then considered to decrease if the particular protocol allows many methods of variation to the basic physical configuration. The fewer configuration design alternatives for planning the network, the easier it is to plan the network.

4.3.1.2 Application

If the topology required by the problem at hand is the simplest possible, a ring, the amount of effort involved in planning and installation, for each of the three example MAN's considered, would be equal. Considering the possible variations to the ring configuration for each protocol, clarifies which MAN is easier to plan and install.

It is easier to plan for DQDB when configurations become larger (more nodes added). It requires, at the most, the shortest continuous path between all the nodes, as DQDB can be physically configured in a bus topology. FDDI must always be configured in a ring topology. Thus, FDDI requires the shortest length of cabling to incorporate all the nodes in a loop. It is thus easiest to plan for FDDI, as there is no choice as to the topology type (bus, ring or matrix) that must be used.

The ease with which initial configurations of Metroring are planned and implemented cannot be clarified easily. In Metroring's case, let complexity refer to the number of segments the configuration is composed of. The effort involved depends on the complexity of the network layout. When the complexity increases, there are more segment end nodes (SENs) to incorporate into the layout. Also compare figures 4.1 and 4.2 to see that segment placement can be critical in increasing the topology reliability. The placement of these SENs now requires more consideration than any node or link placements in DQDB and FDDI.

The planning cost of the network configuration depends on the ease of planning. Thus, for large networks ease of planning decreases in the order: FDDI, DQDB, Metroring.

Implementing the topologies requires effort and cost directly proportional to the number of nodes and links required in the layout. In Metroring two classes of nodes with different costs will also be an additional factor in determining implementation and planning costs.

4.3.2 Reconfiguration of the Network

Planning a network's future changes, must be considered at the initial MAN protocol selection phase of the network project. The ease with which changes in a network's configuration can be implemented, needs to be addressed.

4.3.2.1 Principle

There is no clear conclusion to be made in this case where ease of reconfiguration is considered. The ease with which a network's topology can be changed, depends on:

(1) the current configuration's complexity and

(2) the reconfiguration flexibility of the protocol topology.

Point (2) indicates that the type of topology (ring, bus, star, etc.) and the protocol rules for connection of links and stations must be identified. The protocol rules for constructing the topology, determine how easily a reconfiguration can be accomplished.

To make the evaluation of the reconfiguration straightforward, the present configuration of a protocol will be ignored. The amount of reconfiguration flexibility afforded by a MAN protocol, will reflect the difficulty with which a choice of the available reconfiguration options can be made. This eliminates complex analysis of configurations.

4.3.2.2 Application

The number of choices allowed for reconfiguring a MAN and the effort involved in the reconfiguration illustrates why the number of reconfiguration alternatives is chosen.

The topologies illustrated in figures 4.3 through 4.5 illustrate the dependence of reconfigurability upon network topology. Two different configurations are considered.

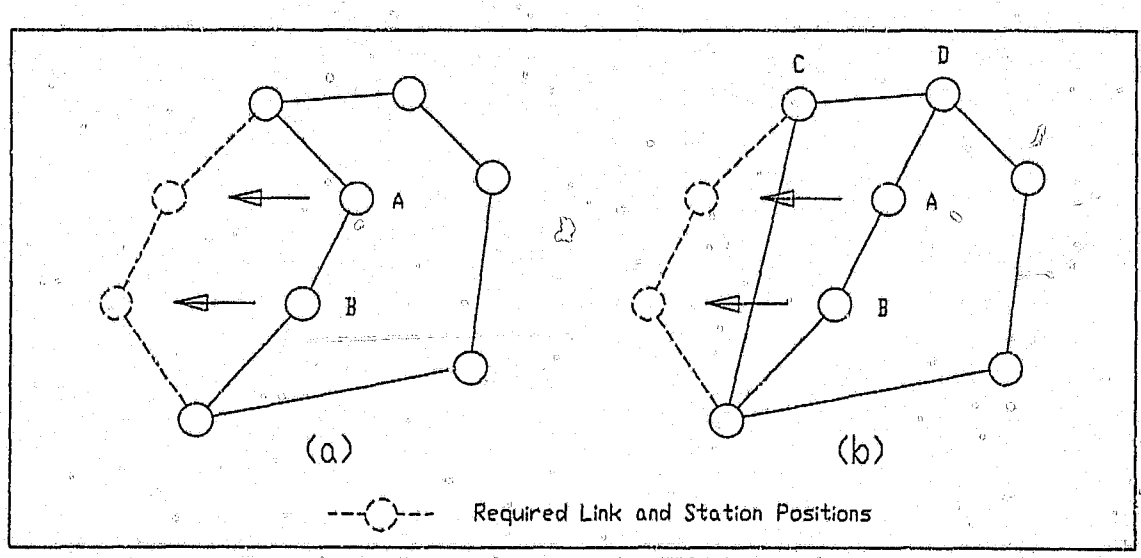


Figure 4.3 Case 1 - reconfiguration of (a) FDDI and (b) Metroring

The first case, as illustrated in figure 4.3, shows how reconfiguration takes place in a FDDI and Metroring network respectively. Here the number of stations and their locations are the same for both cases. Link placement differs for each case. For the repositioning of two nodes (A and B) as indicated, both networks require different adjustments: the FDDI network requires the repositioning of two stations while the Metroring network requires the repositioning of two stations as well as the reconfiguring of stations C and D (as the number of actual ports on each station changes). Thus Metroring requires more effort to reconfigure in case 1.

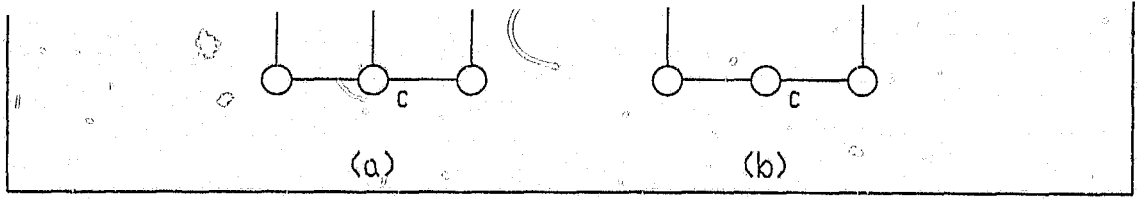


Figure 4.5 Case 2 - reconfiguration of Metroring

In case 2 (figures 4.4 and 4.5) another layout of stations, requires two different link layouts for FDDI and Metroring respectively (configuration a). If the centre station (A) were to be eliminated from the configurations, and the networks reconfigured to take this into account, the changes to stations and links differ in each case. In Metroring, only two links need to be removed with two stations (B and C) reconfiguring their number of available ports. With FDDI, one of two options can be chosen (configurations b or c). The one with least effort only requires a link to be placed from station B to station C. This choice might be the best, but it does result in a loop with the larger ring latency. Ring latency is a factor that should be minimized in FDDI for limiting propagation delays. The other choice, configuration c, has cabling equivalent to the case of the modified Metroring network. The chang[illegible]nvolve repositioning 2 links, removing one link completely and reconfiguring 4 stations ([illegible]E and F). Four stations need to be reconfigured so that input and output ports corresp[illegible]to the direction of frame propagation. However, the ring latency here is the shorter o[illegible]e two reconfiguration options.

As mentioned, each case has different possibilities and the ease of reconfiguration is not easily predicted.

4.3.3 Growth of the Network

4.3.3.1 Principle

As in sections 4.3.1 and 4.3.2, the ease with which a network can be expanded depends on the current configuration as well as the manner in which stations are added. Special characteristics of the physical configuration method of a network topology can make this task easier. However, the more protocol specific options available for expanding a MAN, the

more difficult the choosing of an option for expanding the network. This choosing of an option becomes a problem because the amount of analysis for the alternatives increases as the number of alternatives increase. Thus the expansibility of the network is inversely related to the number of expansion alternatives.

4.3.3.2 Application

The number of choices allowed by the protocol for expanding a MAN and the effort involved illustrates why the number of expansion alternatives is chosen as the index for this criteria. Figure 4.6a shows how FDDI can be used to link the stations in one particular layout. Only two extra links are required with two other links being moved to incorporate the two new stations into the network (b).

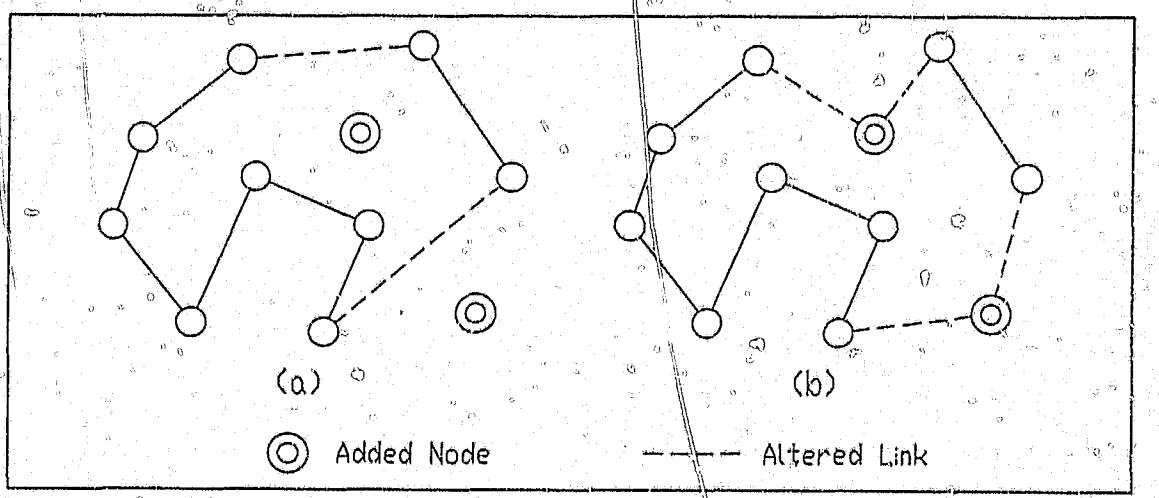


Figure 4.6 Expanding a FDDI network

Metroring could be configured in the same manner to produce a ring to cover the stations. Figure 4.7 shows another configuration taking Metroring's branching ability into account. Two ways of expanding this network are now possible. Firstly, the two stations could just be incorporated into the current links (figure 4.7b) with two links moving and two extra being added. This takes the same effort as with the FDDI case above. Alternatively, a new segment can be added into the network to add these two stations in. Figure 4.7c shows how this can be accomplished. However, three more links are required, and two stations require reconfiguring.

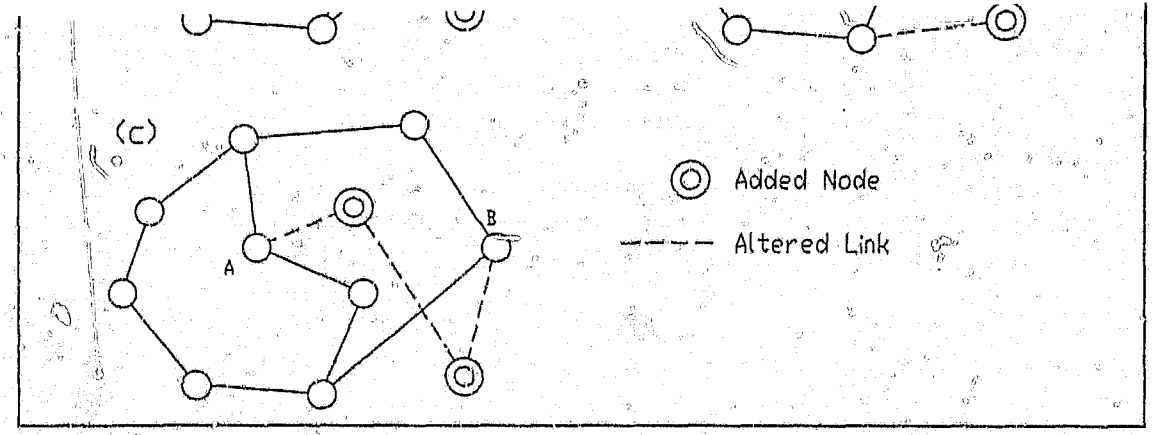


Figure 4.7 Expanding a Metroring network (a) by (b) incorporating nodes into segments and (c) creating a new segment.

All three networks take the same amount of resources to make expansions with minimal effort. However, the number of alternatives for each MAN distinguishes the expansibility better.

4.4 Other Issues

Other issues need to be addressed in order to choose a network. The three issues covered below have been chosen because they are often considered in protocol design. They are also taken into account when choosing protocols for implementing a network:

Fairness: When several nodes concurrently use the network medium, one wishes each to have its share of bandwidth. This partitioning of bandwidth is normally achieved by the MAC protocol. The MAC protocol allows each station equal allotments of time with which it can transmit to the physical network. Equality of the time allotment between stations gives the index of fairness. Thus, unfairness between node bandwidth allocation requires special mechanisms within the MAC protocol to correct the problem.

Flow Control: Flow control is a mechanism in a protocol that enables a sending station to regulate the rate at which it transmits data. Regulation of the data transmission rate is necessary to ensure the destination station is not swamped with data which it cannot process fast enough. This is important in networks that are operated close to their maximum capacity that also need quick efficient information transfer rates.

Capacity Constraints: In heavily loaded networks, capacity allocation can be a critical problem. Capacity allocation is the process of allocating enough channel capacity for any two physically separate stations to communicate across a network. This becomes crucial due to the requirement that the two stations must communicate at a minimum required rate.

When new protocols need assessing using this study of MAN comparison methods, new issues in functionality might be addressed by them. If such issues are raised, they would fall under this section. It must be remembered the functions addressed should be between the Physical and Transport layer of the ISO protocol stack. For new issues to be considered they must also take the limitations and assumptions of section 1.4 into account.

4.4.1 Fairness of the Protocol Operation

Fairness becomes an issue when many nodes in a heavily loaded network need to have a fixed portion of the network's asynchronous bandwidth exclusively for their own use. Unfairness occurs when one station gets more data transferred than another station ie. one station tends to claim more bandwidth to itself than the others. It is important to realise that the loading of each station is the same for fairness to be obvious. The improper design of the bandwidth allocation function of the protocol's MAC layer is where unfair behaviour stems from. Differently loaded stations will obviously place different requirements as to how much bandwidth is required, for servicing their loads. It is not easy to observe unfair operation in these normal conditions where stations are not loaded equally.

4.4.1.1 Principle

In this study of MAN comparison methods, one wishes to establish which protocol is the most fair. Fairness is an issue addressed in most protocols and necessary information can be found in papers discussing the protocol. The extent to which fair operation is achieved by a protocol is the basis for this criteria's analysis.

4.4.1.2 Application

FDDI

FDDI does not need any special mechanisms for fairness to be achieved between its nodes. Fairness is inherent in the access rules of FDDI [Halsall 90]. This applies even with multiple priority class operation.

DQDB

The DQDB protocol has been under much scrutiny with respect to the fairness achieved by it [Banerjee et al 92], [Bisdikian 92], [Phung et al 92], [Conti et al 91a], [Sauer et al 90].

Phung and Breault [Phung et al 92], describe the nature of this unfairness as follows:

> Consider a DQDB network under a persistent traffic pattern (all nodes attempt to transmit continuously) where bandwidth distribution is assumed to be steady state. For DQDB, without any fairness mechanisms incorporated, the bandwidth distribution is represented by:
>
> $$b_1 + b_2 + \ldots + b_N = 1$$
>
> where b_i is the throughput of node i measured over a sampling interval of length T
> N is the number of nodes on the network.
>
> This relationship has no unique solution. It also shows that the final bandwidth distribution depends on the network conditions at the time when the system goes into the overload state.

The BWB mechanism requires every node to let one empty slot pass for every β slots used by the node for data transmission. When the bandwidth balancing mechanism (BWB) is incorporated into the protocol, a new relationship results for the behaviour of each node's throughput:

$$b_i = \frac{\beta}{\beta+1}\left(1 - \sum_{j=1, j\neq i}^{N} b_j\right), \quad i = 1, 2, \ldots, N$$

where β is the BWB modulus.

The effect of this relationship is that some steady state bandwidth distribution is attained for a node, regardless of initial conditions on the network. Another result is that the throughput distribution is independent of the network configuration. Unfortunately, independence of throughput distribution only applies in networks where one priority is used. In the analysis, multiple priority operation resulted in the bandwidth distribution depending on the initial network conditions.

Metroring

In Metroring the question of fair operation has not been fully analysed, but one behaviour problem can be readily observed. The following effect may not be prominently manifest, but over a long period of time its effect will be noted as throughputs of different stations can be accurately measured. Assume logical links between nodes across the network are created and destroyed dynamically over a short period of time.

Figure 4.8 shows three concurrent links on a Metroring network. Assuming the four stations involved on logical links 1 and 2 try and use all bandwidth available, contention arises at station A. Station A, operating at full processing speed, must relay data for both logical links as fast as possible. Thus, station A might cause a bottle-neck situation in the network due to hardware processing rate limitations. If station A does not cause a bottle-neck effect, the physical link used by both connections will limit the rate of information flow on these logical links due to its finite capacity. To simplify the discussion, assume the bandwidth of the physical link is shared 50%-50% between the two logical links. Unfairness will be apparent when a further logical link is present on the network. Logical connection 3 illustrates such a case. No contention is possible as it does not share resources used by other logical connections. It can use 100% of the physical link and station capacities.

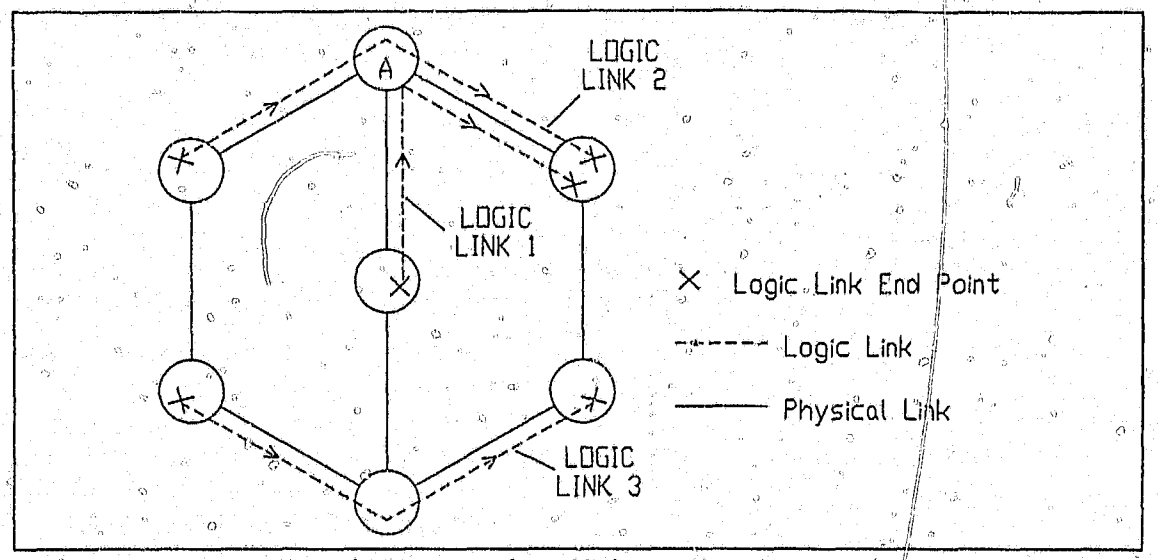


Figure 4.8 Concurrent logical links on a Metroring network

Thus the fairest protocol is FDDI, as its fairness of bandwidth allocation is noted as inherent in the protocol. DQDB directly addresses the problem of fairness and has several mechanisms to eliminate the unfairness (section C2.3). DQDB and FDDI are unlike Metroring, where unfairness of bandwidth allocation can occur due to its' internet-like nature. No mechanisms are used in Metroring to eliminate this possibility.

4.4.2 Protocol Flow Control Mechanisms

4.4.2.1 Principle

The effectiveness of the flow control mechanism is used to compare protocols.

4.4.2.2 Application

FDDI

In a FDDI node, flow control is achieved by a higher layer such as XTP. When a connection is set up by XTP, information flows from source to destination at a rate the destination hardware must handle. To do this, an initial acknowledgement control packet signifies the space available in the receiver's buffers. Then, every time after the receiver has processed a certain amount of incoming information, it sends an acknowledgement control packet to the source to indicate what packets are missing (as part of the error handling). Included in this acknowledgement packet is information indicating what data storage space is available at the receiver. Even when data transfer does not take place at as a high a rate as possible, the sending station does have direct control in this process. It uses SYNC indication in it's frames to 'poll' the destination station's status ie. it forces the receiver to synchronize by sending a control packet back to the sender.

XTP also uses RATE control to allow different station implementations to work according to the slowest receiver's processing rate. The necessary information is also communicated via acknowledgement control packets.

DQDB

DQDB does not implement flow control at the MAC layer. Flow control is also handled by XTP as described for FDDI above.

Metroring

Metroring is implemented in hardware, and information exchange will take place at high rates limited only by the hardware's ability. However, discrepancies in hardware abilities are not properly catered for. Special flow control algorithms are not used in Metroring. Rather, an inherent attribute called **back pressure** is manifest in a heavily loaded Metroring network. Back pressure deals with the problem of flow control.

If a new packet arrives when a station's receive buffers are full, the new packet is discarded. An acknowledgement packet to the source will indicate the discarded packet needs to be retransmitted to the destination station. Acknowledge packets, that are received before the information is transmitted, are then also used to indicate if the destination station's receive buffer is empty. If either frame discarding or full receive buffer notification occurs several times on one link, the source station's frame queue fills up as its new contents are not constantly processed. In turn, the source station will then start discarding incoming packets. Thus, the effect of congestion at a node is referred back towards the source nodes; hence the name back pressure.

The mechanism for Metroring would need time to take effect between the source and destination stations. Thus, higher layer protocols for DQDB and FDDI which directly address the congestion problem, such as XTP, would be the better choice.

4.4.3 Capacity Planning Constraints of the Network

4.4.3.1 Principle

This criteria is only useful if critical logical communication links in the network are identified. The minimum allowable data rate on the communication links must also be specified for capacity planning.

The capacity planning ability afforded by different MAN protocols, is to be evaluated. The more data paths available or the more bandwidth permitted per data path in a network, the better the capacity planning ability.

4.4.3.2 Application

DQDB and FDDI networks share only one data channel between all the stations. However, Metroring has the capability of having at least two data channels between any two stations on the network. For example, figure 4.2a shows a Metroring network where one operational data path is used while the other is non-operational.

4.5 Summary

Table 4.2 shows a comparison of factors relating to the three MANs.

Utilizing the table will require the functional requirements of the network design problem to be prioritized. Attempting to resolve the problem by stipulating the network must have

(1) a synchronous packet service and
(2) the most reliable topology features

leads to conflicts as to which choice will be the best. Instead prioritized criteria can lead to easily resolved problems. Eg:

Factor	FDDI + XTP	DQDB + XTP	Metroring
AVAILABLE SERVICES			
Synchronous	2	1	-
Asynchronous	1	1	1
RELIABILITY: Topology Strength			
Least Configuration Changes	3	2	1
Most Correctable Link Failures	2	2	1
RELIABILITY: Hard Error Recovery			
Most Effective	2	1	3
RELIABILITY: Soft Error Recovery			
Frame Checksum Function	1	2	2
Frame Sequencing Function	1	1	1
MANAGEABILITY			
Ease of Planning	1	2	3
Implementation Cost Effectiveness	2	1	3
Ease of Configurability of Network	1	2	3
Ease of Expansion of Network	1	2	3
OTHER			
Fairest Bandwidth Allocation	1	2	3
Flow Control Effectiveness	1	1	2
Capacity Planning Ability	2	2	1

Table 4.2 Relative effectiveness of functionality criterion with 1 being most effective and 3 least effective. '-' indicates a protocol does not have the attribute.

1st requirement: Must have Synchronous Packet Service
2nd requirement: MAN with Reliable Topology
3rd requirement: Topology Planning must be simple

The table shows that DQDB meets these prioritized specifications most satisfactorily. Generally, to keep the problem simple it is advantageous to use only a few criteria.

Trying to weight four or more criteria by priorities, is not practical as the comparisons in the table are relative comparisons. Instead of a relative effectiveness measure, if an 'absolute effectiveness' measure was derived for how efficiently protocols meet each criterion, weighting would become applicable. An 'absolute effectiveness' measure would mean the effectiveness range of table 4.1 would be far greater than from 1 to 3. For example, effectiveness values from 1 to 10 could replace effectiveness values from 1 to 3. This will allow evaluation of the absolute effectiveness of a protocol's ability to meet one of the criteria. Achieving this measurement process requires

(1) the ideal fulfilment of a criterion requirement to be defined and
(2) the ideal criterion fulfilment equated to an effectiveness value of 10.

CHAPTER 5: ANALYTICAL COMPARISON OF MANS

Networks have different kinds of physical topology shapes like ring, bus or star. Therefore it is difficult to compare certain aspects of their performances such as data throughput and delays. A Standard Configuration is defined to address the problem of analysing different topologies' for comparing analytical performance characteristics of various MAN protocols.

The characteristics of MANs examined in terms of the analytical comparison method (as from section 3.4.2), are:

(1) maximum throughput of the protocol,
(2) bandwidth allocation to stations and
(3) packet transmission delays between stations.

These characteristics are represented qualitatively as parametric indices. They enable comparisons of the structural nature of MANs. The structural nature of MANs is analysed by predicting the theoretical behaviour of MANs under specified conditions, ie. the Standard Configuration. These indices aid in selecting a MAN for an application where such requirements can be explicitly stated.

5.1 A Standard Configuration

The specifications of the standard configuration chosen are:

PARAMETER	VALUE
Physical Topology:	Ring
No. of Nodes:	25
Length of ring (l_{ring})	100 km
Light propagation speed (n_{fibre})	0,665

The physical topology is chosen to be a ring as MANs are generally configured as rings for reliability of operation.

Complicated topologies, such as those encountered in Metroring, are not easily analysed. In addition to the topology structuring conforming to special protocol configuration rules, the loading on different parts of a complicated topology would also play a role in determining performance. As noted in section 1.5.3.2, loading of networks is not one of the factors considered in this chapter - we are looking at the nature of the physical network. Cases with complicated topology and loading would require simulation to analyse the effects of topological complexity and loading effects. Thus for this analytic comparison the simplest topology and loading requirements are considered.

The number of nodes have been arbitrarily chosen as 25. This parameter does not influence the physical performance of the network as much as topology and protocol parameters.

The length of the ring network, l_{ring}, has been chosen to make the standard configuration resemble a MAN in terms of geographical extent. The stations are taken to be spaced equidistantly from each other around the ring.

n_{fibre} is the light propagation speed as a fraction of the speed of light in a vacuum. The value used follows from Frost's simulation of FDDI [Frost et al 90].

5.2 Comparison of Performance

The characteristics for the analytical performance are defined in terms of several indices. The three example protocols are then compared in terms of the analytical performance indices.

5.2.1 Absolute Maximum MAC layer Throughput

The absolute maximum MAC layer throughput index is an idealized measure of maximum throughput of a MAN. This is an idealized measure because no implementation dependant factors get taken into account in defining this measure and several assumptions invalidating certain MAC protocol layer functions. The following assumptions define the exact nature of this performance measure:

- Only information frames of maximum length, as defined by the respective MAN MAC, are transmitted.
- None of the frames are corrupted. In practice this is a realistic assumption as a fibre medium has high reliability.
- Throughput takes place, and is measured, only on one link between stations (No external or secondary paths can be used for packet transferral).
- Information is transmitted in one direction only.
- Station addresses use the shortest format permissible for an information frame. This is viable as only a maximum of 25 stations will need to be addressed in the Standard configuration.
- Station hardware can process the packets as fast as necessary. This ensures that no saturation occurs at station input queues receiving packets from the network. This is assumed to make this measure independent of implementation dependant factors or higher layer mechanisms, such as flow control. Flow control intentionally degrades throughput so that hardware frame processing speeds are not exceeded.

The maximum MAC throughput is thus the "instantaneous" MAC user bandwidth, ie. the maximum throughput a node can attain in which it uses all available network capacity of one link between two adjacent stations. This quantity is useful in determining which network is best suited to a high-speed data communication application, where dedicated logical data links must exist between specific applications for a short duration of time.

Derivations for calculating this measure are found in appendix F. Independent variables used in the calculation are:

Maximum Frame Size - Maximum size allowed by the MAC protocol.
Inter-Frame Gap - The pauses between frames transmitted
Minimum Preamble Length - Unused portion at the beginning of the frame
Minimum Header Length - MAC relevant information before the user data portion of the frame
Minimum Trailer Length - MAC relevant information after the user data portion of the frame
Link Capacity - The raw bit rate the medium operates at

Table 5.1 shows the absolute maximum throughput for each example protocol at the MAC layer for one link. The independent parameter values used are obtained from Ross [Ross 89] and Mollenauer [Mollenauer 89]. Note that none of the MANs use inter-frame spacing. This helps increase the protocol efficiency.

The values given relate to special characteristics of protocols:

Maximum User Data per Frame - Maximum data size allowed to be sent to the MAC from the user layer above

Maximum MAC Efficiency - A measure of how well space on the medium is used to transport the user data

Absolute Maximum MAC Throughput - How much information the MAC user could possibly communicate per unit time

The maximum MAC efficiency is the same as approximately normalized throughput - it gives an indication of how well a protocol uses the bandwidth available to it. This is useful in determining how efficiently a MAC protocol data frame is structured in terms of frame field lengths.

Parameter	Units	FDDI	DQDB	MR
Max Frame Size	Bytes	4500	53	4096
Inter-Frame Gap	μsec	0	0	0
Min. Preamble Length	Bytes	5	0	2
Min. Header Length	Bytes	6	7	19
Min. Trailer Length	Bytes	6	4	0
Max. User Data per Frame	Bytes	4483	42	4077
Max. MAC Efficiency	%	99,622	79,245	99,536
Link Capacity	M bits/s	100	150	100
Abs. Max. MAC Throughput	M bits/s	99,622	118,868	99,536

Table 5.1 Comparison of absolute maximum throughput

5.2.2 Bandwidth Allocation

For the standard configuration, table 5.2 shows the available raw bit rate, b, in bits per second for a station. This is the bandwidth available at the physical layer for one station when it communicates as a component of the network.

Parameter	DQDB	FDDI	M-R
B (bps)	150M	100M	100M
b (bps)	12M*	4M	200M#

Table 5.2 Raw bit rate available per station

Appendix F explains how this value is derived for each type of MAN.

The station bandwidth for DQDB, indicated *, is a maximum value. The actual bandwidth used by a station in DQDB would be dependant on its physical position relative to the ends of the logical bus. For example, if a station is situated at the end of the logical bus, only one bus is necessary for transmission to any other station on the network. On the other hand, if the station were situated near the middle of the logical network and if destination stations on each side of the transmitting station were balanced in terms of number of requests, both busses would be well utilized. Note that the MAC operation would balance the bandwidth out between stations. Due to unfairness in bandwidth allocation to nodes being inherent in the protocol, this intended balancing of bandwidth is not entirely achieved. Some nodes will have slightly less available bandwidth than what this index predicts.

In Metroring, the value indicated # applies to each station since the total bandwidth for transmission is equivalent to that for all ports. The value given is for a station with two ports. The bandwidth available to any station is B times the number of output ports on the station. Another important consideration when using this measure with Metroring, is that each station has to relay information intended for other stations. Thus bandwidth is allocated for logical links even when nodes don't transmit data originating from their own users (as explained in section 4.4.1). In the case of Metroring this measure assumes the station is routing no other station's frames.

5.2.3 Packet Transmission Delays

The standard configuration is used in comparing delays in each type of network. Time is measured from transmission of start bit to reception of start bit. The processing time, t_{PROC} in only the source and destination nodes is not taken into account in these times. The delays for the three example protocols evaluated, are derived in appendix F and are:

t_{PROC} = The processing delay on a packet incurred by the station when the packet address needs to be processed before routing it.

t_{REC} = The delay incurred by the receiving components in the station.

t_{TRANS} = The delay incurred by the transmitting components in the station.

t_{max} = The delay over the longest possible path on which a packet can travel on ie. when source and destination stations are as far apart as possible for the protocol, eg. for DQDB the stations are at opposite ends of the bus. For FDDI when the destination station is the upstream neighbour of the source station.)

t_{min} = The delay over the shortest possible path the packet can travel in the direction of packet flow. This occurs when source and destination stations are physically adjacent to each other.

t_{ave} = The average transmission delay is the average of the measured delays as the number of packets transmitted on the network tends to infinity. Destination stations are assumed to be chosen using a uniform distribution random number generator.

These values are compared as shown in table 5.3. t_{PROC}, t_{REC} and t_{TRANS} are implementation dependant factors. However, their values need not be available for a comparison to be possible; they can all be set to zero.

In this specific evaluation, several values are assumed. Packet processing delay is not available for any of the protocols, but FDDI is specified to have a 10 bit elasticity buffer present in each station. Thus FDDI has t_{PROC} as a non-zero value equivalent to 10 bit durations. The values used for station receiver and transmitter delays are assumed if they cannot be calculated. For DQDB, only a transmission delay is assumed. It is assumed equal to half a bit's duration as a logical OR gate serves as the transmitting device. (Such a logic gate is assumed to have a delay close to its required operating rate.) In Metroring, half a bit duration is similarly assumed for receiving and transmitting the signal. This delay is then proportioned equally between the transmitter and receiver for Metroring. Values for FDDI's required receiver and transmitter delays were derived from Burr [Burr 88]. These delays were taken to lie halfway between the respective optical interface maximum and minimum required signal rise times.

Parameter	FDDI	DQDB	M-R
t_{PROC} (sec)	8,000E-08	0	0
t_{REC} (sec)	2,800E-09	0	2,500E-09
t_{TRANS} (sec)	2,050E-09	3,333E-09	2,500E-09
t_{max} (sec)	0,00037842	0,00037655	0,00024066
t_{min} (sec)	0,00001569	0,00001568	0,00001569
t_{ave} (sec)	0,00019709	0,00019612	0,00019614

Table 5.3 Delays experienced in MANs

These measures can be used to establish which MANs best fit the problem's basic delay requirements eg. the maximum permissible delay allowed between any two stations or the average propagation delay for the network.

CHAPTER 6: MAN SIMULATOR REQUIREMENTS

In deciding which network protocol is best suited to specific network requirements, the actual details of the requirements need to be taken into account, ie. examination of a protocol's performance under a specific network configuration and user load. This is best achieved via simulation of the protocol using the loading and network configuration as inputs.

After establishing the requirements for simulation, discussed in section 6.1, a system must be designed to meet the requirements as closely as possible. Sections 6.2 to 6.4 describe decisions and assumptions made during the design, as well as the operation of the system.

6.1 Problem Identification

For the creation of the simulator, several criteria are identified to define the scope and nature of the simulator design. For this part of the simulator design we distinguish between:

1) Simulator minimum requirements ie. the basic functions it must perform,
2) Special techniques or algorithms to be used in implementing specific simulator functions,
3) Limitations or restrictions needed for practical implementation of the simulator and
4) Tools and methods to be used in the implementation ie. the required resources,
5) Limitations or restrictions on the simulator design and operation resulting from the above factors that can be predicted at this point of the design.

6.1.1 Basic Requirements for the Simulator

Basic requirements are detailed here, while section 6.1.3 lists assumptions made in the implementation of the simulator, which also contribute to the requirements for the simulator.

A simulator that represents all possible detail of real networks is an ideal, however limitations do exist with real processors (processing speed and available memory) and this leads to models approximating what happens in reality. Sections 6.1.1 and 6.1.3 are disassociated because the requirements (as in section 6.1.1) of the design and the assumptions limiting the design (section 6.1.3) need to be separately examined. Separately examination of requirements and assumptions aid understanding and appreciation of the difference between the ideal simulator (accurately depicting the real world) and the practical simulator (approximating what happens in reality).

The minimal set of requirements for the simulator are:

6.1.1.1 The load, topology and basic protocol and network parameters must be user specifiable.

The network's load and configuration are the obvious inputs to the system. Other simulation and system parameters must be specified by the user.

Three ways of specifying the **load type** are possible:

Globally for the network - Particularly in evaluating protocol behaviours comparatively many simulations require the homogeneous (or uniform) behaviour of the network to be evaluated. Therefore, the load needs to homogeneous in nature for the simulation to be valid.

Uniquely for each station on the network - When looking at case study problems, differentiation between the load generated by individual stations on the network is necessary. By specifying the load for each individual station on the network, real life situations will be depicted more accurately.

External source - To depict a network's loading more accurately, the stream of packets generated by an operating network's load could be tapped. To cater for such a situation, the monitored load of the real network must be able to be fed into the simulator. Although the technicalities of this problem can be immense, a simple mechanism for feeding in a stream of messages to the simulator from a disk file is provided. This mechanism could be used for utilizing real loads recorded from operating networks.

The **configuration of stations** on the network is to be specified in terms of:

- The physical links to which a station is attached. Each link can be given an identifier through which stations can keep track of their respective output links.
- The physical length of each physical link.

Other parameters that need to be specified by the user are:

- The number of stations on the network
- User identification names for each station on the network. Association of load, configuration and output information is thereby facilitated for each respective station.
- The selected MAN protocol for simulation.
- Parameters specific to the operation of each protocol. For example, only one parameter was for FDDI: the set of TTRT values (Target Token Rotation Times) for each priority level in FDDI.
- General parameters needed to operate the protocols. These are the common factors needed for specifying protocol operation:
  - Line bit rates (speed of physical link operation)
  - Velocity of light propagation in the physical link medium as a fraction of the speed of light in a vacuum
  - Bit corruption rate; how often a bit is received incorrectly by a station - resulting in packet corruption
  - The hardware latency of a station; the time it takes for a bit to propagate from the receiving port of a station to the resulting output port of the station - this can be due to electronics delays or processing delays
  - The number of levels of applicable priorities for the simulation
  - The proportion of user data messages for each priority level
  - The limit to the size of the MAC queues. Packets arriving which exceed this limit are delayed.

6.1.1.2 Capability to implement the FDDI, DQDB and Metroring protocols

The capability to handle the example protocol's (FDDI, DQDB and Metroring) will enable:

a) Comparison to be made between several investigated protocols

When the protocols are modelled using a common set of assumptions (section 6.1.3), comparison of performance is standardized.

b) The protocol specific adaptions to be illustrated

The modelling of the protocols for their integration into the simulator illustrates how future protocols are to be adapted.

6.1.1.3 Collection of statistics on delays and throughputs of the network

Another important decision in the specification of the simulator, is the measurements it must take for assessing operation of the protocols. The most distinguishing feature of this simulator is its use for comparatively assessing protocol performance between different protocols, while the network configuration and loading are taken into account. Hence common measures of assessing protocol performance are needed. These common measures for protocol performance contrast against the unique measurement criteria for examining specific protocol characteristics.

Specific protocol characteristics have been examined in cases where the FDDI and DQDB protocols have been individually simulated. In the case of FDDI, criteria like token-rotation time, synchronous-access time interval and ring latency [Johnson 88], [Dykeman et al 88], [Jain 91] play an important role. With DQDB an important criteria is fairness which can be assessed from bus access delay and bandwidth utilization of individual nodes [Bisdikian 92], [Conti et al 91a].

General criteria like throughput and delays, as chosen in section 3.4.3, are the definite candidates for performance measurement. They allow the evaluation of the efficiency with which a network uses its capacity and the information transfer rate accomplished. As these are general criteria, the quantities the simulator will measure need to be cautiously specified.

Throughput and delays can apply to different aspects of a network's behaviour. Two important aspects that will apply to this particular situation are:

(1) The information throughput available to a user and the end-to-end delays experienced at this level. This is most important as it deals with the behaviour that any user of the network is exposed to.

(2) The physical network's behaviour. The throughput and delays experienced at this level are limited by the physical nature of the network. When one is designing a network (planning a configuration or load distribution) these indices can provide relative measures of topology strength or "quality" of load distribution.

Hence the minimum set of required measurements are:

1) **End-to-end physical delay** - The delay experienced by a packet from the transmission of its last bit at the source station output port until its last bit reception at the destination station input port.

2) **End-to-end user delay** - The delay experienced by a message from the time of its creation at the user layer until its final bit reception at the destination station. Messages are the units of data used at the user layer, as discussed in section 6.3.1.

3) **Physical layer throughput** - This is the amount of bandwidth used by the physical links during the simulation or the amount of physical bits transmitted on the physical line per unit time. It is different to the physical line bit rate, as it takes into account the periods when no data transmission is occurring.

4) **User layer throughput** - This is the throughput seen by the end user ie. the amount of useful data (in message bits) the system transfers per unit time.

As throughput can vary with time during network activity, measured throughput is to be the average for a period of time. The averages of these values (throughput and delay values) for the duration of the simulation will be the primary outputs of the system. Other outputs useful in describing the nature of the system can be determined from the necessary measurement data gathered to determine the primary outputs (described during design process).

6.1.1.4 Control of the accuracy of the output

The accuracy of the outputs needs to be determined or controlled for any reliable decisions. Controlling the accuracy is achieved by means of controlling the duration of the simulation using a sequential procedure. The aim is to calculate an average value for the specific measurement, with the value falling within a certain confidence interval. The confidence interval size is calculated from a user specified degree of confidence. In other words, the user specifies the degree of confidence with which an output must be given and the simulator will calculate the size of the confidence interval along with the estimate of the real value.

This is a departure from the more straightforward method in which the user can specify the level of confidence along with a required relative precision for the confidence interval. The relative precision of the confidence interval simply refers to the fraction the confidence interval size is of the estimate of the corresponding average. This relative precision is not used, because the user must have a fair idea of the final estimates of the average outputs they are dealing with *for all simulations performed* in the solving of a particular network problem. Figure 6.1 illustrates this scenario.

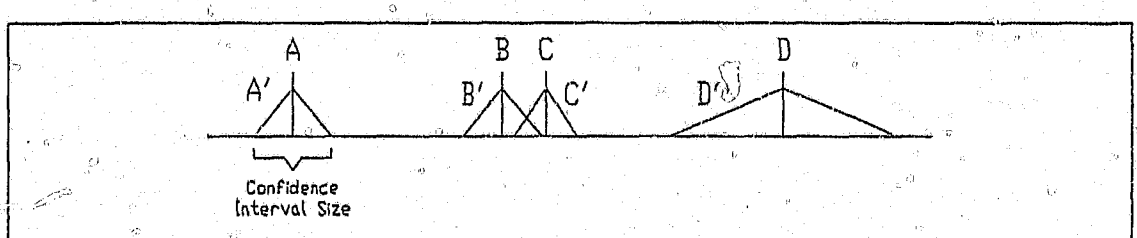


Figure 6.1 Interpreting estimates and confidence intervals
Readings B and C: Overlapping confidence intervals makes discernment of readings difficult, therefore make the relative precisions smaller.
Readings C and D: D's confidence interval is very large, but does not need changing as it does not overlap with C.

If the problem entails varying some of the network parameters to find a good solution to the design of a network, then observing one output (or measurement) of the different simulated scenarios would produce estimates (A, B, C and D) along with their confidence intervals (A', B', C' and D') as shown in the figure. If the estimates of the measurements vary unpredictably, there is difficulty in assessing the maximum size of the required confidence intervals for each case. (Confidence intervals should not overlap for one to assume, with the required level of confidence, one value different from another.)

Although this argument sounds reasonable, a converse situation can counter this. If the estimates are situated close to each other *and* their values be fairly accurately predicted, one can determine the relative precisions for each case of simulation. However, the initial reasons for using this simulator entail the fact that the user cannot predict the final estimates of the outputs. Thus determining feasible relative precisions for simulations is not practical.

While this line of reasoning for the choice of inputs for accuracy control is not very strong, the merit of this choice can only be properly assessed once the simulator is tested and results become available.

6.1.1.5 Accurate implementation of the protocol models

The implementations of the FDDI and DQDB protocols must comply to the IEEE standards (documents ANSI/IEEE 802.5 and IEEE 802.6), as outlined in [Ross 89] and [Sauer et al 90]. The Metroring protocol was not completely specified before its implementation, but due to assumptions 2, 3, 5 and 6 in section 6.1.3 many of the potential problems of modelling this protocol are eliminated. These assumptions also allow the simplification of the FDDI and DQDB protocol models for the simulator.

6.1.2 Special Techniques or Algorithms to be Employed

This section describes techniques and algorithms derived from the literature, which are considered necessary for the simulator's construction.

6.1.2.1 Statistical modelling of the user load

An important aspect of modelling the simulator is the way the load (of a station or the whole network) is modelled. The most practical way of doing this, is to model it statistically as in several other cases [Frost et al 90], [Conti et al 91a], [Sauer et al 90], [Johnson 88]. The load for the MAN simulator is characterised by:

(1) The length of the messages
(2) The inter-arrival time between messages
(3) The maximum length of messages (this is taken care of by maximum allowable lengths of packets at the physical layer.)

The additionally factor, taken into account in this simulator, is the maximum amount of data a station may generate. This is significant because it allows

(1) inactive stations, not produce any load for the network, to exist on the network and
(2) control of how much load a station inserts onto the network during a simulation.

Point (1) is applicable to real world situations eg. where bridges and routers (not using spanning trees) are part of the network and do not produce load for the network. Point (2) is a result of foresight as to the nature of data length distributions. Extra large messages might offset the effective average message length excessively over a simulation of a short duration. However, by specifying a maximum amount of data a station user may generate, the overall average of the data lengths is controlled. The user load, or throughput, is thus kept to the required average for the whole simulation as originally intended thus making the loading more accurate for short simulations.

The lengths and inter-arrival times of messages are specified as averages. The distributions of these generated values are exponential as they replicate asynchronous network load characteristics the most accurately [Sauer et al 83]. Some simulations have chosen packet lengths to be constant or fixed values [Conti et al 91a], [Sauer et al 90]. Nevertheless, these simulations have been primarily intended to analyse protocol specific mechanisms and behaviours. The random distributions of packet lengths will be better suited to analysing the load dependant effects of networks as this is representative of what happens with real user loads.

6.1.2.2 Confidence interval construction for monitored values

In any simulation, the monitored values are derived from the sample values which are derived from the process being simulated. One needs to determine just how accurate the resulting average value is.

As mentioned in section 6.1.1.4, the construction of a confidence interval for the monitored value is to be used. The user will specify a level of confidence (as a percentage/ probability), say p, with which the confidence interval size, say x, will be determined. From these measures we will be able to say the final estimate of the value has a p percent probability of falling within a tolerance band of size x of the true value of the quantity being monitored.

The method employed for constructing confidence intervals follows Sauer chapter 8 section 6 [Sauer et al 83]. It is referred to as the sequential procedure of Law and Carson and is preferred over the other methods since it is applicable to real world problems and dynamically determines the amount of data it requires. This procedure analyses the correlation between collected samples, and terminates the sample capturing, and thus the simulation, when the correlation is taken to be negligible. Correlation between samples was noted to be critical in statistical analysis [Sauer et al 83] and for this reason this different approach was employed to controlling the accuracy of results. Correlation between samples leads to a biased variance and inaccurate averages.

Refer to appendix G for the details of how the sequential procedure employed functions. The method employed in the simulator differs from the original sequential procedure in that the relative precision is not required.

6.1.2.3 Pseudo-dynamic load generation

The actual method of load generation requires special consideration. In the case of discrete event simulation, the packets or messages being generated for load take up finite amounts of memory. The whole load generated can then easily exceed system resources, such as memory, during long simulations. Notice that simulation length is not always predictable and the total number of messages generated can not always be determined.

A special method to generate messages in finite controllable numbers is used from a previous simulation [Dahl et al 91]. This discrete-event token-ring simulator [Dahl et al 91] employed a method for controlling the generation of messages. The method will be referred to as pseudo-dynamic load generation.

Figure 6.2 displays a portion of the discrete-event simulation time-line and helps demonstrate how this technique operates. In discrete-event simulation time is actually incremented in discrete measures or it is advanced to the next event in time. The labelled vertical lines on this time line indicate events in the system which get processed as the simulation time reaches these points.

The events labelled A, B and C correspond to events that will trigger load generation for the system. These points are equidistant from each other. After every load message generation, the next load generation event in the sequence is scheduled. At each of these points load messages are generated for each of the stations in the network for the period of time between the present and the next load generation event. The generation of messages must take place for each station in the network and the messages (A1 to A4 generated from event A, and B1 to B5 generated from event B) are inserted into the respective station scheduled input queues (these queues' sequences and messages depicted by the areas between the horizontal dotted

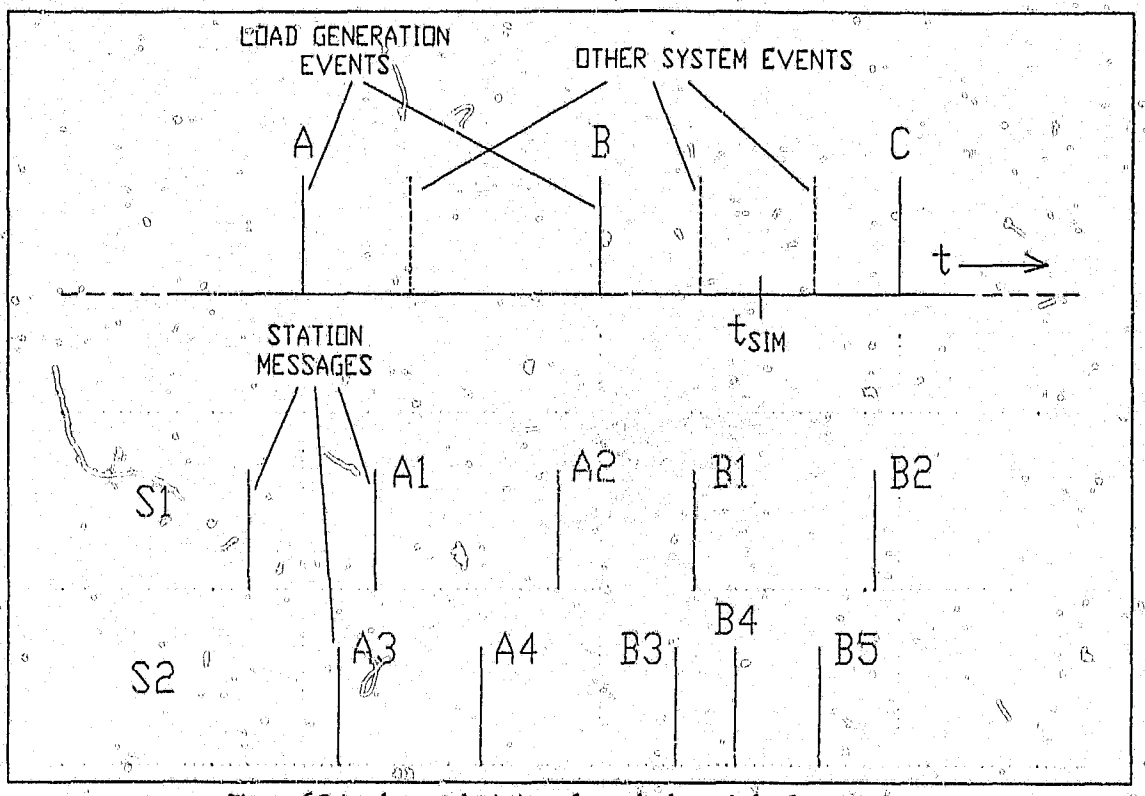


Figure 6.2 An abstract depiction of pseudo-dynamic load generation

lines: S1, S2, etc). The stations then only acknowledge the existence of each of these messages when the simulation time (t_{SIM}) exceeds the message's scheduled time in the system. (Events to the left of t_{SIM} have been processed by the system while messages to the left of t_{SIM} in the station queues have been processed by the stations.)

6.1.3 Assumptions for Practical Implementation and Operation of the Simulator

A detailed implementation of each protocol's operation is impractical. Several assumptions help narrow down the scope of the simulator design further. Many of these assumptions are implicit in the protocol models used in other simulators [Frost et al 90], [Conti et al 91a], [Sauer et al 90], [Johnson 88].

6.1.3.1 A special variation for the protocol stack to be simulated is defined

This is necessary for a common point of comparing protocol performances against each other, as was already discussed for the study of comparison methods in section 3.1.

The protocol functions considered for the simulator are:

- **framing** of a packet by the MAC layer
- **detection of packet bit errors**
- **routing of a packet** through the network, if necessary (not all protocols need this function)
- **addressing of packets** for identification of source and destination stations
- **connection management.** A restricted form is used to simplify higher layer operations being simulated. Only point-to-point connections are established. Only infor-

mation frames are represented.
- fragmentation of packets from the user layer when their size exceeds the maximum allowed by the MAC layer

As the whole of the MAC layer will be modelled, but not necessarily the rest of the protocol stack, the top of the MAC layer is also represented as the user end point. Hence, the throughput or delays measured at the top of the MAC layer can also be referred to as the end-to-end user layer measurements.

6.1.3.2 Only emulation of the basic MAN architectures without bridges or gateways to external networks is necessary

In assessing several MAN protocol's performances (in an absolute or comparative way) their interactions with the outside world are not assessed. This is one of the limitations to the scope of the study of comparison methods set out in section 1.4.

6.1.3.3 Implementation dependant characteristics should be absent from the protocol models

No unspecifiable delays in internal hardware or protocol stack traversal of packets (together generalized as processing time) should be taken into account. As the protocols are modelled from a functional point of view, such implementation dependencies are not considered.

Implementation details are not considered when dealing with the theoretical behaviour of protocols in the design and development stages. This is inherently assumed in papers dealing only with analysis of protocol characteristics [Conti et al 91a], [Sauer et al 90]. Although processing latencies in software do have big effects on performance [Frost et al 90], these delays cannot be modelled for Metroring (as there are no implementations available). Thus, such implementation dependant factors are not modelled.

6.1.3.4 Asynchronous traffic is the only class of traffic simulated

This is one of the original assumptions for MAN comparisons set out in section 1.4.

6.1.3.5 Steady-state network operation takes place

This was assumed in section 1.4 because the performance examined depends on loading and physical configuration of the network. The performance of the protocol would not depend on changes in hardware components or the network configuration. The performance indices used are meant to reflect steady-state operation anyway.

6.1.3.6 The simulator must work as fast as possible

This is a desirable feature of any simulator, thus implementing the protocols as efficiently as possible is necessary. This also means derived protocol models must be as simple as possible without omitting any functions as mentioned in section 6.1.3.1.

6.1.4 Tools and Methods for Implementation

The following resources and techniques for implementing the simulator have been selected:

6.1.4.1 Object oriented design using C++

This is a popular tool used for software implementation and development today. It is a language that supports software management by allowing the user to exercise flexibility and modularity in design [Stroustrup 92].

6.1.4.2 Discrete-event simulation techniques are used in the system

The discussion of the simulator's requirements has thus far assumed that discrete-event simulation is taking place. This method of simulation is used because it is best suited to time dependant simulations, ie. it works fastest [Sauer et al 83].

6.1.5 Resulting Limitations or Restrictions

These are limitations that can be anticipated ahead of time. Other requirements for the development or successful operation of the system can be identified at this stage.

6.1.5.1 A limited number of resources will be able to reside in the system at one time due to memory limitations.

This factor must be taken into account throughout the design phase. It implies memory should be used as efficiently as possible. To achieve this small data structures should be used and preventative measures taken for protection against memory saturation, which could possibly cause the simulator program to hang.

6.1.5.2 Simulation runs might be unacceptably slow even if designed for maximum speed.

In some protocol simulations, operation may be unavoidably complex and take large amounts of processing time to complete simple operations. This complex operation makes the simulation slow. If a specific simulation model is known to take long to execute, a time limited simulation option can be used.

6.2 Identifying the System Components

In creating a system that simulates different protocols, it is best to create an environment of software resources that each protocol entity can use. The protocol entity can then use these resources in a manner that best suites its implementation. The environment chosen consists of software objects that can be handled by the protocol entity. The protocol entities would also be software objects.

Figure 6.3 shows the component types of the simulator: the environment components, the protocol components and the user interface.

The component types are described below. The design of the environment and the protocol components are described in the next sections.

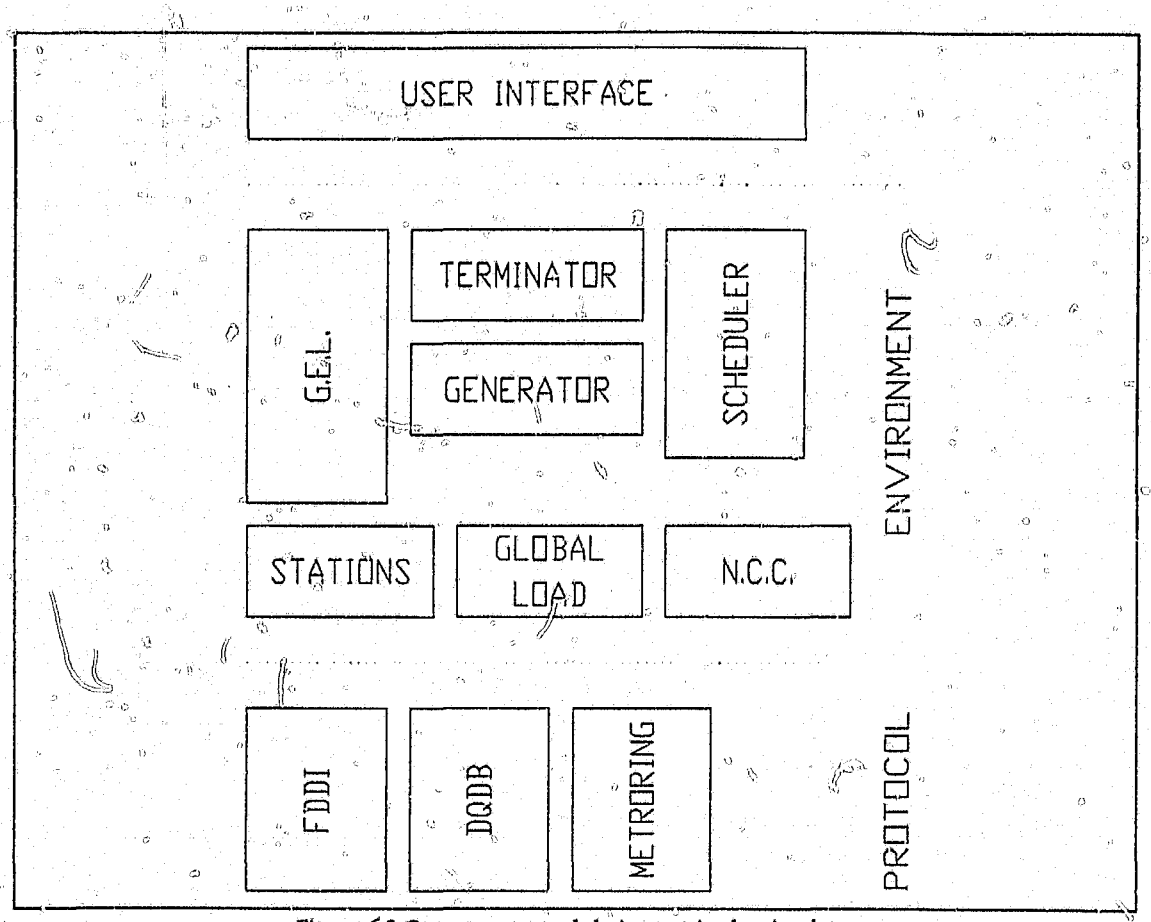


Figure 6.3 Components and their types in the simulator

6.2.1 User Interface

The user interface is not directly associated with the environment components or the protocol components, but it does influence the behaviour of the environment. Using this component of the system, a user can control simulation behaviours via parameters. These parameters are method of termination, loading of the network, topology of the network, the protocol to be used. Other criteria necessary for either the specific protocol to use or special settings for the system (like those listed in section 6.1.1.1) can also be controlled by the user from this user interface component.

The design of the user interface is not detailed, as it does not constitute the simulation mechanism.

6.2.2 Simulator Environment

The environment objects must, individually or collectively, provide functions to the protocol objects to be of any use. One important mechanism that must be managed by the environment is communication between the protocols and the environment objects. Here, events are thus used for communication between individual environment objects and between protocol objects and environment objects.

Two critical components needed for such a system are an event scheduler/ processor and a sequential store for events processed. These components, operating together, also help implement the discrete event simulation mechanism as well as allow communication between environment components.

6.2.3 Protocol

Protocol components or protocol objects are the entities which emulate the behaviour of stations on the network. Protocol objects reflect specific MAC and physical layer standards which specify the network's operation, eg. FDDI or Ethernet. Events, of which packets are a type of, are needed to activate and operate these protocol components. Several protocols are dealt with by the simulator via a protocol control module. The protocol control module will reference one of the several protocol model components (FDDI, DQDB and Metroring) for processing network events.

6.3 The Design of the Simulator Environment

Two aspects of the simulator environment's specification are separated. The individual components constituting the environment and their interaction with each other.

6.3.1 Components Constituting the Environment

The following are the objects constituting the environment. The two used for handling communication and simulation, the **Scheduler** and the **Global Events List**, are typical for such an application. The **Generator** and **Terminator** objects are unique to this system. They perform tasks providing self contained functions that the simulator operation requires.

Network Characteristics Controller (NCC):

This component simply routes events from the **Scheduler** through to the user selected protocol object. The general purpose of the NCC is to allow events to be channelled to one of several protocol entities present in the system. The protocol object to be used in the simulation has to be registered with the NCC.

Generator:

The **Generator** creates messages, for representing the data sent by the user. These messages are then broken down into packets (fragmented) for transmission on the network. The protocol object has control over this fragmenting process.

Terminator:

This is an object that checks whether the simulation must terminate or not. It is activated by an unique event sent from the **Scheduler**. After processing, it returns a code to the **Scheduler** to tell whether to terminate with the simulation or not. One of three different kinds of checks is performed, as determined by the user in section 6.1.1:

- Terminate by time limit (simulated time),
- terminate after a certain number of packets are received by the network or
- terminate after enough samples are available to meet a specified confidence interval.

Scheduler:

This is the component that actually performs the simulation. It extracts the time-ordered events from the **Global Events List** and routes each type of event to its corresponding processing element or component in the system. For this routing to occur correctly, each event processing component in the system must be registered with the **Scheduler** when the system is initialized.

Load Generation Information:

This is global data that specifies how a load must be generated for the network as a whole. It is used for setting up data type events, packets or messages, in each station. The overall effect is that the network's load is specified by the **Generator** using this load generation information.

Stations:

Stations are represented by unique classes, dubbed *Station*, that contain queues, links (ports) and logic link stations. They contain data on how the specific station is to generate a load statistically. *Station* also contains measurement data for the station's performance indices. All network stations' status data is stored in one global array, *Stations*. *Stations* must simply be referenced to access the correct data structure element (*Station*) for the corresponding network station.

Global Events List (GEL):

This is the storage point for all active events in the system such as packets and messages. When a packet is travelling on the network it is placed in this list. Then when it is to be received at a destination station it is removed from this list and passed to the protocol object to emulate a frame reception. Load generation and termination checking events are also placed in here. When the time comes for these events processing they are removed from the GEL and passed to the **Generator** or **Terminator**. The **Scheduler** is the component that actively works with these GEL contents.

6.3.2 Interaction of Environment Components

The interaction of information in the system is shown in figure 6.4. The **Scheduler** is the heart of the system - it actively passes events from the **GEL** to other components in the system. The **Scheduler** is activated from the **User Interface** and is not event driven like the other components.

Different types of events are processed by different objects. Specific events are used to activate the **Terminator** and **Generator**. Several types of events are passed to the **NCC** from where they are passed to the correct protocol entity for further processing (some of these events being packets). The **Terminator, Generator** and protocol entity then produce events which are placed back in the **Global Events List**. In this manner the simulation continues until the **Terminator** component causes the simulation to stop.

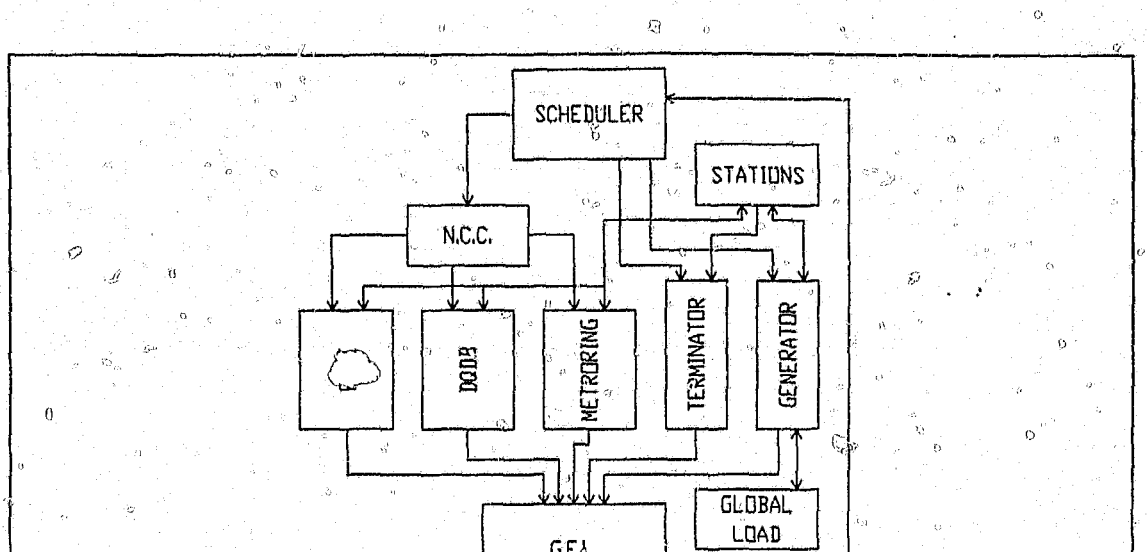


Figure 6.4 Information flow in the simulator system

6.4 Design of Protocol objects

The functional requirements for protocol objects are described for designing class methods to represent the protocol models working in the protocol object. An example of how modelling a protocol for implementation as a protocol object in the system is also demonstrated.

6.4.1 The functioning of the protocol object

A specific station's status and control information is used for simulating that station's behaviour. This information comes from two sources:

(1) The general information storage elements, applying to all protocol models and the system environment, from the *Stations* array.

(2) Information storage elements unique to each protocol model needed for correct protocol operation (described in section 6.4.1.5 below).

The functions performed by protocol objects using this information are:

6.4.1.1 Implementation of the MAC mechanism

The standardized version of the MAC mechanism is not represented here, as other functions from (section 6.1.3.1) must also be implemented. These functions are intended to realise end-to-end transfer of data units, thus, when necessary, the protocol objects must also take care of:

- routing of packets on different paths through the MAN
- concurrent data transmissions or receptions on several physical links connected to one station
- processing queues of packets with multiple priorities by giving precedence to respective priority classes

6.4.1.2 Emulating physical network behaviour

The physical behaviour of packets on the physical network is emulated by scheduling packets by their intended time of reception at their destination stations. The behaviour of packets while in their source stations is emulated whilst in the station's input queue.

The attributes of the physical network must be used to calculate the respective delay between transmission and reception of a packet. This transmission delay is derived from
- the length of physical links leading from source to destination stations,
- the speed of propagation of the packet on the medium and
- station hardware latencies.

Address fields in the packet event can be used for several purposes. Thus, as a rule of thumb, the source and final destination addresses must only be set in the packet being transmitted by a station before it is inserted in the GEL for processing.

Absolute event processing times of packets are *Packet* fields that are used for measuring points in time where the packet is created, transmitted or received. These fields can also be used for several purposes. Thus these fields must only be set to their required values at the latest possible logical points in the program.

After a protocol object receives a packet from the network (sent from the GEL by the *Scheduler* object) it must reference the respective station status data for the protocol object to simulate data receipt. This is achieved by using the receiving station's address as the reference.

6.4.1.3 Take performance measurements

Performance measurements are activated by the protocol object. A structure called *StatisticsType* uses the fields of the packet to collect the data for the required performance measurements (of section 6.1.1.3). The readings at the physical layer are prompted by:

```
Stations[CurrentStn].StatisticsState.phys_e2e[IPpkt->Priori
ty].
    add_delay(IPpkt->tRecv - IPpkt->tXmit);
Stations[CurrentStn].StatisticsState.phys_thru[IPpkt->Prior
ity].
    add_data(IPpkt->PktLen);
```

The readings at the user layer are prompted by:

```
Stations[CurrentStn].StatisticsState.MAC_e2e[IPpkt->Priorit
y].
    add_delay(msg->tRecv - msg->tCreation);
Stations[CurrentStn].StatisticsState.MAC_thru[IPpkt->Priori
ty].
    add_data(msg->PktLen);
```

6.4.1.4 Framing of data by the MAC layer

All user data passing through the MAC layer to the network has protocol control information overhead added to it. This function needs to be reflected by the protocol object operation. The fragmentation of the user data into acceptably sized packets must take this into account too. The designer of the protocol object has control over where and when the overhead

bytes of a packet are added to or taken off from a packet's *PktLen* field. The only important rule is that the adding or stripping of these bytes must be done between physical and user layer performance measurement activations. This rule is necessary as framing of packets is performed between the top of the MAC (or user) layer and the top of the physical layer.

6.4.1.5 Initialize protocol specific status data

Two types of data need to be initialized by the protocol object:

(1) The protocol specific data needed by the protocol model for operating the network. This data is comprised of fields local to the protocol object.

(2) The station specific data needed by the protocol model for operation of stations on the network. This data is represented by an array handled by the protocol object. Each element of the array is a structure that is defined inside the protocol object and contains the information necessary for making stations operate according to the protocol. Each element of the array refers to one specific station on the network.

6.4.1.6 Connection management

Connection management is necessary as the reconstructed messages must be processed at the correct times ie. only after the message's complete reception and reconstruction.

Destination stations keep track of all messages they receive via logical link connection structures, called *LogicLinkType*. There are ten of these *LogicLinkTypes* per station in an array called *LogicLinks* and each *LogicLinkType* can be used for managing one logical connection to a station. One of the *LogicLinkTypes* gets initialized at the destination when the first packet of a message is to be transmitted from the source. After all the packet's belonging to one message are received at a destination and the message reconstructed, the corresponding *LogicLinkType* is reset.

Two fields of packets are used to aid implementation of this connection management:

(1) The *tXmit* field of packets resulting from one message, are labelled with sequential numbers starting from zero. This enables determination of the first packet in the series belonging to one message so that logical links can be established in the destination stations before the packets are transmitted.

(2) The *tRecv* field holds the original message's connection identifier. The message identifiers ensure that the correct logical link is referenced in the destination station for every packet received. This is the same as the *MsgID* field of *LogicLinkType* structures, it is set equal to *no_of_packets_from_message* * 10 + *rnd*, where *rnd* is a random number from 0 to 9.

tXmit and *tRecv* are both set when the messages are first fragmented into packets.

This connection management at the destination is illustrated by the following skeleton code:

```
for (char g = 0; g<NoLogLinks; g++)
   if (Stations[CurrentStn].LogicLinks[g].MsgID == IPpkt-
     >tRecv) {

    IPpkt->tRecv = tCurrent;    // Update tRecv correctly
                                   before further pro-
                                   cessing
```

```
// take station's physical layer measurements for this
    packet
....
....
// update station's status
....
....
if (!Stations[CurrentStn].LogicLinks[g].NoPkts) {
    // close link and make a message
   Packet* msg = (Packet*)IPpkt->copy();
   if (msg) {
    msg->PktLen                                          =
        (Stations[CurrentStn].LogicLinks[g].MsgID/10
        -1)*  (FDDImaxPktlen-FDDIOverhead)  +  IPpkt-
        >PktLen;
    msg->Type = Message;
    msg->tCreation = IPpkt->tCreation;
    msg->tXmit = IPpkt->tXmit;
    msg->tRecv = IPpkt->tRecv;
    Stations[CurrentStn].LogicLinks[g].MsgID = 0;
    Stations[CurrentStn].LogicLinks[g].AdjStnNo = 0;

    // take station's physical layer measurements for
        this packet
    ....
    ....
    delete msg;
   }
  }
  break;
 }
if (g>=NoLogLinks) delete IPpkt;  // ie. the packet has
   been processed
```

6.4.1.7 Manage the generated load

This involves

(1) converting the generated messages (global data units) in the first queue of each station to data units (packets and slots) used by the protocol and

(2) sorting the messages in the first queue into the correct queues.

The messages represent the user datas needing to be sent across the network to their destination stations. They must be fragmented into packets so that the MAC layer can transmit them in viable form. This is achieved by fragmenting the data into packets of a size the MAC layer is intended to handle. Fragmentation is accomplished by duplicating the original source

message with its *Type* field set to *DataPacket* and the *PktLen* field (packet length) set to the maximum that the protocol will allow. This duplication is thus performed a specific number of times only, as demonstrated in the code listing below:

```
// pp points to the message to be fragmented
// the identification for the connection is calculated here
            int    ID    =    (pp->PktLen/(FDDImaxPktlen-
   FDDIOverhead)+1)*10+random(10);

// check if enough memory is available for creating the
   packets
if (ID/10*(sizeof(Packet)+16) < farcoreleft()) {

   pp->Type = DataPacket;
        for  (int  i=0;  i<int(pp->PktLen/(FDDImaxPktlen-
      FDDIOverhead)+1); i++) {
      np = (Packet*)pp->copy();
      if (np) {
         np->PktLen =FDDImaxPktlen-FDDIOverhead;
         np->tCreation = pp->tCreation;
         np->SchTime = tCurrent+i*FDDIinterFragTime;
         np->tXmit = i;
         np->tRecv = ID;
         L.insert(*np);    // insert the new packet into the
            station's queue
      }
   }
   if (np) {
   np->PktLen = pp->PktLen % (FDDImaxPktlen-FDDIOverhead);
   }
   delete pp;              // rid of message as it should not
      exist any more
}
```

If the station protocol model requires more than one queue to operate, all the messages generated are inserted into the first queue of the station and need to be sorted into the correct queues. Exactly how this sorting is performed depends on how the protocol object uses the queues. For example, sorting may be done with respect to packet priorities or destination station addresses.

6.4.1.8 Bit error detection

Packet errors are the results of bit errors. These bit errors are controlled by the protocol object designer via the '*Bit Error Probability*' input. This value, represented by the global floating point format variable *ErrBR*, must be used to calculate if a packet is meant to arrive at its destination corrupt or not.

ErrBR must be used in connection with the static floating point format variable *transmitPkt::accumErrProb* to determine whether a packet must be assumed as 'corrupt' or not. This packet corruption is determined by adding the probability of corruption of every packet transmitted to *transmitPkt::accumErrProb*. When *transmitPkt::accumErrProb*'s accumulated value reaches 1 the packet currently being transmitted is taken as corrupted. The *Error* bit of the *Flags* field in the corrupted packet is then set to 1.

This mechanism is not often used because logic connections opened in a destination station may never close when one packet in a sequence is never received. This is turned off by setting *'Bit Error Probability'* in the user specified parameters to zero. The default value is zero which has applied to all simulations performed for the MAN selection method.

6.4.1.9 Termination check activation

The **Terminator** object in the environment is activated by the scheduler for determining when a simulation must end. It is not meant to be referenced by the protocol object. Only when the 'Confidence interval termination' condition is used for the simulation, must measurement data from the packet fields be used to determine if the confidence interval condition is met. This measurement data is passed by:

```
ci_check (msg, CurrentStn, IPpkt->Priority);
```

where *msg* is the last packet received by the station,
CurrentStn is the number of the station that received the packet,
IPpkt points to the last packet received.

6.4.2 Implementing FDDI

To examine how the FDDI protocol is implemented, the operation of the MAC protocol operation is examined in advancing levels of functionality. The most fundamental operations are examined and modelled first, then the functionality is added in stages and incorporated into the protocol model. When protocols are modelled, such evolutionary modelling steps must be chosen and ordered carefully so that each stage is realized easily and the additions made easily. Otherwise the model realization process becomes too complicated and modelling errors are easily introduced.

6.4.2.1 Representing Token Rotation

The rotation of the token is illustrated in figure 6.5a. The equivalent in the time domain is represented on the time line in figure 6.5b by sequential events B, C, D and A marking the times where the token passes stations.

The symbols L_a, L_b, L_c and L_d on the network, figure 6.5a, represent the lengths of links between the stations. L_1 represents the physical length of the token on the network. On the time line, figure 6.5b, the same symbols represent the equivalent propagation times for packets traversing links and token transmission time.

One global packet is used to represent the token. All that happens for this behaviour to be modelled, is

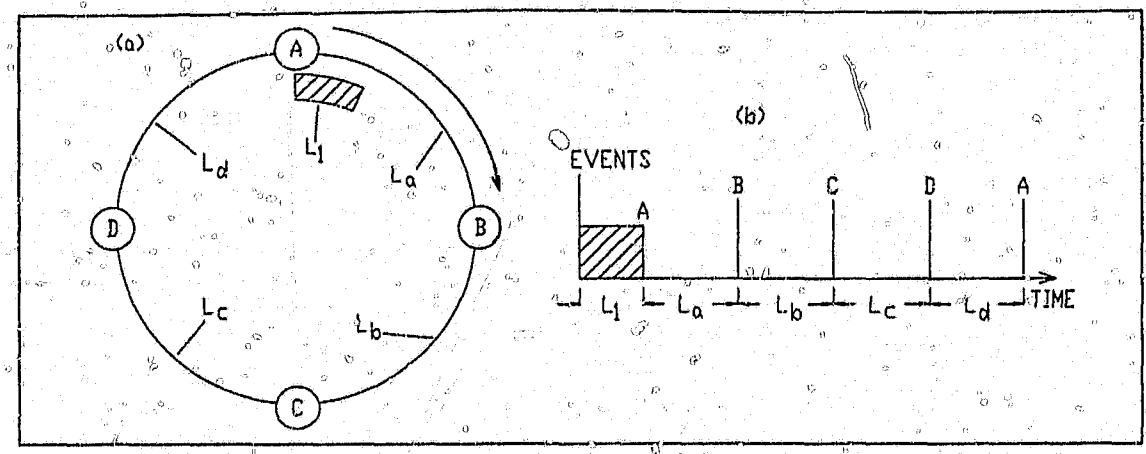


Figure 6.5 Token rotation representation in FDDI (a) on the network and (b) in the time domain

- one station, consisting of the protocol object with referenced station status data, receive the token event,
- the station performs necessary internal processing triggered by the token,
- the station schedules the token in the GEL for its receipt at the downstream neighbour.

The only delays involved in this process are the station's hardware latency and the propagation delay due to the length of the link between this station and its downstream neighbour.

6.4.2.2 Representing a Single Packet's Transmission

A token reception will allow a station to look through its input queue to see if a packet is available for transmission. If no packet is available then the token is simply passed on to the next station on the ring, as in the above section.

Otherwise, as shown in figure 6.6, the station places the available packet onto the network where it must then propagate from its source station (A) to its destination station (B). After reaching the destination station, the packet continues around the network until the source station where it is eventually removed from the network.

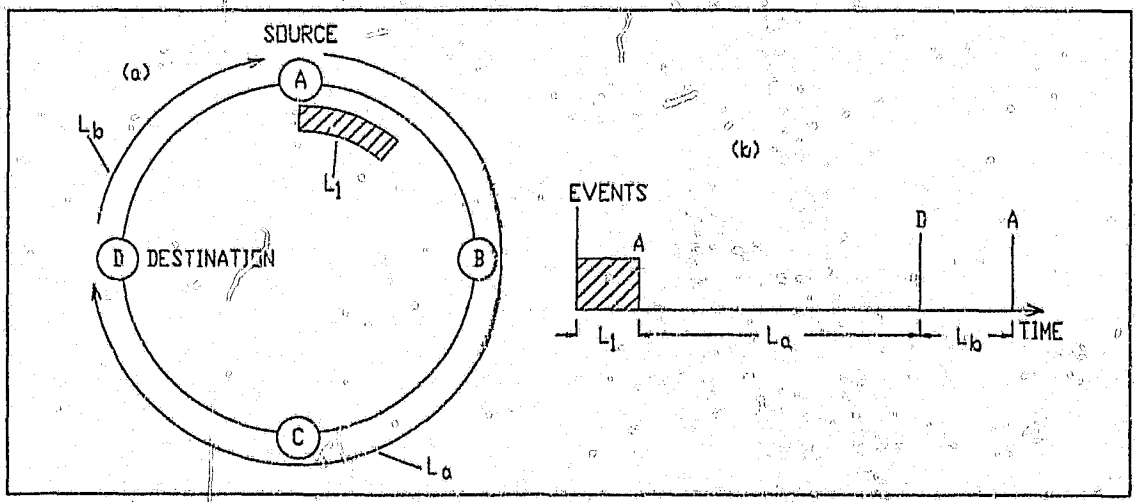


Figure 6.6 Packet transmission representation in FDDI (a) on the network and (b) in the time domain

This implies every time a station transmits a packet, the receipt of the packet by the destination station and the receipt of the packet by the source station, ie. itself, need to be modelled. This requires that the protocol object must differentiate between whether, an input data packet is intended for the destination station or the original source station. The protocol object must then reference the correct station status data for further packet processing. This differentiation can be determined by the *Last* field in a *Packet* structure, which discloses the last station to process the packet. If the *Last* field is the same as the *Srce* field (giving the originating station of the packet) then the *Dest* field (the destination station address) gives the true address of the station, otherwise the source station is being dealt with.

When the source station schedules a packet to emulate packet transmission, it must include both

- the packet transmission time (how long it takes to place the packet, first to last bit inclusive, onto the physical network) and
- the time taken for the propagation of the packet's last bit from the source station's output port to the destination station's input port.

All the physical link lengths from the source around to the destination stations along with all the stations' hardware latencies will contribute to this delay. After the destination station has processed the received packet (updating measurement data, etc.) it must schedule the packet for the source station again, now using the physical link lengths and the station latencies from the destination station back around to the source station.

A source station may only remove a packet from its MAC queue after the copy on the network is received from the destination station and it has been positively acknowledged (the *Copy* bit of the *Flags* field in the received packet is set). Source and destination schedulings of the packet must also update the *transmitPkt::accumErrProb* field to see if a packet corruption must be emulated.

6.4.2.3 Representing Multiple Packet Transmission

FDDI stations are capable of transmitting several packets before they give up possession of the token. This ability must also be modelled. However, simple continuous transmissions being sequentially scheduled during one event's processing by the protocol objects is incorrect. Termination and generation events in the system may need to be processed during this long single continuous transmission period of the station. These events could modify the current station's state before its transmission period is over. The transmitting station might need to update its input queue during the transmission period. For example, message fragmentation could re-determine the next packet in the output queue transmission sequence. Thus, the protocol object will need to employ a special mechanism for multiple packet transmissions for the duration of the multiple packet transmitting process.

The method employed is to use a special event for activating a station every time it is to attempt placing a further packet on the network. A *TransmitMode* event is used. After a station receives a token and it is possible to transmit at least one packet, it schedules the receipt of the packet at its destination station plus a *TransmitMode* event. The *TransmitMode* event is scheduled for itself at the end of the transmission delay for the packet. After both these events are scheduled, the protocol object releases control for system event processing to take place. The *TransmitMode* event is then processed by the Scheduler and the transmitting station resumes control with its last state of transmission. In this way every time after a packet is transmitted the station possessing the token can still try and transmit another packet that is due for transmission in the same transmission period.

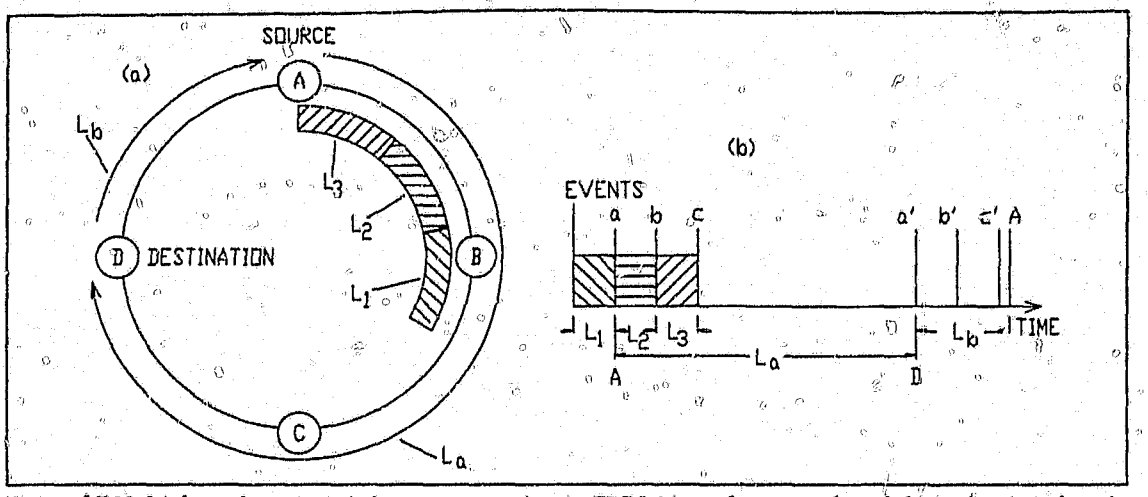


Figure 6.7 Multiple packet transmission representation in FDDI (a) on the network and (b) in the time domain

As depicted in figure 6.7 the *TransmitMode* events are scheduled for time points a, b and c. These events are scheduled after the respective packets 1, 2 and 3 are scheduled for receptions at the destination (assuming all packets are destined for the same station) for times a', b' and c' respectively. It is important to remember that this can only take place when there is time available (THT is positive) for packet transmissions.

6.4.2.4 The Timed Token Rotation Protocol

Figure 6.8 helps summarize how priority operation is implemented in FDDI. Frames of eight priority levels are processed, using eight corresponding time values (referred to here as priority times). The lowest priority's time is the largest.

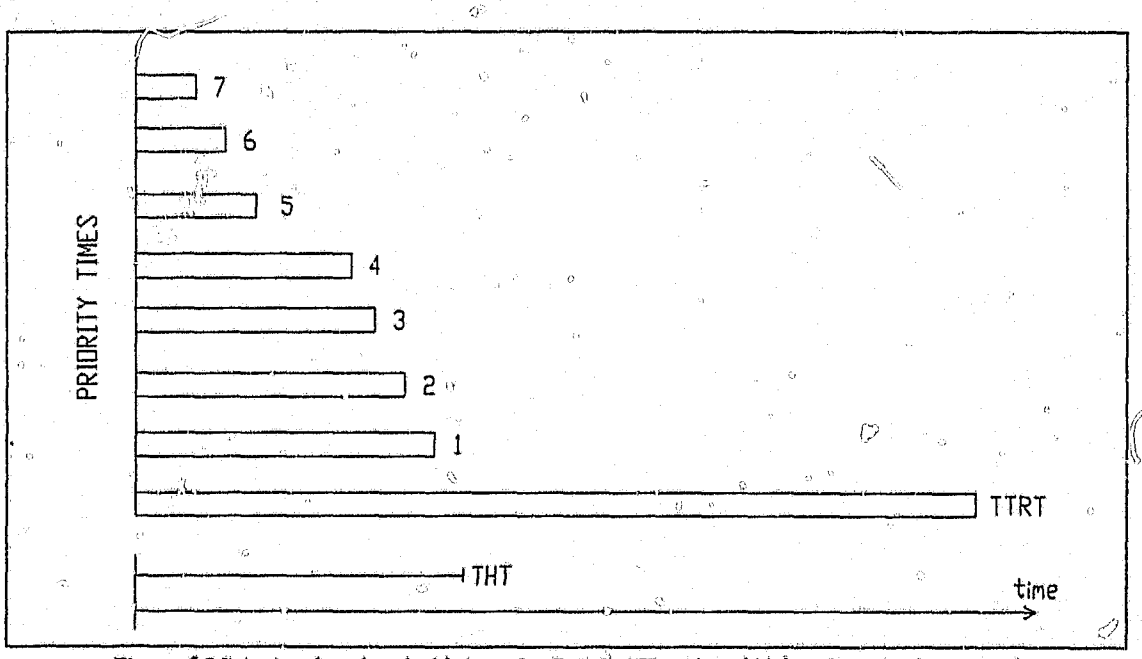


Figure 6.8 Priority class threshold times for FDDI's TTR (Timed Token Rotation) protocol

Frames of classes that have priority times less than the value of the THT will have transmission preference over other frames.

Implementing this rule, firstly requires the user to be able to specify the TTRT and the priority times. Thus an array of eight time values must be created for the protocol object and user interface to make reference to user specifiable priority times. Secondly, a routine that searches the input queue for a packet to transmit must be implemented for the priority mechanism operation. This is achieved by:

(1) Sequentially searching through the queue from the first packet (closest to the physical layer) to find the first packet which has its priority's time value less than the THT's value.

(2) If a packet is not found in step (1) then choose the first packet in the queue.

(3) If no packet is available then exit the routine and release the token from the current station.

CHAPTER 7: MAN SIMULATOR IMPLEMENTATION

Significant issues pertaining to the actual implementation and development of the simulator are discussed in section 7.1. The testing of the simulator's protocol models are then presented in section 7.2. Section 7.3 then considers the assistance of the simulator in making decisions in the selection and design of MANs.

7.1 Implementation and Development of the Simulator

The simulator was programmed in Turbo C++ V1.0 and named MANSIM (MAN Simulator). Several special features were used during program development are presented. A number of limitations also arose during the development and are thus also listed.

7.1.1 Features used for development of the program

To ensure efficient development of the program the following factors were taken into account and the necessary features implemented:

7.1.1.1 Ensure memory is used as efficiently as possible

Dynamic memory allocation was used to implement the event mechanisms of the environment. In Turbo C++ this is achieved by the user claiming and releasing memory in finite blocks. As the resulting system was complex, possible leakage can result in the available memory if less memory is released than was claimed. This will result in all memory being dynamically used up by some part of the system and not enough being available for proper overall system operation.

To ensure this did not happen, special memory checking routines were designed to keep track of memory allocation. During development, as soon as a memory leakage problem occurred, it could be rectified. Three routines were crucial in this checking:

long int **memreport** (void); This routine displays and returns the available space at the top of the memory heap (memory available for dynamic allocation). This enables automatic checking of whether memory that was allocated at one point in the program is released correctly at another point.

int **map_heap** (void); Here each block of allocated memory is listed to the screen so that the developer/ programmer can examine allocated blocks to see which object has not been released from the memory heap correctly.

void **list_objects** (void); The layout of each object as it would be allocated on the memory heap is displayed on the screen by this routine. Using these memory dumps as a reference the developer could use map_heap to identify which object was incorrectly unreleased.

These routines were only used during the development phase of the program. The code that performed the memory checks was thus conditionally compiled using programmer switchable compiler directives during this period of development.

7.1.1.2 Allow the tracing of operations performed during simulations

To help the developer the simulator's operations during a protocol's simulation, all operations performed by the simulator were printed to screen as they occurred. A limited form

of this operation tracing was implemented into the *Scheduler* object. The Scheduler object would print out the event currently being processed along with the simulation time of the event's activation.

Another form of tracing could be controlled by the programmer. Using a programmer switchable compiler directive, all operations performed by the protocol object could be listed to screen. However, the programmer would have to insert the print commands for each action and make these print commands conditionally compilable using the compiler directive.

7.1.1.3 Allow for short simulations during the protocol model testing phase

A simulation run using the confidence interval method of termination could take a long time to complete. Simulation durations using this termination method could easily exceed one hour. To delay analyses for such long durations during the operation testing phase of the protocol models was unnecessary. Control of the simulation duration during the operation testing phase would be necessary. Thus an extra simulation termination method was made available: the 'number of received packets' limitation. This would allow the program user to trace a limited number of packet propagations through the network before returning control to the operating system.

7.1.1.4 Known accuracy of results for all methods of simulation termination

Terminating the simulation via the time limited or packet limited termination options, will imply the required confidence interval might not be met. To provide a measure of the accuracy of the resulting measurements in these circumstances, the standard deviation for each network performance measurement is supplied with the output results. Standard deviation was the only measure of accuracy used in the LANNET II.5 simulator (described in section 7.2.1).

7.1.2 Limitations imposed on the Environment by the Development System

Several limitations arose during the development of the environment of the simulator which lead to some changes in the intended simulator's behaviour.

- Due to the limited amount of memory, a finite maximum number of packets could exist during simulations. If the generated load increased faster than it was processed, the memory could run out causing the simulator to crash. To prevent crashing, a minimum amount of free space was required after message generation for fragmentation and transmission of packets. If this minimum space could not be provided, the further generation of messages for the user load was inhibited. Thus in networks with large numbers of stations or a load greater than the network's capacity, the load could be inhibited at stages thus making results inaccurate. The user must examine results with this fact in mind.

- To limit the sizes of stations, the number of logical links a station could keep track of was limited to 10. This implies that in network simulations with large numbers of stations not all required logical links might be created. This leads to transmitting stations holding back packets of messages that can not be transmitted at the required times.

- The confidence interval method of termination uses an object that requires a very large array of floating point numbers to monitor just one measurement in the simulation. As memory must be used cautiously, only one of these objects was used for determining the length of a simulation when required.

  The value monitored was the end-to-end user layer delay of one station that was chosen at random when the simulation started. One station was chosen at random because it would imply that the rest of the stations would receive enough packets to approximately meet the same confidence interval construction requirement. These stations' precision in meeting this confidence interval requirement would depend on how equal the destination station mix was for generated packets. Generally some stations would meet the confidence interval requirement better than necessary, while others would not meet the requirement. However, the overall effect would denote the average end-to-end user layer delay *for the network* would meet the confidence interval requirement.

  As the end-to-end physical layer delay is directly dependant on the configuration of the network, it was not critical to monitor this measurement closely. This observation is further justified by the fact that the end-to-end physical layer delay is a fraction of the end-to-end user layer delay. Thus, as the end-to-end physical layer delay is directly dependant on the configuration, hence a constant measure, its effect is taken into account in the end-to-end user layer delay.

7.1.3 Protocol Limitations

Although each protocol model will have limitations specific to the protocol, some limitations are common to each protocol model:

- Protocol objects are not completely self contained because in some places in the user interface objects some protocol model specific inputs were required and owing to the initial user interface design these inputs were 'hard-wired' into the interface, ie. they were coded as part of the user interface. This creates unnecessary complexity when implementing new protocol models into the program.

- No MAC layer packet receive buffers are modelled. They would only be required if implementation specific packet receiving details were modelled. It is assumed the receiving software and electronics have no delays, as required in section 1.4.

Protocol Limitations in FDDI

- As no implementation dependant factors are taken into account, the frame preambles and elasticity buffers in each station are fixed. The latency of the ring is taken to be fixed, thus all station transmitters operate at the same fixed frequency, the baud rate of 125 MHz.

- Only non-restricted tokens are used. Details on restricted tokens were not available.

Protocol Limitations in DQDB

- No fairness mechanism is employed in the DQDB model as the standard revision used did not specify any ISO/IEEE accepted fairness mechanism for the protocol.

- No erasor class nodes were modelled as information was not available on th

- The physical configuration is initially set up as a ring before the network configuration is specified by the user. The logical break is restricted to lie between the first and last nodes for the duration of the entire simulation. This limitation was intentionally implemented in case the user intended the network be laid out in a bus topology.

Protocol Limitations in Metroring

- No Fast Forward option is implemented as it is only used for frames indicating critical network failures - something not needing modelling.

- A maximum of only 3 ports per station is allowed for the configuration of the network. This was to limit the memory used by Station objects.

- Each segment in the network configuration must contain at least one two port node between the SENs at the ends. This was to simplify the construction of the routing tables for each node in the network.

- All routing tables are not dynamically updated but are created at the protocol initialization stage and used as static data tables for the duration of the simulation. No temporary station status' are taken into account to determine the routing table paths. This is to eliminate complicated routing table updating during the simulation. It is assumed for this stage of Metroring development, that static routing tables will not affect the performance considerably.

7.1.4 Extent of Program Testing

Two areas of the simulator's operation have not been completely verified for correct operation:

- Multiple priority class use during a simulation. Only correct use of one priority class of packets in a simulation has been properly verified. The need for multiple packet priority operation of networks did not arise during the course of research.

- Input and output packet streams to or from the simulator. The need for using special streams of packets for loading the network or the analysis of output packets did not arise.

7.2 Testing the Operation of the Protocols

A thorough testing of the implemented protocols of the simulator was not possible because:

- Feasible network measurements were not available at the time of simulation tests. Full working DQDB networks have not been found in South Africa. FDDI networks were too small (in terms of physical size or loading) for consideration. Metroring has no implementations.

- Actual working simulators were difficult to access. One accessible FDDI simulator, called LANNET II.5, was used. Results from parameter driven simulation runs on this simulator are used. Other comparisons have been against results from papers which contain results from actual simulations. No complete Metroring simulator was available for comparisons.

The aim of the tests was to compare MANSIM's physical layer throughput and end-to-end user delays as a function of offered load to the network. Series' of at least 5 readings, offered load at least ranging from 40% to about 100% of the network's capacity, were taken so that the trend of the measurements could be followed. The effects of handling priority was not studied, thus in MANSIM only one priority class was used.

7.2.1 FDDI Tests

Two sources are used in the evaluation of MANSIM's FDDI implementation.

(1) A simulator called LANES (Local Area Network Simulator) is used by Johnson [Johnson 88] to analyse performance of the FDDI media-access control protocol.

(2) LANNET II.5 is a simulator capable of handling many types of LANs including FDDI. This simulator was used with the permission and guidance of Mark Chodos of WEB systems.

Both these sources allow for the comparison of MANSIM's physical layer throughput and end-to-end user delays as a function of offered load. They did not provide the measurements for end-to-end physical delays or user layer throughput comparisons.

7.2.1.1 LANES Results

The conditions and parameter values used for the LANES simulations come from [Johnson 88] and are:

Station hardware latency	60 bits
Number of nodes on ring	20
Distance between nodes	30 metres
Complete frame size	4040 bytes
TTRT	40 msec

The values of channel utilization and average frame delay for varying offered load are given in section H.1 with figures 7.1 and 7.2 graphically representing them along with the other simulator results. The two measures here correspond to physical layer throughput and end-to-end user delays. (Channel utilization is a percentage representing the fraction of channel bandwidth being used, which can be taken to be equal to physical throughput, while the average frame delay definition for LANES corresponds exactly to the definition of the end-to-end user delay.)

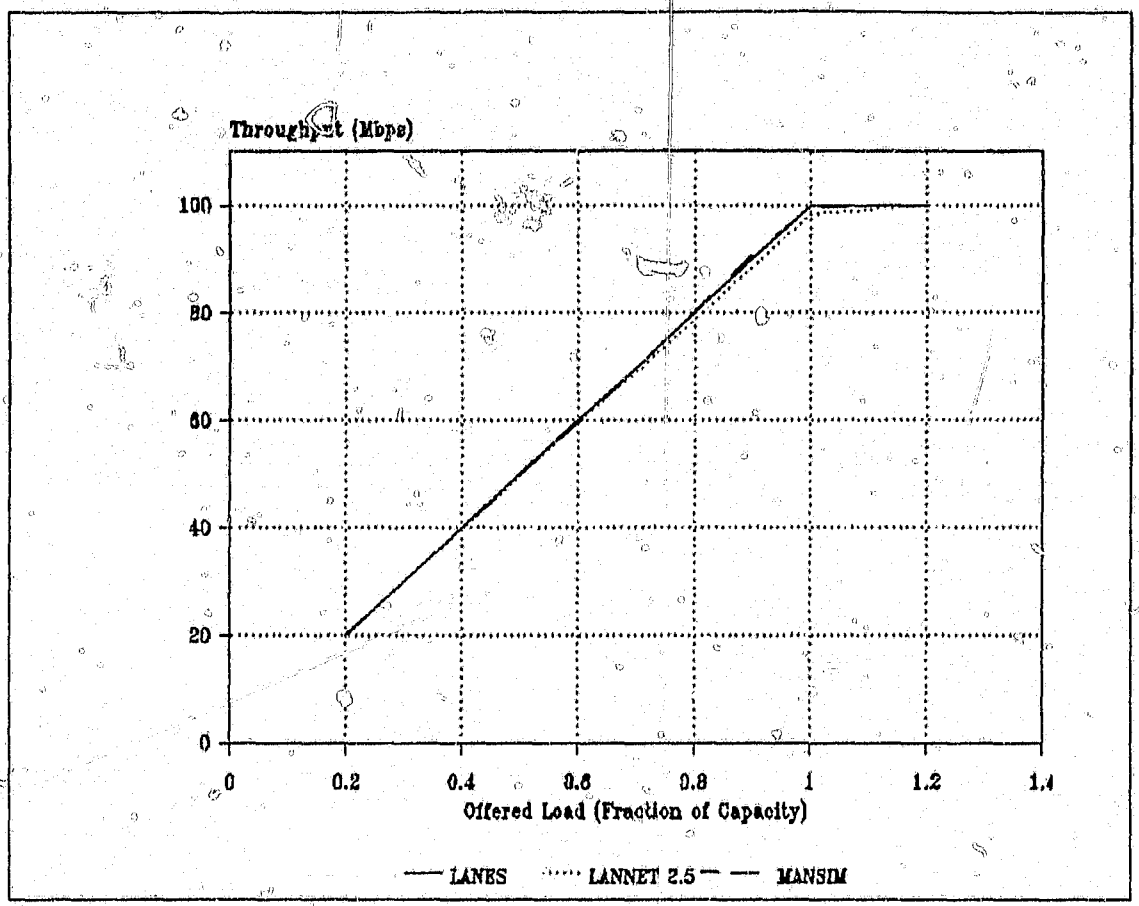


Figure 7.1 Physical layer throughput vs offered load (FDDI test results on three different simulators)

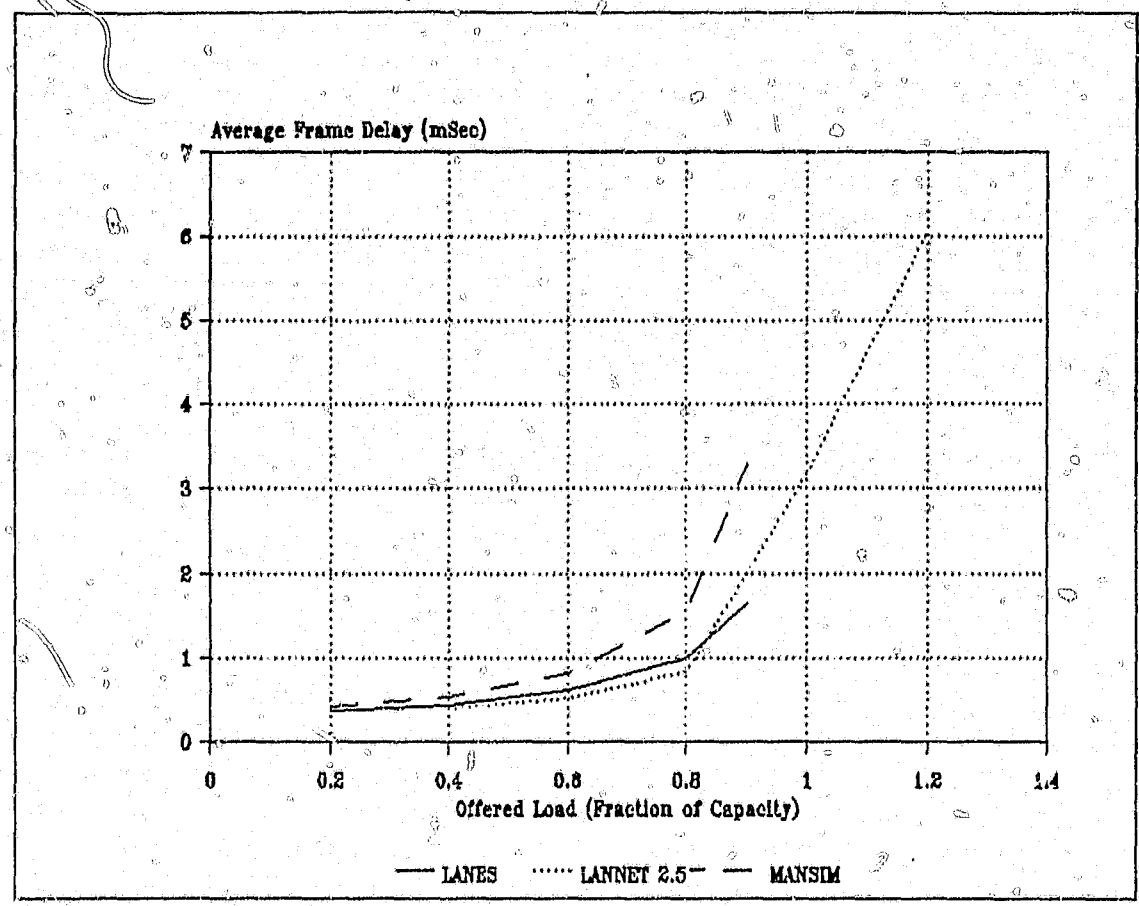


Figure 7.2 Average end-to-end user delay vs offered load (FDDI test results on three different simulators)

7.2.1.2 LANNET II.5 Results

LANNET II.5 is a parameterised LAN simulator (just like MANSIM) capable of simulating FDDI. Network and simulation parameters are set as close to those used in the LANES simulation as follows:

Fixed parameters (non-specifiable):

Medium Capacity (raw)	125 Mbps
Frame overhead	44 symbols
Symbols per frame	8956
Synchronous Allocation	2,2 μsec

User specifiable parameters:

User (MAC) Message size	fixed at 31968 bits (3996 bytes)
Average frame inter-arrival time for each station	(exponentially distributed, values given in section H.2)
Number of nodes on ring	20
Simulated time (run length)	1 sec

Parameters that could not be specified were: station hardware latency, distance between nodes and TTRT. It must be noted that synchronous allocation of bandwidth was fixed at a finite value (2,2 μsec) - this was inherently taken as 0 in the MANSIM and LANES as they did not take synchronous traffic into account.

The values of channel utilization and average frame delay for varying offered load using LANNET are given in section H.2. Figures 7.1 and 7.2 show these results along with the other simulator results. The required measures were interpreted from readings from the LANNET output report files:

- Physical layer throughput comes from Percent of time LAN is busy

- End-to-end user delays comes from
  (Average LAN request delay + Average LAN usage time)
  This is also equivalent to the weighted average of the stations' Average execution time (weighted with Completed message transmissions)

These LANNET measurements are interpreted as follows:

Percent of time LAN is busy indicates the fraction of time the ring is busy transporting packets.

Average LAN request delay is the average amount of time a station will wait before claiming the token for transmission

Average LAN usage time is the average time used by the LAN to transfer the whole message from source to destination station.

Average execution time is the average of the complete time a station takes to transfer a message from the source user layer to the destination station.

Completed message transmissions gives the number of message transmissions sent from this station to its destination stations.

7.2.1.3 MANSIM Results

The runs for the simulator model being evaluated use the same parameter values as used in section 7.2.1.1 with the following extra parameters:

Load generation method	Global network load
Maximum generated data	150 M bytes
Average frame inter-arrival time	(given in section H.3)
User (MAC) Message size	4014 bytes (exponentially distributed)
TTRT	40 msec
Station Hardware Latency	0,48 μsec
Simulation ending condition	95% confidence interval construction for end-to-end user delay

The values of average physical layer throughput and average end-to-end user delay for varying load are given in section H.3 with figures 7.1 and 7.2 being the graphical representations.

The average physical layer throughput for each simulator corresponds very closely to the LANES results from the paper. MANSIM can be said to simulate the average physical layer throughput for FDDI correctly for decision making purposes ie. the trend of the line for FDDI in MANSIM and the accuracy of the results are very close to the other results (which are accepted as correct).

Trends of results for the average end-to-end user delay shown in figure 7.2 are very similar. The MANSIM values at the point where offered load is 100% of the medium capacity were much too great to show on this graph while the corresponding values for LANES were not available. It can be taken that the trends tend to infinity at the point where offered load is 100% of the medium capacity for LANES and MANSIM. The trend of LANNET II.5 at the same point shows that the effect of saturation of the medium capacity does not effect the average end-to-end user delay. This indicates that the LANNET II.5 model of FDDI is not as accurate as it should be when the effect of medium saturation comes into play.

One distinct difference in the LANES and LANNET II.5 simulator models compared against that of MANSIM is that the MANSIM model is more pessimistic (has larger delays) than the others by a margin that increases as the offered load increases. This difference is due to the affect of two factors:

(1) Average end-to-end user delay is measured for *messages* originating from the user layer in MANSIM. In MANSIM messages can then be fragmented into several frames if they are too large for the MAC layer to transmit in one frame. In LANES and LANNET the frames that are eventually transmitted are created at the user layer (above the MAC layer).

(2) Message lengths are exponentially distributed in MANSIM, while they are of fixed lengths in LANES and LANNET.

In MANSIM, the proportion of messages larger than the maximum permitted by the MAC, which are also fragmented, would be the same for all values of offered load. The average end-to-end user delay for these fragmented messages is measured over at least two packet transmissions and thus causes the overall average end-to-end user delay to be greater than in the cases of LANES or LANNET.

7.2.2 DQDB Tests

Although many asynchronous load simulations of the DQDB protocol are documented in papers, they are mostly concerned with the problematic behaviours of the protocol like fairness. The results are mostly given in graphical form showing indices like node throughput or average node delays against the node index (position of the node on the bus) [Banerjee et al 92], [Bisdikian 92], [Conti et al 91b], [Phung et al 92]. Such results cannot be properly compared against MANSIM's behaviour mainly since throughput and delay measurements are recorded at a slot's destination station on the queue for MANSIM while similar measurements are recorded at the source station for other simulators. Since MANSIM sets packet destinations randomly for each source station, it is not possible to trace these measured values back to their source stations. Another major problem with such simulations is that they investigate the incorporation of fairness correcting mechanisms with the protocol and most often results do not include measurements taken without fairness correcting mechanisms included.

Conti et al [Conti et al 91a] use a simulation tool called RESQ (research queueing package) to perform a similar task investigating the fairness of DQDB. However he uses an index for fairness called MAC delay (which applies to the network in general) which is similarly defined as MANSIM's average end-to-end user delay. This criterion, which is tabled as a function of offered load, is investigated and compared against MANSIM's average end-to-end user delay. Throughput has a similar problem of whether it was recorded at the source or destination station - only overall network throughput would apply, but this is not addressed or tabled

in the paper [Conti et al 91a]. (The factor favouring use of results from this paper was that they were taken with and without the inclusion of the Bandwidth Balancing mechanism - see appendix B regarding fairness in FDDI.)

7.2.2.1 RESQ Results

The conditions and parameter values used in the RESQ simulations [Conti et al 91a] are:

Single Bus Capacity	150 Mbps
Number of nodes on busses	50
Distance between nodes	1016,33 metres
Complete slot size	53 bytes
Slot overhead (MAC)	5 bytes
User message size	48 bytes (fixed)
Buffer size	10 segments

One major factor affecting this simulation was the modelling of internet traffic, ie. slots destined for a gateway that would lead them to another LAN or MAN. Here 10% of the offered load was internet traffic and on each bus the gateway was a station situated at the end of the bus. This implied the destination address distribution is not uniform and can effect the measured throughput.

The average MAC delay, which is being measured here in the RESQ simulator, is defined as being the delay experienced by a slot from the time of its insertion into the distributed queue up until its complete transmission.

The values of average MAC delay for varying offered load are given in section H.4 with figure 7.4 graphically representing them along with MANSIM's average end-to-end user delay results. Only a small range of offered load is covered by these results as the aim of the investigation by Conti et al was to examine the behaviour of the DQDB protocol under underload and slight overload conditions (respectively referring to an offered load slightly lower or higher than the medium capacity).

7.2.2.2 MANSIM Results

The runs for MANSIM use almost the same parameter values as used in section 7.2.2.1. Other relevant parameters (or adjusted values) are:

Load generation method	Global network load
Maximum generated data	5 M bytes
Average frame inter-arrival time	(given in section H.5)
User (MAC) message size	48 bytes (exponentially distributed)
Buffer size	10 segments
Simulated time (run length)	2 msec

Note that a confidence interval is not constructed in order to control the length of the simulation. Initial confidence interval runs for DQDB were found to take at least 2 hours each. Instead, time limited simulation runs were performed as their simulation durations were approximately 1 hour long.

Three major differences exist between the MANSIM and RESQ simulator DQDB models:

(1) In MANSIM no internet traffic is modelled.

(2) In MANSIM all nodes contribute the same percentage to intranet traffic (slots destined for nodes on this network). In the RESQ model each node generates an amount of intranet traffic proportional to the number of downstream nodes. Although the effect of two busses implies each station contributes an equal percentage to intranet traffic in RESQ, the actual process of generating a destination node address for a slot differs from that of MANSIM. MANSIM generates destination addresses using a uniform distribution over the range of node addresses. RESQ would generate addresses for slots by considering the position of the source node on a bus and then using a uniform distribution for only the range of the number of downstream nodes from the source node. These two methods will produce slightly different results during short simulations because RESQ will intentionally portion the load from a station between the two busses, while MANSIM will only determine the bus for transmission of a slot from its intended destination address. When the mix of destinations for a node is unbalanced, which is most likely to occur during short simulations, the destination distribution for MANSIM, it will be slightly less balanced than that of RESQ.

(3) The DQDB input queue's behaviour differs for MANSIM and RESQ when it is full. MANSIM's extra slots are delayed, while RESQ's slots are discarded.

The values of average end-to-end user delay and average physical layer delay for varying load are tabled in section H.5. Figure 7.3 graphs these results.

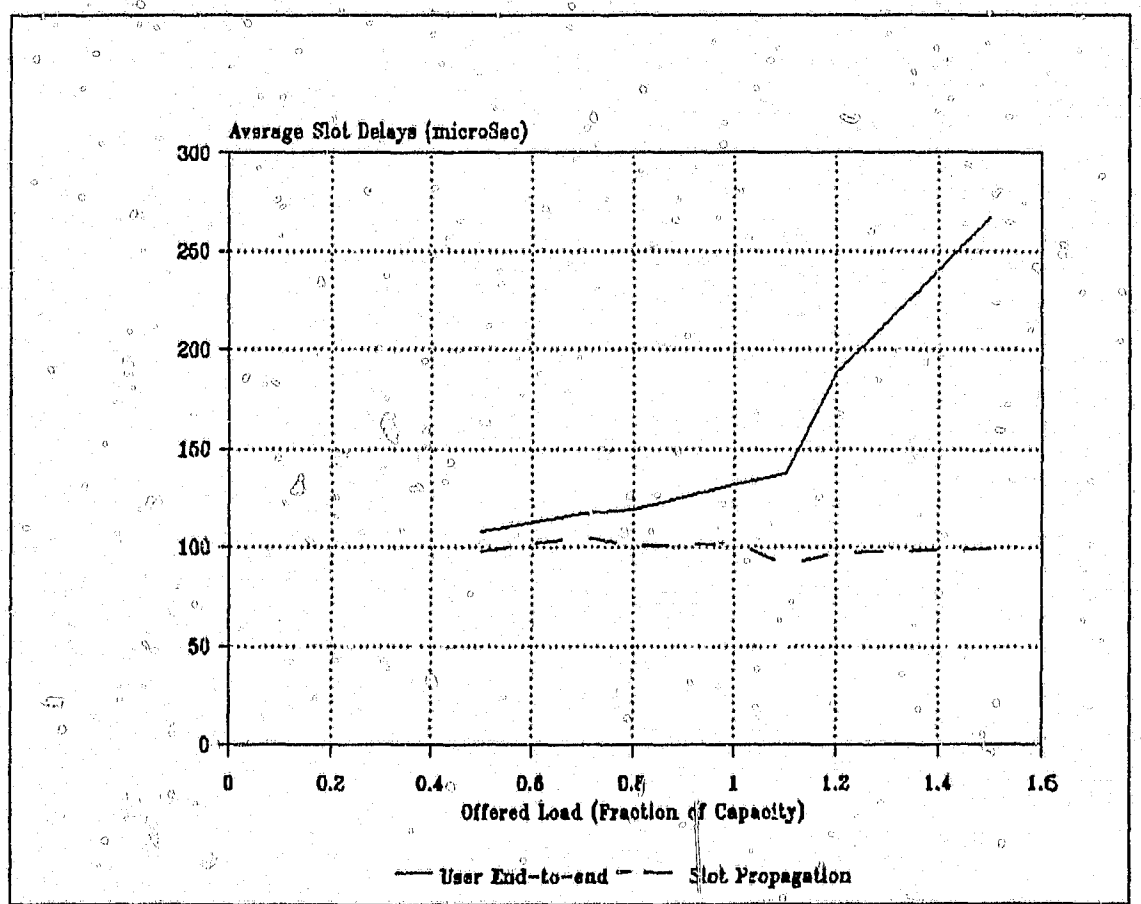


Figure 7.3 Average delays for MANSIM (DQDB test results).

The physical layer end-to-end delay in MANSIM is equivalent to the propagation delay of the packet. As the average physical layer end-to-end delay is a component of the average user layer end-to-end delay for MANSIM it is possible to represent the DQDB access delay more accurately. The access delay for MANSIM DQDB is thus taken to be the difference between the user layer end-to-end delay and the physical layer end-to-end delay.

Figure 7.4 graphs (1) the MANSIM DQDB average access delay, the (2) average MAC delay for the RESQ results and the (3) ideal access delay characteristics for DQDB [Hullet et al 88].

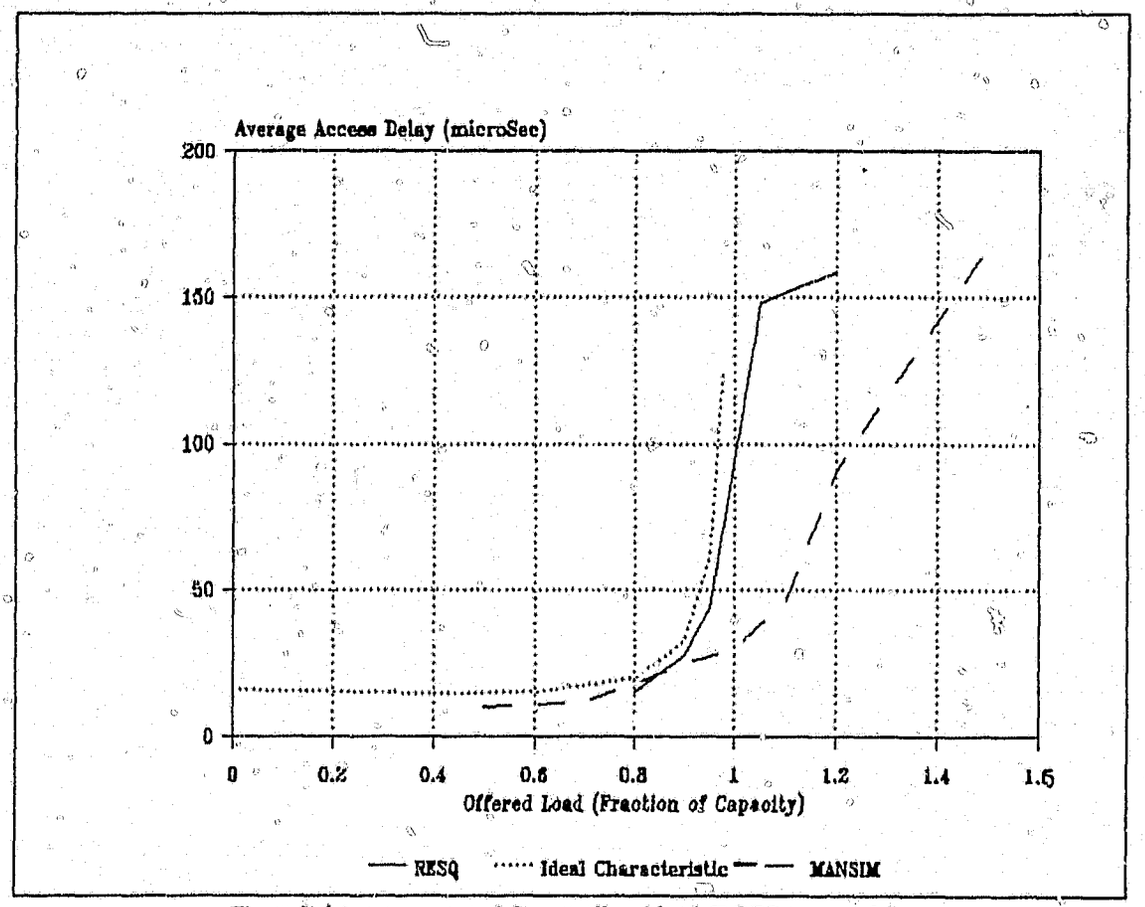


Figure 7.4 Average access delays vs offered load (DQDB test results).

Hullet [Hullet et al 88] gives these ideal access characteristics for a DQDB network operating under the following network parameters:

Single Bus Capacity	100 Mbps
Number of nodes on busses	50
Distance between nodes	2000 metres
Complete slot size	45 bytes
Average user packet size	100 bytes

There is no precise equivalence between either simulator and the ideal access delay characteristics for DQDB, even when the load is below 90% of the link capacity. Three factors contribute to this difference:

erived from a configuration with inter-node spacing of
ors' configuration has a inter-node spacing of 1816,33
s thus present when considering the ideal characteris-

ıus capacity was only 100 Mbps for the ideal case but
Higher access delays would thus be present in the
B.
eristics, Hullet et al used a shorter slot size than that
would decrease the average access delay slightly.
why the ideal characteristics differ from the simulator
comparison of the simulator and ideal characteristics,
ıalized to the simulator operating parameters. It was
cities was the factor that affected the bus access latency.
ncies and slot sizes do not differ tremendously as does
fer by a maximum of 10%, slot sizes differ by about 16%,
:0% thus being very significant even when bus capacity
work. (The differences are measured as the ratio of the
average of the two values.)

s a constant, the bit speed the cables are operated at,
g the cable. Accessing the bus now requires more time,
ass it. The ideal access delay characteristics for DQDB
ısidering the operating speeds for both sets of network
ɔmes:

thus shown normalized to the RESQ model's network

SIM access delay confidently follows the same trend as
offered load, a difference is visible between the MANSIM
SQ measurements. The RESQ simulator measurements
ı the ideal after about 105% of offered load (the ideal
100% of the offered load [Hullet et al 88]). Thus both
hen the network becomes saturated. It is assumed that
e simulators account for this difference.

user end-to-end delay will then be taken as acceptable

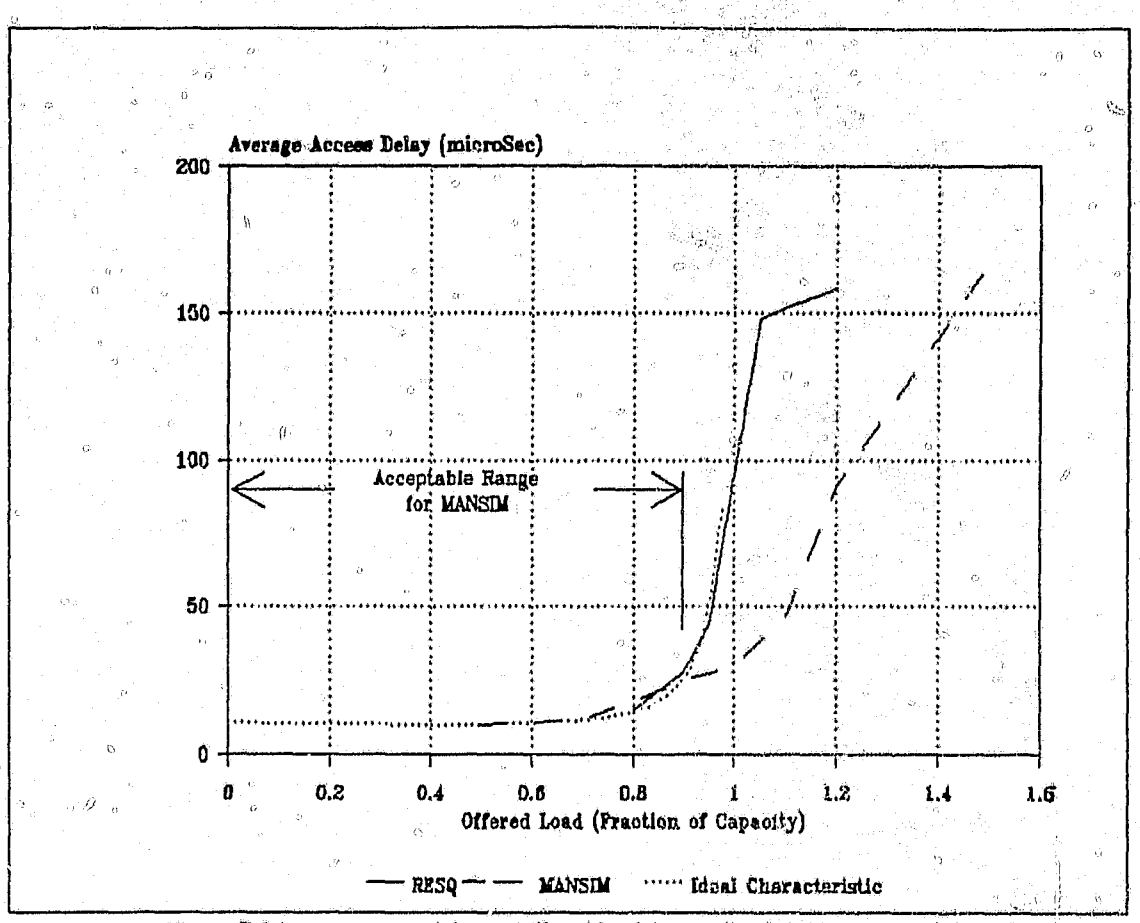


Figure 7.5 Average access delays vs offered load (normalized DQDB test results).

7.3 Application of the Simulator in Making Decisions

In section 6.1.1.3 throughput and delays are selected for comparing each protocol being examined, to determine which protocol performs best for a given configuration and load. To make clear decisions, comprehension of delay and throughput indices is important in interpreting each of these simulator output measurements.

As decision criteria, the user layer delay and throughput measures should take primary importance. Delays take priority at the user layer, as they take both physical configuration and user contention on the network into account. User layer throughput will not take precedence as it is mainly dependant on the load generated by the users. This prioritization of indices is possible because the more design parameters an index reflects, the more applicable it is in the decision process.

Ideally, the user layer throughput should follow the offered load from the application. As the capacity of the medium is reached, the effects of saturation make the throughput level off to a constant value. Protocol efficiency, which is less than 100% for real protocols, will cause the real user layer throughput to fall below the ideal value. End-to-end user layer delays should generally only become very large when the offered load equals or exceeds the medium capacity. End-to-end user layer delays will also be subject to the nature of the medium access method and the effect of packets queued in the user layer input queue.

At the physical layer, throughput reflects how well the network's physical capacity is utilized. This indicates when a data path in a network requires an extra link to reduce saturation. The physical delay reflects the physical size of the network, in terms of number of stations and physical link sizes. It indicates an optimally arranged network configuration (minimum end-to-end physical layer delays).

Thus the physical layer indices are most useful for designing network configurations and load allocation layouts. The user layer indices are best at providing the performance measures that allow selection of MAN protocols for solving a load and configuration problem. After all, the real decisions are based on what the users of the network require.

CHAPTER 8: APPLICATION OF COMPARISON METHOD

The application of the study of comparison methods is discussed in this chapter. Information on an existing LAN backbone was available and precipitated an illustrative investigation into which of the three investigated MAN protocols performed best in place of the LAN backbone.

8.1 LAN Backbone Investigation

The LAN backbone spans about 2 km^2 and is used to interconnect 32 buildings into a ring-like topology. The current protocol used is the IEEE 802.3 Carrier-Sense-Multiple-Access with Collision Detection (CSMA/CD), better known as Ethernet. The bus architecture required for the protocol to operate correctly is formed from the ring-like topology by segmenting the ring into 8 CSMA/CD collision domains using bridges at the collision domain boundaries as in figure 8.1.

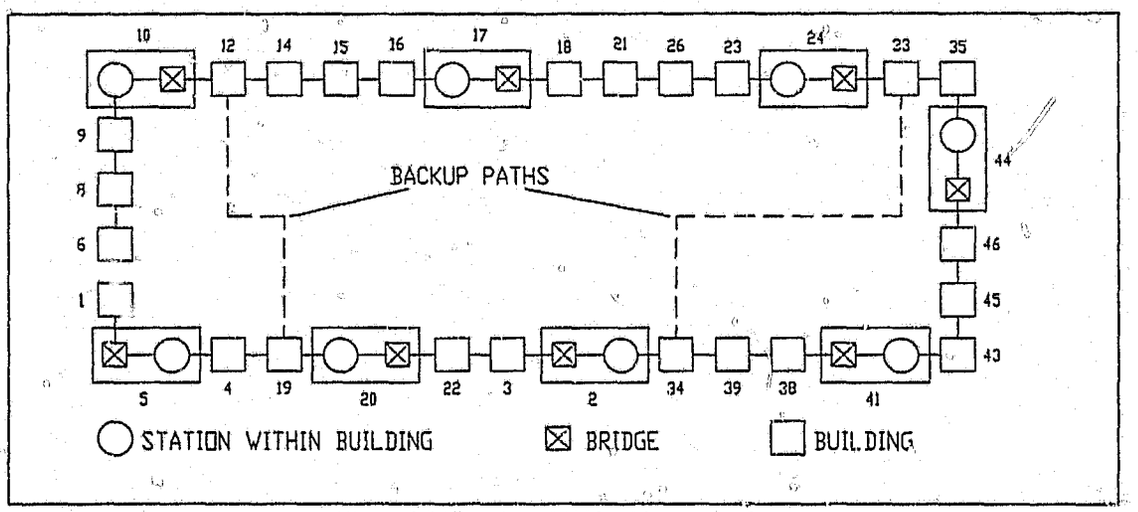


Figure 8.1 The given LAN backbone topology

An investigation into which MAN protocol would best suite this backbone's implementation is undertaken to illustrate how the comparison method works. We also examine (1) the connection of buildings as individual nodes and (2) the connection of LAN domains to the backbone to see which method of configuration to the backbone is most efficient.

8.1.1 The Nature of the Problem

Information on this backbone included the configuration of the backbone but no physical link sizes, time relative traffic measurements for each collision domain and limited general background information regarding the bridges and management system used for the backbone. However, not all of this information proved useful:

Collision Domain Traffic Measurements: Utilization and frames per second were given for every 10 minute interval over one day for a measurement site on the network. Eight measurement sites, each located on one domain, were monitored to derive the traffic profile for each domain on the network. However, utilization does not help determine loading profiles (which would help in the MAN applicability analysis) for each of the collision domains as it is the resulting load that is measured. The only notable observation made from these readings is that the load for the backbone averages 10%.

Management System Operation: Bridge and network management was described briefly only mentioning the degree of control a network manager has over the network.

Bridge Operation: The local bridges for the backbone are controlled from one station using Bridge Protocol Data Units (BPDUs). The functions of the BPDUs were only briefly listed giving no useful technical details when considering network behaviour.

The configuration of the backbone (figure 8.1) provided:
- The number of buildings (nodes) on the backbone
- The configuration of the backbone (one ring with two back-up paths)

It did not provide information on:
- Load placed onto network by each station
- Physical link lengths

For the investigation to be plausible, several assumptions were made:
(1) The back-up paths (in figure 8.1) separate collision domains. Eight distinct collision domain loading profile measurements were identified but not all collision domains are identifiable in the given configuration. The back-up paths serve as ideal boundary points as they would separate the backbone into 8 collision domains.
(2) A loading is assumed for the network, each node (building) contributing equal amounts. In the simulation case where the collision domains connected to the backbone, load from each collision domain (modelled as individual nodes) is proportional to the number of nodes on it.
(3) The ring circumference is taken as 6 km because the circumference of the 2 km^2 area is assumed to be formed by a rectangle 1km x 2km. Nodes are spaced equidistantly around the ring. In the case of collision domains connected to the backbone, the distances between the collision domains are taken to be proportional to the number of buildings on the collision domain.

8.1.2 Comparing the Performance of FDDI, DQDB and Metroring as Backbones

Functional and analytical comparisons do not apply in this case because such requirements have not been specified or assumed and, thus, the respective parts of the study of comparison methods will not apply. Simulation of the backbone will prove to be the best way of determining which of the three protocols will apply best to it. The primary index (also see section 7.3) used to determine this would be the average end-to-end user delay, for it the most noticeable effect at all levels of user level loading. User layer throughput, will be the secondary index here, as it only becomes significant in situations where overloading (load is greater than the network's capacity) at the user level occurs (and this is rare in reality).

Dahl and Harris [Dahl et al 93] also investigate the matching of protocols to network configurations and loads. They perform similar simulations for the same configuration.

8.1.2.1 Parameters for the Simulations

For each protocol examined, a slight variation in parameter values and network configuration and loading is required. Five simulations for each case are performed where the offered load takes on values from 50% to 150% of the medium's capacity so that the effect of load variation is taken into account. (The actual load averages 10% of the medium's capacity but this is impractical to this investigation - we wish to look at situations where the MAN technologies investigated are used because of their potential.)

For each protocol investigated, the two proposed methods of configuration, (1) the connection of buildings as individual nodes and (2) the connection of LAN domains to the backbone, are studied separately.

FDDI

Buildings on the Backbone

The configuration for FDDI consists of 32 stations in a ring topology spaced equidistant from each other by links of 187,5m in length. Figure 8.2 illustrates how the link numbers and port numbers generally work for the specifications of the configuration in the simulator. For this simulation the user need not specify all these link and station relationships as the simulator sets up the basic ring structure automatically. Note that for FDDI only one port link specification is required, while for DQDB two port specifications need to be specified. Every port specification in the configuration refers to a unidirectional link. Hence FDDI stations have one port per station for the unidirectional ring and DQDB stations have 2 ports per station for the bidirectional bus topology.

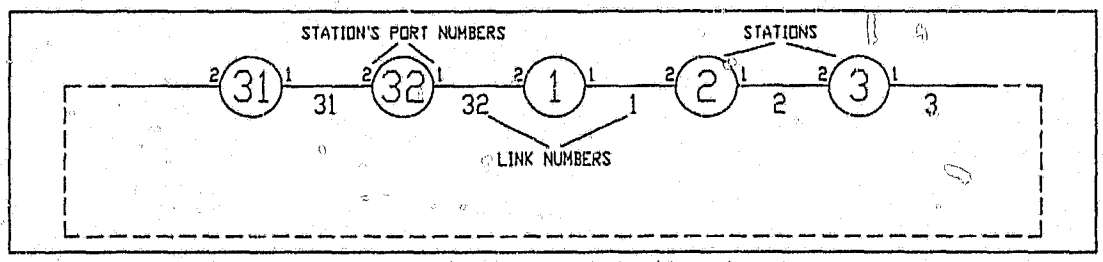


Figure 8.2 Representing buildings on the backbone in a ring topology

The other parameter settings are (other simulation parameters use default settings):

Load generation method	Global Network Load
User (MAC) Message size	4474 bytes (exponentially distributed)
Maximum generated data	200 M bytes
Average frame inter-arrival time	(given in section I.1a)
TTRT	10 msec
Station Hardware Latency	0,48 μsec
Simulation ending condition	95% confidence interval construction for end-to-end user delay

Collision Domains on the Backbone

This configuration uses 8 stations in a ring topology to represent the 8 collision domains connected onto the backbone. Figure 8.3 specifically illustrates how the link numbers, port numbers and link lengths (LEN) work for the specifications in the simulator for this case. The weighting factor for each station (WT) refers to the relative load weighting required to simulate the loading of the respective CSMA/CD collision domains. See section I.1b as to its use.

The other parameter settings are:

Load generation method	Individual Station Load
User (MAC) Message size	4474 bytes (exponentially distributed)

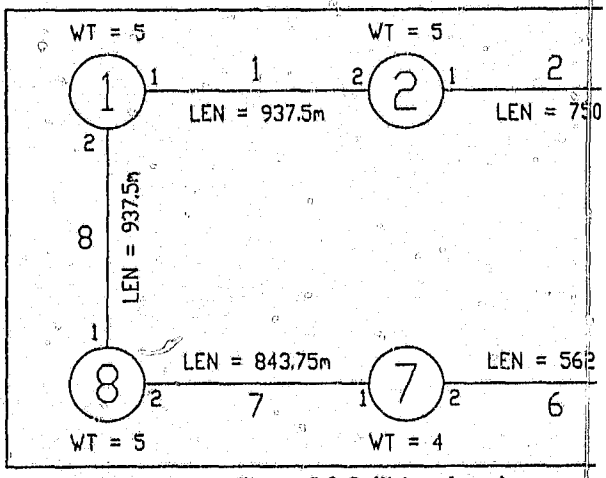


Figure 8.3 Collision domain repres

Maximum station generated data	10 M byte
Average frame inter-arrival time	(given in s
TTRT	10 msec
Station Hardware Latency	0,48 μsec
Simulation ending condition	95% confi user dela

DQDB

Buildings on the Backbone

The configuration for DQDB is virtually the sar simulator automatically sets up the basic ring s specifications are used for every station.

The parameter settings relevant to this particul

Load generation method	Unique S
User (MAC) Message size	48 bytes
Maximum generated data	1 M bytes
Average frame inter-arrival time	(given in
Simulation ending condition	Simulate

The simulation ending condition was simulated to approximately 1 hour. Confidence inter simulation to continue for at least 2 hours.

Collision Domains on the Backbone

This configuration for DQDB is again virtually

The other parameter settings are:

Load generation method	Unique S
User (MAC) Message size	48 bytes

Maximum station generated data	100 M bytes
Average frame inter-arrival time	(given in section I.2b)
Simulation ending condition	Simulated time of 2 msec

Metroring

For Metroring, the two configurations consist of 2 extra stations to allow representation of the back-up paths as shown in the original topology. Refer to section 7.1.3 as to the reason for this requirement. These Metroring topologies thus have 6 segments.

Buildings on the Backbone

Shown in figure 8.4 are the link numbers and port numbers, which work slightly differently than before, in order to take these two extra stations into account. It is thus necessary for three outgoing links for each station to be specified. The user must specify the special connections for node numbers 1, 11, 20, 25, 33 and 34. The two extra stations, 33 and 34, will generate no load.

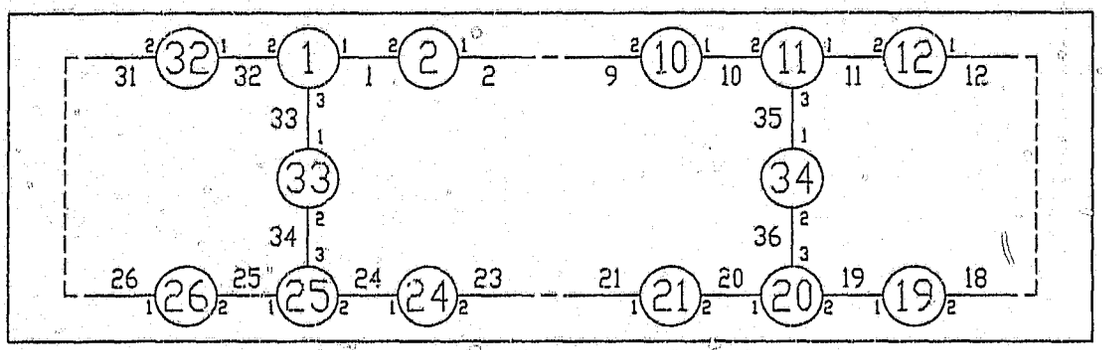


Figure 8.4 Representing buildings on the backbone for Metroring

Non-default parameter settings are:

Load generation method	Unique Station Load
User (MAC) Message size	4079 bytes (exponentially distributed)
Maximum station generated data	200 M bytes
Average frame inter-arrival time	(given in section I.3a)
Simulation ending condition	Simulated time of 5 msec

The simulation ending condition was again simulated time. As in DQDB simulations, it restricted the simulation duration to approximately 1 hour.

The large simulation durations encountered for confidence interval terminations in DQDB and Metroring was due to their protocol model implementations. DQDB and Metroring models used polling events, analogous to the *TransmitMode* event used in FDDI's model (section 6.4.2.3). The main difference is that these polling events were activated for every station on the network every few microseconds of simulated time. This caused the simulation time for DQDB and Metroring to be far larger than for that of FDDI for equivalent simulations.

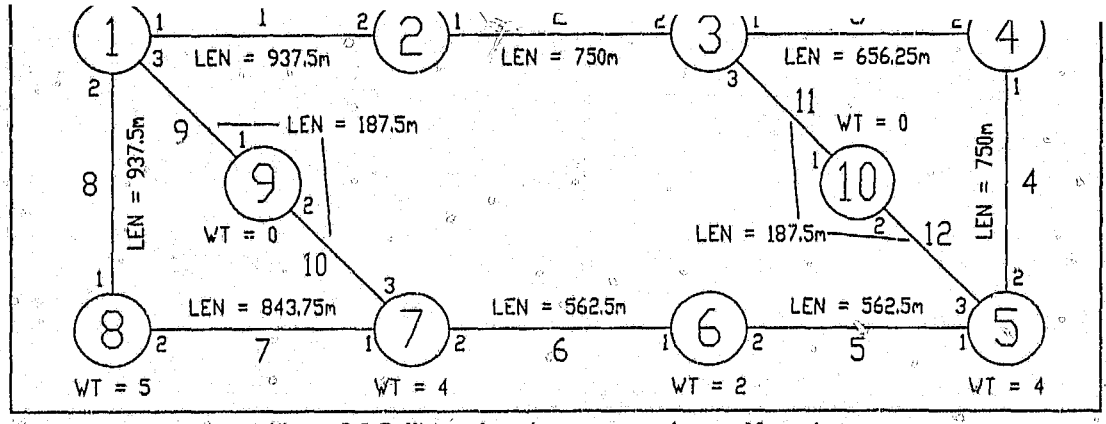


Figure 8.5 Collision domain representation on Metroring

The other parameter settings are:

Load generation method	Unique Station Load
User (MAC) Message size	4079 bytes (exponentially distributed)
Maximum station generated data	10 M bytes
Average frame inter-arrival time	(given in section I.3b)
Simulation ending condition	Simulated time of 5 msec

8.1.2.2 Performing the Simulations

Setting up of the parameters in the simulator should be followed in a particular order for each particular simulation. As all parameters have default settings upon start-up, not all parameters need to be set and thus only the necessary set is given. If the settings for each simulation are saved on disk, one can load these in using step 5 only and then proceeding to step 6. Sequentially they are:

Simulation Options

MAN Type - This is set to the protocol being studied.

Simulation Termination Method - Time Limited Simulation or Confidence Interval Precision Limited methods can be selected. Confidence intervals of 95% are used for terminating FDDI simulations while DQDB and Metroring are time limited (as given in their parameter settings).

Network Load Characterising Method - For the cases where buildings are connected to the backbone, the 'Stochastic - Globally for Network' option is chosen, while for cases where collision domains on the backbone are simulated the 'Stochastic - Unique for each Station' option is used.

Edit N/W Globals

This option brings up a screen referred to as a **value panel**. This simply displays a list of user specifiable values in a menu-like format. Any of the displayed values can be changed by simply pressing the corresponding letter labelling the value and then entering the number or string. This process is terminated by pressing ESC.

Only Station Hardware Latency needs to be set when FDDI is simulated. (DQDB and Metroring are assumed to have no station hardware latency.)

Edit N/W Topology

This allows the user to specify the network's topology in detail. There are two parts to this editor.

Starting the Editor

When option C is chosen for the first time for a simulation, three parameters need specifying:

(1) The number of stations on the network. For FDDI and DQDB this value is either 32 or 8 - depending on whether it is buildings or collision domains on the backbone. With Metroring it becomes either 34 or 10 respectively, as two extra station are required in specifying the two back-up paths (see figure 8.1).

(2) The 'new' station defaults need specifying. First of all the New Link Default Length must be specified for all new generated links' initial lengths. It is best to enter the value which occurs the most in the actual topology specifications and then change the other link lengths to the correct values later.

(3) The default station loading statistics then must be specified using a **Node Load Specification** value panel. The average message inter-arrival time, average message length and maximum amount of generated data are the values in this value panel. Again the set of values which will occur the most frequently for a station on the network should be entered here.

If this editing option is not being chosen for the first time for the simulation, an alternative introduction asks if the user wants to change the number of stations on the network. If the number of stations are increased, then the parameters mentioned above need to be specified for the extra stations added onto the network. Whether stations are added or subtracted from the network, the first and last stations are always linked together by the editor thus this might need respecifying.

Changing Specific Station Characteristics

An editor is then shown, displaying the physical link characteristics for each station. Individual stations can be then be selected using the cursor keys to point to the station and the ENTER key to acknowledge selection. Generally two steps are then taken in changing a station's link characteristics:

(1) On DQDB and Metroring networks more than one outgoing link per station is used. To select which of these links to edit, press the number of the corresponding outgoing port number.

(2) A value panel is then displayed where Station ID, Link Number, Adjacent Station Number and Link Length can be changed. Station ID need not be specified for these simulations, but the other three values should be correctly specified if the defaults generated by MANSIM are not as required.

This editor is ended by pressing ESC.

Edit N/W Loading

This option will induce an individual station load editor if the loading is to be unique for each station on the network. If the network is globally loaded, one Node Load Specification value panel allows the specifications to be set. In the former case, a station will need to be selected using the cursor keys before a Node Load Specification value panel allows a station's load behaviour to be tailored. This editor is also exited by pressing ESC.

Loading Simulation Settings

Option G on the main menu leads to a sub-menu for loading and saving whole sets of simulation parameters. Option D allows simulation settings to be loaded from disk. After selecting this, one might be prompted as to whether current settings are to be overwritten (when they have been recently changed). Press Y to acknowledge that loading is to proceed and then enter the name of the file with the simulation details.

Performing the Simulation

Selecting E brings up the simulation status screen which is displayed for the duration of the simulation. When the simulation ends an alarm sounds periodically. The user then returns to the main menu by pressing any key on the keyboard.

Examining the Results

Option F on the main screen allows one to examine the results.

It is best to save results to file by pressing option A any entering the output file name. The results are stored in ASCII format and can be examined using an editor or even from DOS by entering the command
TYPE <file> | MORE
where <file> is the name of the results file.

A special results viewer can be used by selecting option B. First the general simulation statistics and settings are shown - then screen is passed by pressing any key. The following three screens show consecutively Average End-To-End Delays, Average Queue Delays and Throughput for the network and for each of the stations. The next screen is displayed when ESC is pressed. After all three screens are viewed and after ESC pressed again the user returns to the main menu. Only one station's statistics are shown at a time. To view several station's statistics the cursor keys must be used: UP and DOWN switch between consecutive station statistics displays. PG UP and PG DN switch between station statistics displays in jumps of 5. HOME and END jump respectively to the first or last of the station statistics displays.

8.1.2.3 Results and Conclusions

Section I.4 gives the results of simulations with figures 8.6 and 8.7 showing the results for the two cases: (a) buildings connected to the backbone and (b) collision domains connected to the backbone.

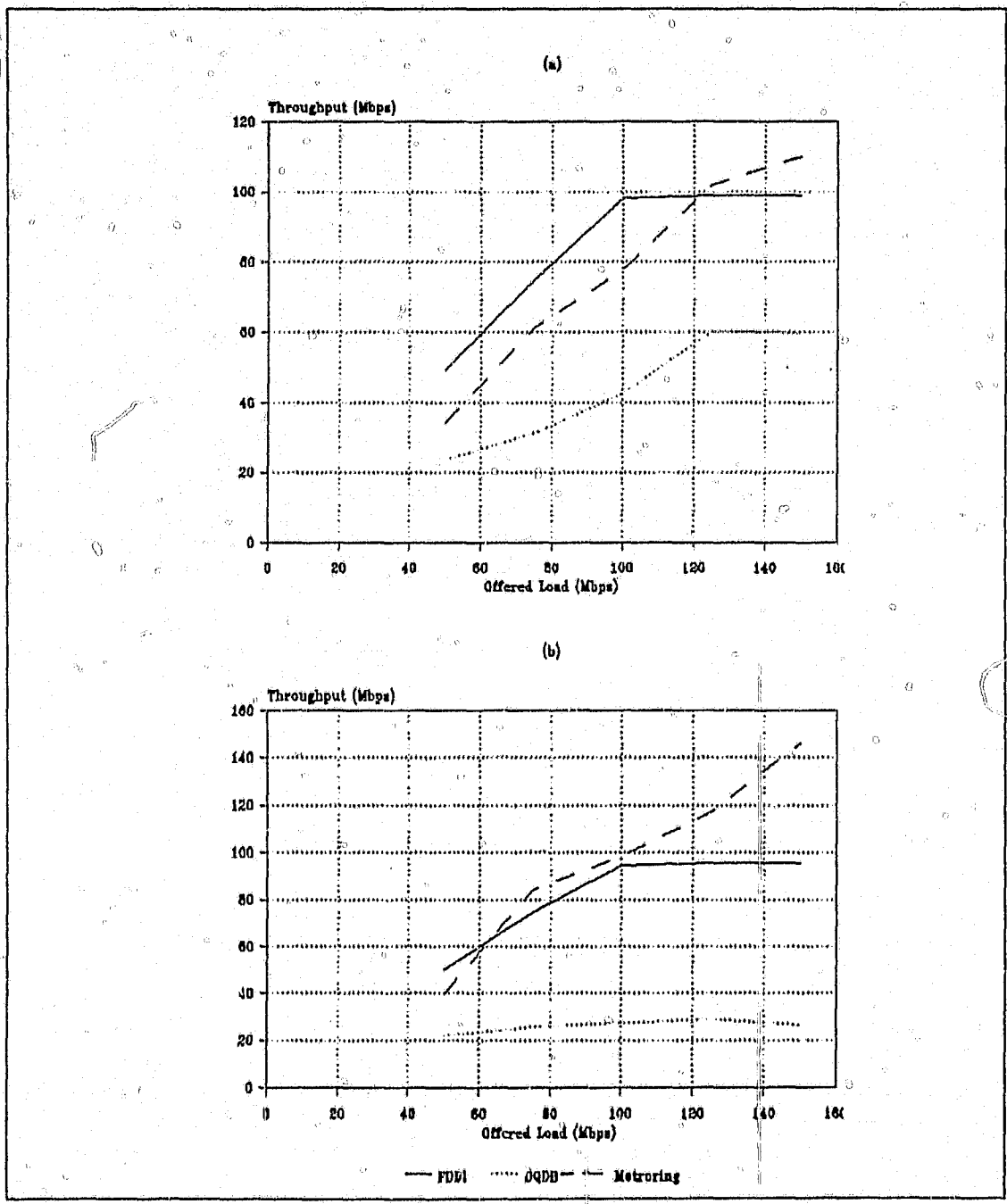


Figure 8.6 User layer throughput vs offered load when using FDDI, DQDB and Metroring for CSIR backbone

The graphs allow easy comparisons as the trends for each type of MAN can be easily observed. Virtually all trends are distinct enough for reliable decisions regarding performance.

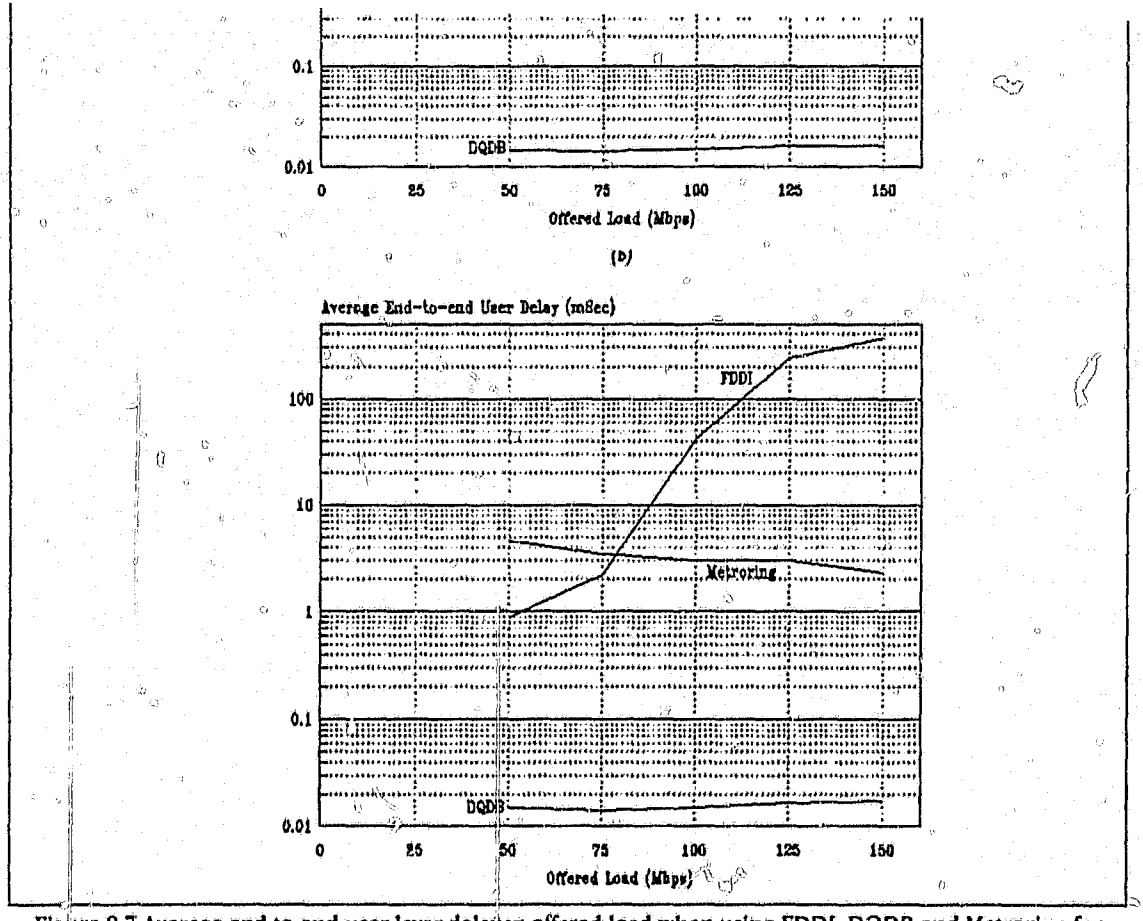


Figure 8.7 Average end-to-end user layer delay vs offered load when using FDDI, DQDB and Metroring for CSIR backbone

The DQDB results have two very opposing characteristics. The delays are extremely small favouring high speed logical connections but the throughput behaviour definitely does not follow the users' requirements ie. offered load. The throughput does not follow the users requirements as at least 50% of slots are not fully used; ie. due to the exponential distribution of the user message lengths many slots are not filled up with user data. Due to the message length distribution, the slots get used very inefficiently for data throughput.

The FDDI delays are very substantial in overload conditions (when offered load is greater than the medium capacity) and so is the saturation of the throughput. Only in underload conditions would FDDI's behaviour be acceptable.

Metroring's delays do not vary greatly as the offered load varies and it offers the necessary bandwidth to allow the throughput to follow the users' demands fairly well.

When considering whether buildings connected to the backbone, or collision domains connected to the backbone are more efficient, significant differences are visible when studying the throughput of the three types of MANs. The most notable difference is in the performance of DQDB's throughput. For the case where buildings are connected to the backbone, the throughput increases steadily as the load increases to 125 Mbps and then levels off. However, when the collision domains are taken to be connected to the MAN, the user layer throughput rises steadily and then drops after the offered load exceeds 125 Mbps.

FDDI, on the other hand, hardly changes its throughput performance in the case for which collision domains are connected to it. Metroring shows a noticed improvement in matching the throughput to the offered load when the collision domains are connected.

As average end-to-end user layer delay is the primary index for determining the MAN best suited for a backbone, the following prioritization of MANs can be used:

(1) DQDB - the lowest average end-to-end user layer delay all round
(2) MR - the next highest delays but also fairly constant
(3) FDDI - after an offered load of about 80 Mbps it has the largest delays which increase to completely unacceptable levels

Although the average end-to-end user layer delays index give a distinct way of selecting MANs, the user layer throughput of each MAN must also be considered. In this case study, consideration of throughput is essential as DQDB performs best when it comes to delays, but worst when it comes to throughput. Further assessment of the protocol performances becomes necessary for determining a viable protocol to solve the problem.

The throughput characteristics of FDDI and Metroring are thus studied. FDDI is observed to perform poorly with respect to user throughput when in overload. Metroring is noted to handle the applied load very well, as the user throughput is almost proportional to the applied load. Thus considering how well protocol handle the throughput, Metroring becomes first choice.

In finally considering Metroring's average end-to-end user layer delays, Metroring can be deemed acceptable, even though its delays are at least 100 times greater than DQDB's (about 30 msec for Metroring, about 15 μsec for DQDB). In this case where Metroring is taken as first choice, it is obvious to configure it as a backbone with connected ethernet collision domains.

(2) Factors affecting protocol selections could be interpreted as indices or quantified, in order to aid evaluation of protocols.

This lead to a number of assumptions. These assumptions were made as they defined the area of research more clearly and eliminated a number of protocol variables, such as internet effects and implementation dependant factors, that would expand the scope of research to unmanageable proportions. These assumptions were also reflected in the design of the MAN simulator where a special version of the protocol stack is used.

Section 1.5.1 identifies the factors affecting performance in all classes of networks. All networks can be described in terms of the same set of basic elements. MAN performance would then be dependant on all of these factors.

In section 3.3 these factors were examined to identify indices, or attributes, that are used to compare different MAN protocols with each other. The indices selected could then be used objectively for determining MAN protocol performance. The assumptions and limitations behind these choices did not restrict the validity of the chosen indices for performance comparisons. Rather, the choices considered a vast scope of factors affecting MANs from a common frame of reference (the assumptions and limitations of the study). Once the three methods of comparison were established, indices for each method of comparison were selected from these chosen factors.

9.1 Unique Features of the Study

The method of comparison encompasses several unique features:

- Several unique methods can be used for comparing protocols against each other.
- Established protocols through to protocols in the conceptual stage of development can be evaluated and compared.
- Any number of protocols can be incorporated into an existing set of evaluated protocols. Any selection of protocols from this large set of evaluated protocols can then be compared against each other.
- This method of comparison applies to MANs. Other comparisons apply, by and large, to LANs and WANs.

9.2 Application of Methods for Solving Problems

Each method of protocol comparison has distinct advantages and disadvantages.

9.2.1 Functional Method of Protocol Comparison

The functional method of protocol comparison allows a choice of a MAN protocol based on how well the compared protocol's meet prioritized functional requirements. The more abstract form of protocol performance is thus analysed by considering the qualitative requirements for the network's performance.

This is the more difficult comparison method of the three available methods to apply to a problem, as the principles behind the criteria here, are subject to evaluation of protocol rules, ie. a qualitative assessment of the protocol is required for every criterion of this comparison method. This comparison method compares protocols relative to one another. Thus, it is important to consider of all the compared protocols simultaneously when evaluating which protocol meets a criterion best.

9.2.2 Analytical Method of Protocol Comparison

The analytical method of protocol comparison allows the choice of a MAN protocol based on protocol performance under a standard network configuration.

The comparison method also enables alteration of the parameters of the standard configuration to suit the problem's requirements as close as possible. The basic behaviour of the physical and MAC layer is represented by idealized throughput measures and physical delays allowing a limited amount of implementation factors to be taken into account. Thus the idealized performance of a MAN is represented to allow several protocol and topology architectures to be effectively compared.

9.2.3 Simulation Method of Protocol Comparison

With the simulation all details of the asynchronous load and the intended configuration allowed the user to examine which protocol meets the network's requirements best.

The physical layer measurements are suited to the designing of the topology of the network while the user layer measurements are suited to the comparison of MAN performances and the planning of the load allocation on the network. This illustrates how practical a tool simulation could be.

Each of these methods of comparison thus have varying levels of practicality and require different depths of analysis. Note that when protocols have been analysed for the functional and analytical methods of comparison, no extensive analysis is required when considering the protocols again in further MAN selection problems. New network problems requiring comparison via simulation, require re-assessment of protocol performance for every protocol considered. The applicability of each method of comparison is summarized in table 9.1.

MAN Comparison Method	Practicality in Solving Protocol Selection Problems	Required Depth of Analysis for New Protocols
Functional	Qualitative and abstract requirement analysis.	Comprehension of rules for all protocols considered.
Analytical	Idealized performance of man protocol and topology architectures.	Analysis and derivation of network topology related indices.
Simulation	Choice of protocol and configuration for specific network problem.	Breakdown of protocol functioning and design of a protocol object.

Table 9.1 Evaluation of resulting MAN comparison methods

The Functional and Analytical methods are thus easiest to apply when the method has evaluated many protocols over the long term.

Each comparison method was applied successfully to the three example protocols. The protocols used in the illustration of the MAN comparison method's application are at different stages of the network protocol development life cycle. This range of maturities of these example protocols shows just how the study of comparison methods can be used in selecting MANs for future requirements. This implies long and short term networking solutions can be reached by using it. Well established technologies as well as newly emerging protocols can be compared against each other using this method, thus achieving the aim of the investigation.

CHAPTER 10: FURTHER RESEARCH

The functionality or procedures for the MAN comparison method may be developed further. Research in several other areas would make the MAN comparison method more general in scope:

10.1 A Detailed Comparison of Reconfigurability and Expansibility of a Topology

The method for functional comparison of MANs requires more research for analysis of the reconfigurability (section 4.3.2) and expansibility (section 4.3.3) of a network topology. The effects of the current configuration of a network should also be taken into account when looking at these criteria. The complexity with which stations are arranged on the network needs to be taken into account, as mentioned in section 4.3.1.

10.2 Indices for Aiding the Functional Method of Protocol Comparison

Section 4.5 proposes an extension to the functional comparison of MANs. If the criteria for choosing the MAN protocol could be weighted, a more rigourous procedure for choosing a MAN in terms of functional requirements will result. This, however, requires each criterion to be represented by an index that measures the absolute effectiveness of a MAN protocol's ability to meet the criterion.

10.3 Faster MANSIM Protocol Model Operation for Simulation

Computer network simulation is a complex topic. The example protocols have been modelled in a straightforward way for application to the simulator environment (section 6.1.2 and 6.1.4). However, the protocol models could be derived differently via other modelling techniques. This can result in more efficient model operation when more current simulation techniques are applied.

10.4 Internetworking of MANs

The interconnection of separate MANs should also be investigated further. This will entail taking internetworking effects into account in the procedure for MAN performance comparison. Evaluation of several internetworking effects is possible:

- ATM switching between MANs. This requires all packets requiring routing to be broken down into fixed length segments (or slots, for DQDB).
- Expansion of MANs to CANs - country-wide area networks. This effect is achieved when Metroring networks are of a scale where several Metroring regions exist.
- Linking MANs, as is required for extensive networks using FDDI and DQDB.

The results of research in this area, may help network developers to direct efforts to the option which will have the best long term benefits to the networking industry.

10.5 Generalization of the Comparison Method for all Classes of Network

The procedure of MAN performance comparison can be generalised in several places to allow the incorporation of LANs and WANs into the MAN comparison method:

- A greater scope of factors affecting network performance can be considered.
- Derivation of a new standard configuration for Analytical comparisons is necessary to model all classes of networks.
- Derivation of a new topology representation is required for Analytical comparisons when the indices are derived.
- The basic environment components and characteristics of the simulator will need to be refined to incorporate any class of network, eg. the rules for topology structuring may need to be variable or the method for transmitting packets may need modifications.

This will make the comparison method applicable to any type of computer network.



APPENDIX A: THE OPEN SYSTEMS INTERCONNECTION (OSI) REFERENCE MODEL

The International Standards Organisation Protocol Reference Model is a seven layered model, each layer representing a software entity that performs a well defined function with respect to the other layers.

A.1 The Physical Layer

This layer comprises the specifications for the mechanical and electrical network interface. The network layer uses the physical layer to transmit packets as serial streams of bits across the network medium.

A.2 The Link Layer

Here the network layer is provided with reliable information transfer facilities. The link layer thus detects errors in transmissions and if necessary will retransmit packets. Two ways of accomplishing this task (services) are provided:

Connectionless Service: Each information frame is treated as self contained and is transferred using a best-try approach.

Connection Orientated Service: Groups of information frames are transmitted over a logical connection - a connection that exists by the source and destination stations knowing the state of the information transfer mode. This therefore endeavours to provide error free information transfer in a more controlled way.

A.3 The Network Layer

Addressing of packets for routing across network wide connections is performed here. It also controls the establishing and clearing of connections between transport layer entities. Sometimes flow control is performed here too.

A.4 The Transport Layer

This layer shields the higher application layers from the three lower network oriented layers by making sure all error and flow control terminates here. These services are provided in five classes of service to cater for the varying quality of service provided by different network types. Class 0 would provide only the basic functions needed for connection establishment and data transfer. This would range up to Class 5 that provides these functions with full error and flow control.

A.5 The Session Layer

This introduces dialogue synchronization points in the communication as well as a point where exception reporting occurs (non-recoverable exceptions during a transaction). The presentation layer thus uses the session layer to set up and clear dialogue channels.

A.6 The Presentation Layer

Here the syntax of the data (abstract data syntax) from the application layer is converted to a syntax common (transfer syntax) to both application entities. A number of transfer syntaxes exist for this purpose and the corresponding presentation entities must somehow agree on the syntax to be used. Data security issues would also be resolved in this layer.

A.7 The Application Layer

At the top of the OSI stack the application layer provides the user (interface) to a number of network-wide distributed information services. Examples of these information services are: file transfer, access and management (FTAM), electronic mail and virtual terminal emulation. These services are then linked to the station's operating system (OS) and operation of the communication sub-system is transparent to the user.

APPENDIX B: THE FIBRE DISTRIBUTED DATA INTERFACE STANDARD (FDDI)

The component entities of the FDDI standard are re-specified with respect to the bottom two layers of the ISO reference model. Component entities of a FDDI station, as shown in figure B1, are:-

Station Management (SMT) - The local part of the network management process. Control is exercised from here for the proper initialization and operation of the FDDI station.

Logical Link Control (LLC) - This is a standard component which ensures reliable transfer of data across the data link. The LLC does not require redefining as it is a standard protocol (IEEE 802.2) which interacts with the MAC.

Medium Access Control (MAC) - Specifies the lower sublayer of the data link layer. Here medium accessing, addressing, data checking and data framing are performed.

Physical Layer Protocol (PHY) - Specifies the upper sublayer of the physical layer. Clocking, encoding/ decoding and framing for transmission rules are specified here.

Physical Layer Medium Dependant (PMD) - Specifies the lower sublayer of the physical layer. This includes specifications for power levels, optical transmitter/ receiver characteristics, interface optical signal requirements and permissible bit error rates.

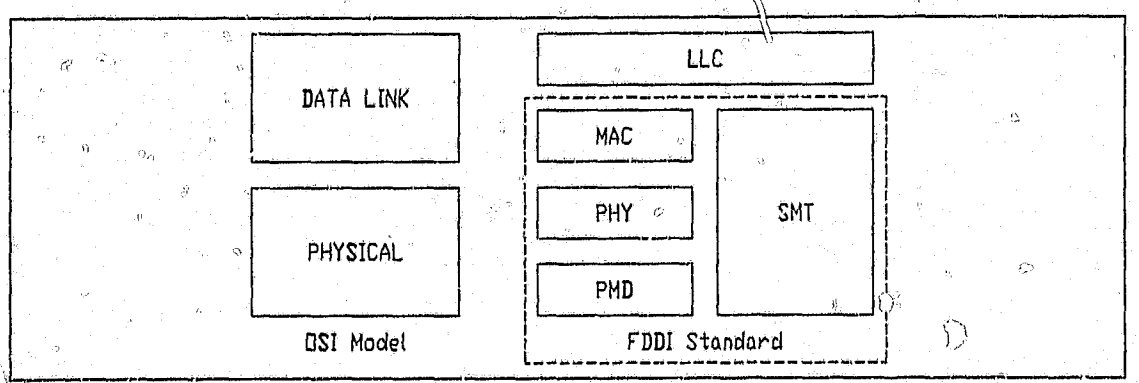


Figure B1 Component entities of a FDDI station

The rules of the protocol are classified into different areas:

B.1 Physical Layer

B.1.1 Classes of Stations

- Dual attachment station: Contains two PHY entities. It may have more than one MAC entity. The station can be attached directly to the trunk ring or indirectly by means of a concentrator.

- Single attachment station: Contains one PHY and one MAC entity and may only be attached to a concentrator.

- Concentrator: This is a dual or single attachment station with extra PHY's connected in a manner to allow other stations to be connected to it in order to be attached to the main ring. Concentrators come with varying levels of functionality.

B.1.2 Topology and Structure

A trunk ring consisting of two counter rotating rings. A maximum of 500 stations may be attached to the ring. The ring stations may be linked by a maximum of 100km of duplex fibre-optic cable.

B.1.3 Clock Rates, Encoding and Decoding

A baud rate of 125 Mbps provides a 100 Mbps data rate using *4B/5B symbol Manchester encoding*. The 4B/5B encoding uses 32 symbols, each symbol represented by 5 bits. 16 of the 32 symbols from the four out of five group code are used as data symbols, the rest of the symbols are used for line state signalling, control indication, frame field delimiting and transmission error indication (packet errors).

Each station uses a *local fixed-frequency clock* with 0,005% stability to transmit. Frames of up to 4500 octets can be transmitted not to exceed limits of (at least) a *10 bit elasticity buffer* in the station.

Each receiver compensates for the difference in received and local frequencies with the variable length elasticity buffers. Receivers use a technique such as *phase-locked loop* to lock on to the incoming frequency. This compensation is necessary to keep the ring latency constant (a changing ring latency causes errors in transmissions on the ring). Compensation is achieved by removing or adding IDLE symbols from the preamble.

As the action of this compensation could lead to the shortening of the preamble (which was used to synchronize the clock in the receiver), a further mechanism called a *smoothing buffer function* is needed to prevent the preamble from disappearing from the front of the frame. This smoothing buffer function removes IDLE symbols from preambles longer than the symbol threshold (16 symbols) and redistributes these symbols to lengthen short preambles, in order that the preamble length is never shorter than 4 or 5 octets. The algorithm used for the smoothing function reduces frame loss probability to less than 10^{-12}.

B.1.4 Frames and Their Formats

In FDDI a token ring like access protocol is used, therefore an access token and a data frame are used.

Fields common to both the token and information frames are:

PA - Preamble (16 symbols)
This consists of IDLE line state symbols and is used to synchronize the receiver's clock.

SD - Starting Delimiter (2 symbols)
This consists of the J and K symbols and is used in signifying the begin of the data fields.

FC - Frame Control (2 symbols)
This is used for signifying the type of frame and con functions associated with it.

ED - Ending Delimiter (1 or 2 symbols)
This is comprised of one or two T symbols and signi the end of the data fields.

B.1.4.1 Token Format

Two types of token are used in FDDI: restricted and non-restricted. A non-restricted token is used during normal ring operation while a restricted token grants only two stations control

of the asynchronous frame service. Dialogue then occurs between only these two stations to facilitate high speed transfer of burst data (eg. between harddisks on the communicating stations). Figure B2 shows the layout of the fields in a token.

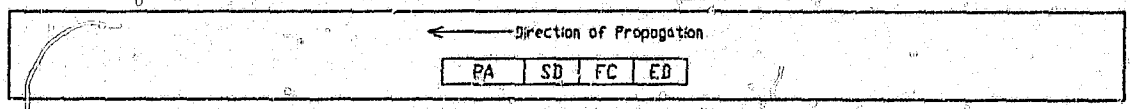


Figure B2 Layout of fields of a FDDI token

B.1.4.2 Data Frame Format

The data frame, as shown in figure B3, has several more fields than the token frame:

DA - Destination Address (4 or 12 symbols)
The MAC destination station address for this frame. The FC field distinguishes the length for the address.

SA - Source Address (4 or 12 symbols)
The MAC source station address that originated this frame. The FC field distinguishes the length for the address.

INFO - Information (0 or more symbol pairs)
This variable length field carries all data for protocol layers above the MAC.

FCS - Frame Check Sequence (4 symbols)
For checking at a destination node whether the data in the frame is corrupted or not.

FS - Frame Status (2 symbols)
This consists of three control indicators for: indicating whether the frame is error free or not, whether the destination station recognized the frame or not and whether the destination station copied the frame or not.

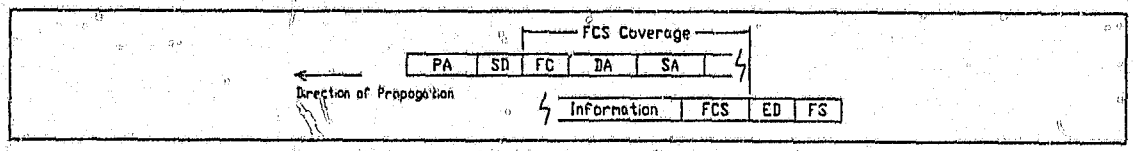


Figure B3 Layout of fields of an FDDI information frame

B.2 Medium Access Control Layer

The entity in the MAC layer decides which station has control of the medium. It schedules and performs all data transfers on the ring.

B.2.1 Access Control

The Timed Token Rotation (TTR) protocol is used to control access to the medium. This protocol allows two classes of service:

- Synchronous service: Use of the token is made whenever synchronous frames are ready for transmission.
- Asynchronous service: The token can be used to transmit asynchronous frames whenever they are available *and* time is permitting.

To keep access delays around the ring to a minimum, all stations refer to a time called the Target Token Rotation Time (TTRT) which specifies the maximum time a token can take to rotate the ring once. This maximum value of token rotation time (TRT) ensures each station sees the token on average once every *TTRT* seconds for synchronous service requirements to be met. The value of TTRT for the whole ring is negotiated during the claim process (section B.2.4.1) when the ring is initialized.

Three counters in each station are also used for controlling the operation of the protocol:
- Token Rotation Timer (TRT): This countdown timer measures the time between the receipt of tokens at the station. TRT is said to expire when it reaches a value that exceeds the TTRT (the TRT countdown timer reaches zero). When it expires it will restart the timing (counting down from *TTRT*) and continue until the token is received again. During normal operation, it is restarted (loaded with the TTRT value) whenever the token is seen passing the station.
- Token Holding Timer (THT): This holds the time for which the station may transmit asynchronous frames. This time is derived from the remaining time in the TRT when the token arrives early ie. TRT has not expired yet.
- Late Counter (LC): This is incremented whenever the TRT expires and the token has not yet been claimed by the station. The LC is decremented when the token eventually does arrive. It is used to indicate how many *TTRT* cycles the token is behind. When tokens do arrive and the LC does not equal zero, no asynchronous data transmission may take place.

When a station MAC entity does not have possession of the token, the MAC must:
- Repeat any passing frames from its upstream neighbour to its downstream neighbour
- Copy the frame to a local buffer if the frame's destination address is the same as the station's MAC address and subsequently notify the LLC or SMT.
- Notify the LLC or SMT of the received frame's receipt if no error in the received frame is indicated or detected
- Change the symbols in the frame status field if (1) an error in a passing frame is detected, (2) the MAC recognizes its own address as the destination address of the frame's or (3) the frame is copied to the MAC buffer.

A station's MAC takes possession of the token when it has frames in the LLC to transmit and it receives the token from its upstream neighbour. When possession of the token is taken, a station does not remove the whole token from the ring but transmits the succeeding fields of the information frame as the header fields of the token pass by. When the header of the token is seen the TRT is restarted (except in the case of a late token).

When possession of the token is taken, the MAC entity must:
- Transmit frames from the LLC awaiting transmission.
- Remove frames from the transmit buffer when it determines they were received correctly (after they are received with the copy flag in the frame status field set).
- Stop transmitting frames when either (1) the measured time since the last received token (TRT) exceeds the TTRT or (2) there are no more frames in the LLC awaiting transmission. The station must then retransmit the token right after the last transmitted frame.
- Remove its own frames from the ring once they have circulated. (This occurs whether frames have been received correctly or not.)

Synchronous frames are transmitted whenever they are ready as the token passes by. Asynchronous frames are transmitted only after any synchronous frames have been transmitted. The time allowed for continuous asynchronous frame transmission is held in the THT counter and set equal to the value in the TRT when the token is captured and before

the TRT is restarted. If the TRT expires before the token is received, the LC is incremented to remember this so that when the token comes round the next time the TRT is not restarted; this allows the node to accumulate the lateness by not resetting the TRT.

The amount of time for which a station may transmit synchronous frames is controlled by station management (SMT) in a bandwidth-allocation process. After this process' completion each station requiring synchronous frame bandwidth will have an assigned period of time for this purpose. This bandwidth is allocated in such a way such that the sum of all the synchronous transmission times for all stations on the ring does not exceed the TTRT but still allows time for asynchronous frame transmission. This ensures a worst case delay of the token of twice the negotiated TTRT.

B.2.2 Priority Control

Priority control is implemented for asynchronous frames in eight priority classes. Each class of priority has an associated time value for which it may transmit asynchronous frames. Whenever the time remaining in the THT is greater than the associated transmission time for a frame's priority, the frame can be transmitted. Frames with higher classes of priority take preference over frames with lower ones.

Frames in the MAC queue are processed successively according to this rule. The effect is that a mix of priorities (from a queue filled with frames of all priorities) is transmitted with the scope of frame priorities narrowing over time until *THT* equals zero. Obviously, if frames of the highest priorities do not exist while the THT value will only permit their transmission, available frames with lower priorities will be transmitted.

B.2.3 Fairness Control

No specific fairness mechanism is implemented in FDDI as fairness is inherent in the logical MAC operation. (Each station has a chance to transmit frames for each rotation of the token.) This inherent fairness also applies to the priority operation of FDDI [Halsall 88].

B.2.4 Station Initialization

When a FDDI ring is powered up, it first assesses whether links to adjacent stations and the stations themselves are functioning correctly. Secondly a higher layered station-connection management protocol takes action by controlling internal connections within a station and switches the MAC entity into the ring. When a station is connected to its neighbouring stations via this action, it is referred to as being in the "through mode".

When all the stations are eventually connected as one full ring, one MAC will detect the need for ring initialization. When a station MAC detects the need to initiate the ring, it enters the Claim process to start token flow and TTRT negotiation.

B.2.4.1 Claim Process

One station initiates this process by transmitting a claim frame. This frame has the source and destination addresses set to the initiating station's address while it also contains a bidding value for the TTRT.

All other MAC nodes continually process incoming claim frames, comparing the incoming TTRT bid with its own TTRT bidding value. Whenever a higher bid value (the faster the required TTRT, the higher the bid value) is read from a claim frame a station defers from transmitting its claim frames if it was originally transmitting. Whenever a station reads a lower bid value than its own, it strips the frame with the lower bid value from the ring and transmits claim frames with its own bid value even if it was not originally in the claim process. Thus all stations that have bid values higher than the claim process initiator will be the only stations transmitting claim frames. Every station then remembers the last highest bid value and will deduce the TTRT from this value. This claim process then completes when the station with the highest claim value receives it own claim frame back. This station thus wins the claim process and will issue a new token. (If two stations request the same TTRT, the length and magnitude of the two station's addresses decide who wins the claim process.)

B.2.5 Soft Error Recovery

Several types of soft errors must be catered for:

B.2.5.1 Bit Error During Frame Transmission

Whenever an error occurs in the transmission or delivery of a frame, the frame is known to be corrupted. Any station which such a frame passes detects this error and sets an error bit in the Frame Status field of the frame; thus when the frame's source station removes it from the ring, it itself will detect this error by the status of the frame error bit in the Frame Status field. The frame will thus need to be retransmitted.

B.2.5.2 Bit Error During Token Transmission

If a token becomes corrupted during its rotation of the ring, the station detecting the error will initiate the claim process. This will then determine which station will issue a new token.

B.2.5.3 Full MAC Receive Buffer

When a MAC entity recognizes a frame from the ring as destined for itself but has a full receive buffer it cannot copy the frame. However two control indicator symbols in the frame status field at the end of the frame indicate whether (1) the destination station recognised it own address and (2) whether the destination station copied the frame. Thus the source station can assess whether its returned frame was copied or not, and if the destination station was functional.

B.2.6 Hard Error Recovery

Hard errors are handled at two levels in FDDI. The management entities try and correct the problem, while the MAC layer goes into a Beacon mode to detect when the ring error is corrected.

B.2.6.1 Station or Link Failure Management

The physical-connection management protocol continuously monitors links between stations to see if they are functioning correctly.

When a station or link fails, the two stations adjacent to the failed component reconfigure internally (according to the station-connection management protocol) to link the backup ring into the functioning part of the ring. This is shown in figure B4.

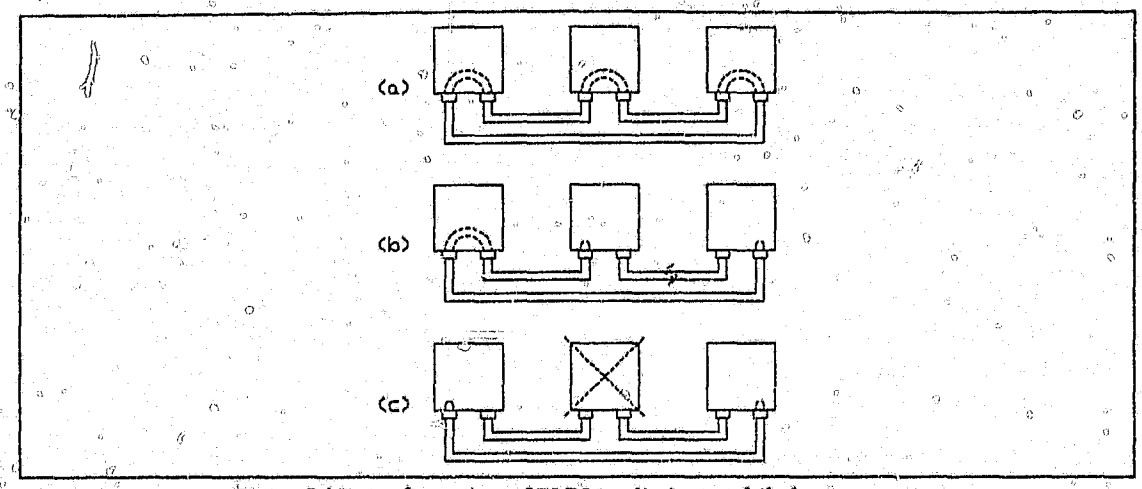


Figure B4 Reconfiguration of FDDI to eliminate a failed component

An optical switch within each station allows stations to pass the signal through it without being received or transmitted by the MAC. This by-pass mechanism is used by the station when it is powered down or an internal error occurs (as the network management function would detect).

B.2.6.2 Beacon Process

The MAC uses the Beacon process when claim token fails, when the ring is first initialized or any time the ring has been broken. This process is used to test the ring integrity at the MAC level to see if the logical ring is functioning correctly.

In the Beacon process, initially stations continually transmit beacon frames. When a station receives a beacon frame from its upstream neighbour, it stops transmitting its own beacon frames and passes the upstream beacon frames. Thus when a station receives a beacon frame that it originally transmitted, it knows that the ring has been established or restored and will then enter the claim token process to initialize the ring.

APPENDIX C: THE DISTRIBUTED QUEUE DUAL BUS STANDARD (DQDB)

The components of the DQDB standard are re-specified with respect to the bottom two layers (Physical and Data Link) of the ISO reference model. The component entities of a DQDB station, as shown in figure C1, are:-

Station Management - The local part of the asynchronous network management process. Control is exercised from here for the proper initialization and operation of the DQDB station.

Isochronous Management - The part of the network management process controlling the operation of the isochronous service.

Logical Link Control (LLC) - This is a standard component which ensures reliable transfer of data across the data link. The LLC does not require redefining as it is a standard protocol (IEEE 802.2) which interacts with the MAC.

Medium Access Control (MAC) - Specifies the lower sublayer of the data link layer. Here medium accessing, addressing, data checking and data framing are performed.

Physical Protocol - Specifies the upper sublayer of the physical layer. Clocking, encoding/ decoding and framing for transmission rules are specified here.

Physical Medium Dependant - Specifies the lower sublayer of the physical layer. This includes specifications for power levels, optical transmitter/ receiver characteristics, interface optical signal requirements and permissible bit error rates.

Isochronous Channel Control - This component will link the network data service to the devices that process the data (eg. telephones, fax machines).

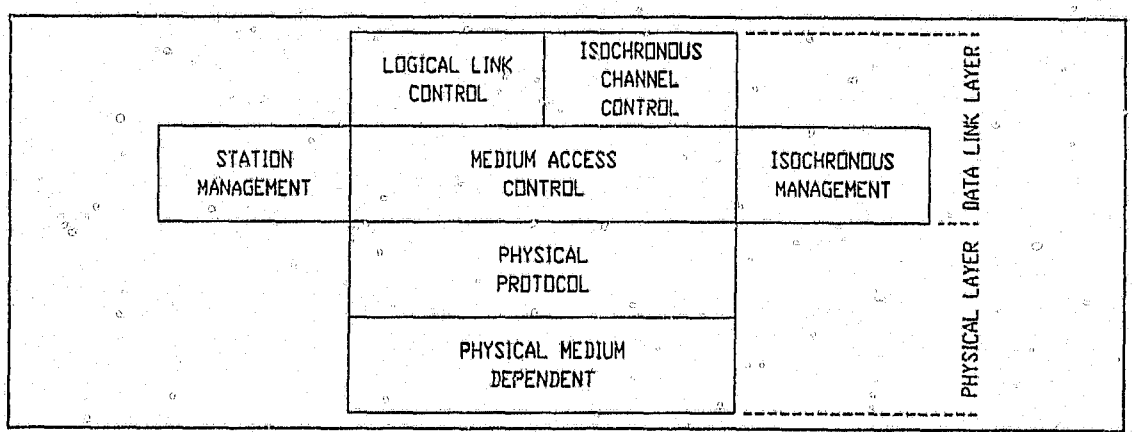


Figure C1 Component entities of a DQDB station

The rules of the protocol are classified into different areas:

C.1 Physical Layer

C.1.1 Components of Network

Four type of stations (besides the medium) constitute the network (three types of stations shown in figure C2a):

- Head of Bus (HOB): This is a DTE (data terminal equipment) performing a frame generation function. The frame generation component generates the synchronized and formatted time structure of slots that each node may use for information transfer.
- Node: This is a DQDB DTE performing no specialized function (although for ring structured networks every DTE must be capable of being a HOB or EOB).
- Erasor Node: These are specialized stations which recognize slots which have passed their destination stations and release them (transmit empty slots in their place) for subsequent use [Garret et al 91].
- End of Bus (EOB): This is simply a DTE acting as a sink to the information flow on a bus.

HOB's and EOB's are found together in pairs at the ends of the busses such that one HOB has a complementary EOB at the end of its line.

C.1.2 Topology and Structure

The topology is a dual bus consisting of two contra-directional lines as seen in figure C2a. Stations on each of the dual busses are linked by a maximum of 50km of fibre-optic cable.

Nodes lie adjacent to the medium in that data does not flow through them. Instead each node uses unidirectional reading and writing taps and writes information using logical OR gates, as shown in figure C2b, onto a synchronized and formatted time structure generated by the frame generator at the head of each bus.

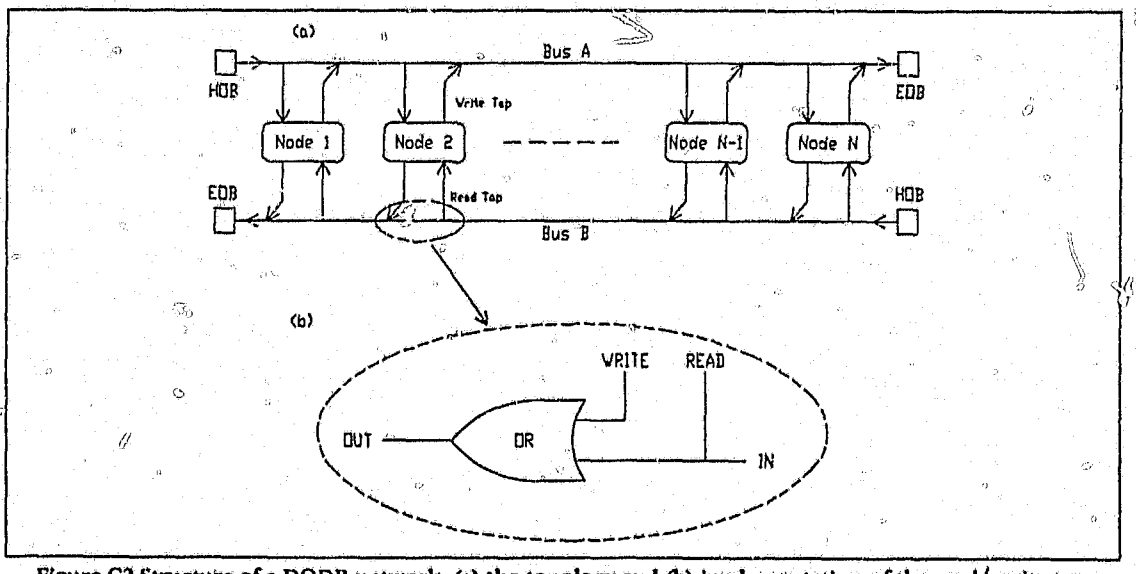


Figure C2 Structure of a DQDB network, (a) the topology and (b) implementation of the read/ write tap.

C.1.3 Clocking and Encoding/Decoding

A 150 Mbps data rate is used on each bus. This dual bus operates with any synchronous transmission interface.

The empty slots used for carrying the information are generated continuously by the HOB at the same rate but not always the same phase.

C.1.4 Frames and Their Formats

Slots are used to convey data in DQDB. The HOB generates empty slots, onto which segmented frames from the LLC layer are written. Each slot carries information for one bus' user data (in its forward direction) and the other bus' requests (in the reverse direction).

C.1.4.1 Slot Format

These slots are composed of many fields that add up to 53 octets in total. The structure of the slot is shown in figure C3.

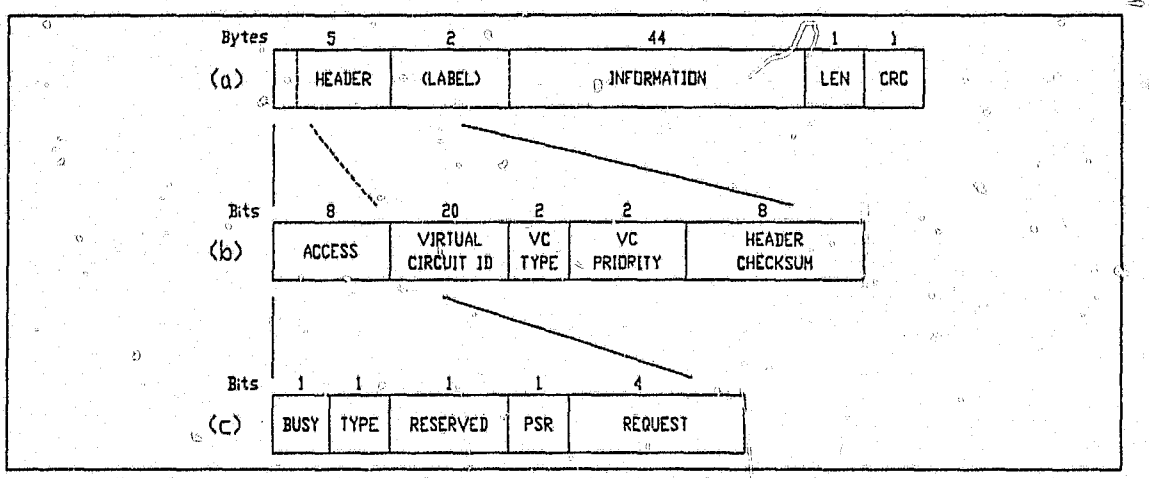


Figure C3 DQDB slot structure showing fields of the (a) slot, (b) slot header and (c) the access control field in the header.

Header field: This is a section of the slot with control information.

Access Control Field: This contains 8 bits, each of which is used differently.

Busy Bit: Indicates whether this slot is presently being used (by an upstream node).

Type Bit: Indicates whether the slot is used for isochronous or asynchronous data.

Previous Slot Received Bit (PSR): This is used for acknowledging the receipt of the previous slot. This is implemented so because a node cannot immediately acknowledge receipt of a slot as it passes. (It has control of only one bit at a time.) This bit is used when an erasor node needs to know when to transmit an empty slot instead of a busy slot.

Request Bits: These four bits are used by stations for reserving slots for four different priorities.

Reserved Bit: For future use.

Virtual Circuit ID: Used for identifying which two nodes access this slot - hence 'who belongs to this virtual connection'.

Header Checksum: For verifying the validity of the whole header.

Information field: For isochronous slots all 48 octets in this field are used for data. Each octet then represents one channel which will be repeated every 125μs. For asynchronous slots this information field is broken into sections:

Label: These two octets contain a value unique over the DQDB bus used for guiding reassembly of segments at the destination.

Information: 44 bytes for a segment are used for carrying information.

Length and Cyclic Redundancy Check (CRC): These two fields take up two octets together and are used by the MAC layer for error checking.

C.2 Medium Access Control Layer

C.2.1 Access Control

Use of the distributed queue protocol is made to control each station's use of the medium bandwidth in terms of slots. The principle to be presented applies to both busses but will only be described for one bus.

Although the distributed queue protocol uses one bus for information transfer in one direction (the forward bus), the other bus (the reverse bus) is used for the forward bus' control purposes. Thus when considered fully, each bus carries user information for its forward path and control information for the other bus - this control information being only four REQUEST bits in the access control field of a slot. The simple case of one priority class is considered first.

For one node, three entities participate in the distributed queue protocol:
- A local queue: This is a buffer in which segments to be transmitted are queued and served on a 'first come first serve' (FCFS) basis. (A segment is a fixed length piece of data that will fill one slot while many segments make up a message.)
- A REQUEST counter: This is a counter for keeping track of the number of downstream nodes that have requested slots.
- A COUNT_DOWN counter: This counter is used to keep track of how many empty slots to allow to pass on the forward bus, before using one for its own data transmission.

A station can be in one of two states:
- IDLE: The local queue is empty (no segments need to be transmitted) but the node monitors the state of the distributed queue (the apparent queue caused by network node requests).
- COUNT_DOWN: The local queue is holding at least one segment and has requested an empty slot but now waits for downstream nodes to be served first.

In the IDLE state:

- If a request arrives on the reverse bus, the REQUEST counter is incremented by one.
- If an empty slot arrives on the forward bus, the REQUEST counter is decremented by one if it is not zero.
- If a segment arrives at the local queue for this node, the queue goes to the COUNT_DOWN state. The COUNT_DOWN counter takes on the value present in the REQUEST counter at this point and the REQUEST counter resets to zero.

Only one segment is serviced at a time in the COUNT_DOWN state:

- If a request arrives on the reverse bus, the REQUEST counter is increased by one.
- If an empty slot arrives on the forward bus and the COUNT_DOWN counter is greater than zero, the slot is allowed to pass while COUNT_DOWN is decremented by one. The REQUEST counter is unaffected.
- If an empty slot arrives on the forward bus and the COUNT_DOWN counter is zero, the slot is used to transmit the waiting segment. The queue status then returns to the IDLE state.

C.2.2 Priority Control

Priority is implemented with asynchronous frames in four classes of priority (0 to 3). Frames with highest priority (3) take preference over other frames. The frames in each priority class are broken down into segments and inserted into corresponding queues for each priority. Each bus is serviced independently, thus each node has eight queues for handling prioritized segments.

There are also eight sets of REQUEST and COUNT_DOWN counters for each priority. Whenever a node requests an empty slot for a segment of a certain priority, the proper request bit in the access control field of the slots on the reverse bus are set. The COUNT_DOWN and REQUEST counters for the priority control the media access for that priority of segments.

With respect to each other, these priority queues have a few differences in operational characteristics when compared to the media access method for one priority as described in section C.2.1:

- In the IDLE state when a priority n request arrives on the reverse bus, all REQUEST counters for priority n queues and less in the IDLE state are incremented by one.
- If in the COUNT_DOWN state a priority n request is received on the reverse bus with an m priority queue waiting to be served the COUNT_DOWN counter for priority m will be increased if $n > m$.
- If in the COUNT_DOWN state for a priority n queue, an empty slot is received, and no other higher priority queue is in the COUNT_DOWN state, the priority n queue uses the slot.

C.2.3 Fairness Control

Unfairness is imminent in this protocol in a large network due to propagation delays. Two effects are noted:

- Hogging by upstream stations: Stations near the HOB gain unfair advantage by using free slots not knowing there are requests on the way from downstream stations.
- Stale requests effect: Stale requests are requests from downstream stations that are granted before the request reaches the EOB on its bus. They cause upstream nodes on a bus to pass empty slots unnecessarily.

This unfairness will clearly seen in asymptotic conditions (when each station is trying to seize all the medium capacity) in the network [Conti et al 91a]. A number of mechanisms were proposed to counter this "unfairness" of operation and smooth the distribution of bandwidth [Zukerman 90]:

Bandwidth Balancing (BWB): After using empty slots for a certain number of times (BWB_MOD), a station updates its counters if it had received a priority 3 request. Thus the station allows a free slot through for every BWB_MOD slots used.

Access Protection and Priority Control (APPC): Here bounds are set on the request counter values to guarantee bandwidth balancing. These bounds would be a function of the node's position on the bus.

Cyclic Request Control (CYREC): Here control is exercised over the number of requests sent from each station per cycle. These cycles would be controlled by the HOB which would generate resets at the beginning of each cycle.

C.2.4 Soft Error Recovery

C.2.4.1 Bit Error During Transmission

Whenever an error occurs in the transmission or delivery of a frame, the frame is known to be corrupted. The slot's header field checksum and the information field's cyclic redundancy check are used by the MAC to detect when slots contain errors. Actual error recovery must then be performed by higher layer protocols.

C.2.5 Hard Error Recovery

C.2.5.1 Station or Link Failure

When a link fails, one or both of the stations adjacent to the failure detect from a lack of slot receipts that a link has failed. Such a failure can only be catered for in DQDB networks configured in ring topologies.

When configured as a ring, two adjacent stations perform both HOB and EOB functions for both busses and the physical links between such the two stations are not used. This unused connection between two stations is referred to as the logical break in the ring. In the case of a physical break (failure) in a link, the simple solution is to shift the HOB and EOB adjacent stations so that the physical failed links correspond to the logical break thus still making the network operational.

A by-pass bus within each station allows stations to pass the signal through it without being read or overwritten. This by-pass mechanism is used by the station when it is powered down, or an internal station error occurs. This by-pass mechanism is controlled by hardware input-to-output coincidence checking - meaning when a station is not transmitting the input to the station must match the output.

APPENDIX D: THE METRORING NETWORK (MR)

Metroring is characterized by three layers with a management entity as shown in figure D1:

Path Layer - This layer incorporates the Data Link and Network layers of the OSI reference model and also includes the MAC sublayer. The OSI Transport layer's routing function is also incorporated here. Four sublayers are found in the Path layer (to be described later): MRMAC (Metroring MAC layer with same function as the conventional MAC layer), LC (link control), RCP (routing control protocol) and the MP (management protocol).

Physical Layer - Specifies the physical layer as found in the ISO reference model.

Station and Ring Management - The local part of the network management process. Control is exercised from here for the proper initialization and operation of the MR station. This entity also manages the routing tables used in each station.

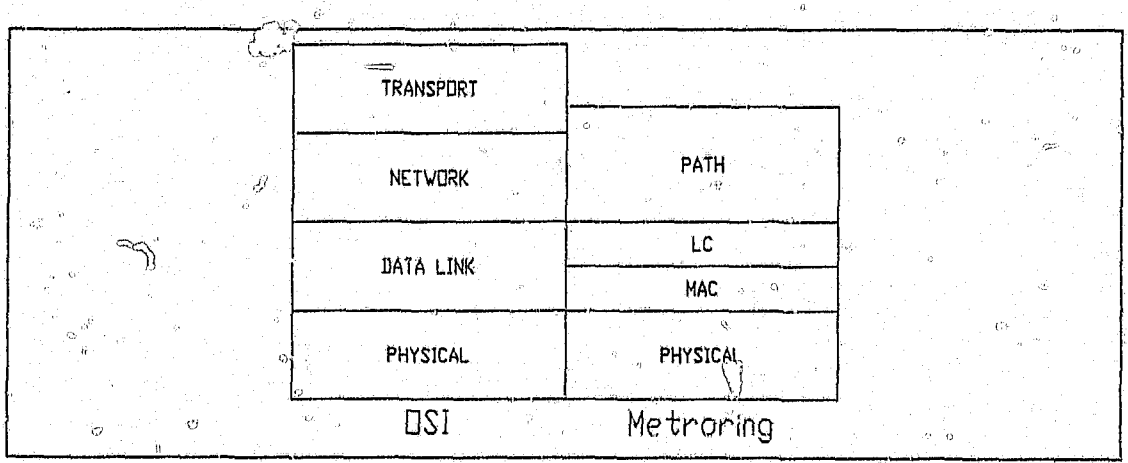


Figure D1 The component entities of Metroring

The Metroring specification can be found in two forms. These forms include all the above mentioned components with the class of physical and MRMAC layers distinguishing them. MRMAC specifies the lower sublayer of the data link layer where medium accessing, addressing, data checking and data framing are performed. One form can contain currently available standards such as Ethernet for the MRMAC and physical layers while the other would have the MRMAC and physical layer redefined to the new level of functionality required by Metroring. (The unique Metroring MAC will not have to deal with contention issues and similar problems.)

D.1 Physical Layer

D.1.1 Topology and Structure

The nodes are connected together with point-to-point dual contra-directional cables collectively referred to as a link. Each link is connected to an input/ output port with each node requiring connection at at least two pairs of input/ output ports to be incorporated into a fully functional Metroring network.

Nodes are connected into rings and the rings are then interconnected to form a region - which is the level of complexity to which MR is defined to operate (for limiting management protocol complexity). Regions may contain a maximum of 4294 nodes. But to explain the ring interconnection better, rings are constructed from at least two Arc-shaped segments. These ring segments, which are uniquely identified in the addressing scheme, are then used to construct interconnected rings as shown in figure D2.

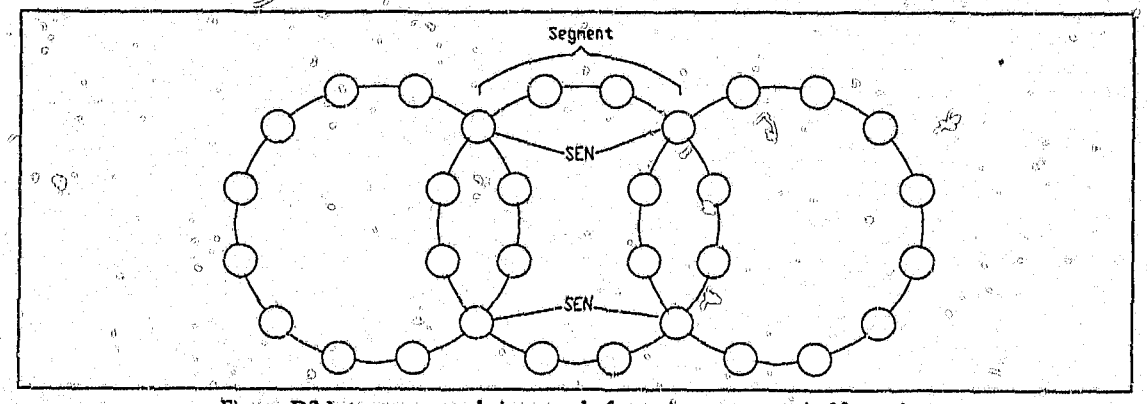


Figure D2 Interconnected rings made from ring segments in Metroring

D.1.2 Classes of Stations

As each piece of equipment connected to the network requires at least two pairs of input/ output ports and some equipment will need to perform special functions, several classes of stations are defined for MR.

Two Port Node - Also referred to as a node, this is a DTE (Data Terminal Equipment) with 2 sets of input/ output ports and at least one application program running.

Segment End Node (SEN) - This is a DTE running at least one application and having at least 3 and a maximum of 16 sets of input/ output ports. Its special purpose is to route received data to adjacent nodes or segments.

Bridge Node - This is DTE with four sets of input/ output ports. It is used for demarcating MR regions and is connected to segments on the regions it bridges.

D.1.3 Clock Rates, Encoding and Decoding

Each link in a MR network operates independently of the others thus boosting overall throughput well above the individual link capacities. On dedicated links, such as used here, transmission of sequential packets is hindered only by the bit corruption of packets thus implying a large degree of functionality (like resolving resource contention) that would be found in implementations such as Ethernet would be eliminated.

The data transfer rate that will be used is 100 Mbps. On the first implementations of Metroring, Ethernet technology will be used for uni-directional cables between stations. At a later stage a Metroring MAC will be used that will use Ethernet encoding schemes but no medium access functions. This later MRMAC will use Manchester encoding and 4B/5B encoding, as in FDDI, meaning the clock rate will have to be set at 125 MHz.

D.1.4 Frames and Their Formats

Two types of frames are used in MR: Information and Management. Both Frames share a common header segment with an information segment following that determines the type of frame, as seen in figure D3. Information and management packets can flow in both directions on a link between stations.

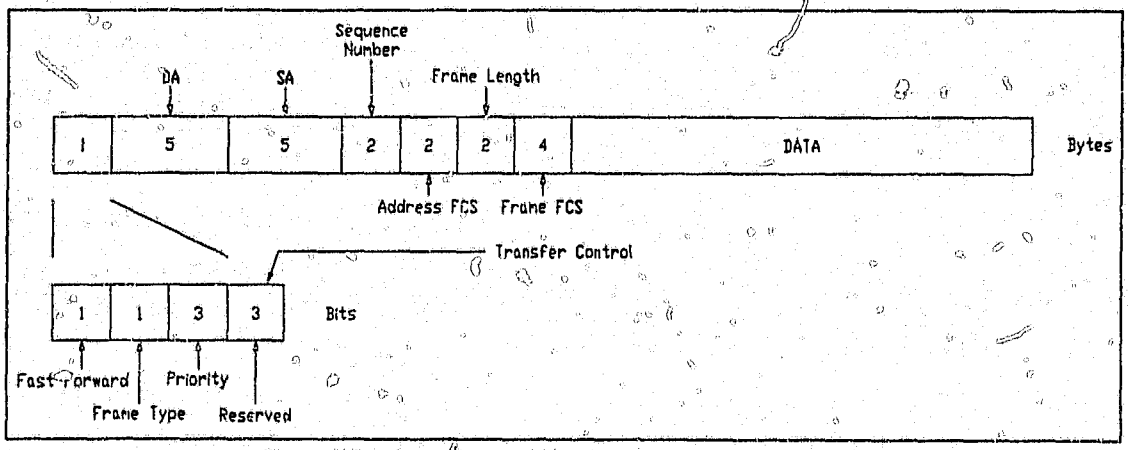


Figure D3 Information and Management frame composition

D.1.4.1 Header Segment

Transfer Control Field: (This contains 2 subfields)

Priority bits: Three bits for eight levels of priority control.

Fast forward bit: Signifies whether this frame can be fast forwarded or not.

Frame Type bit: Indicates the type of frame (information or management).

Destination Address (DA): The address, as in section D.2.3, of the station intended to receive this packet.

Source Address (SA): The address, as in section D.2.3, of the station from which this packet originated.

S - Sequence Number: This contains the sequence number of a sequential information frame for information frames, or the sequence number of management information for management frames.

AFCS - Address Frame Check Sequence: This field contains the checksum for just the address fields of this frame.

L - Frame Length

FCS - Frame Check Sequence: This field contains the checksum for the whole frame.

D.1.4.2 Variable Length Field

This variable length field is used for either management information or user data ie. higher layer protocol data units and application data.

D.2 Metroring Medium Access Control

Each input and output port used for an inter-DTE connection has a dedicated MAC for controlling information flow between the two DTEs. Each line of this input/ output port has a corresponding input and output buffer that can each accommodate at most a 4 kByte frame (maximum length) or alternatively a number of smaller frames.

D.2.1 Access Control

As there is no contention for the medium, frames are transmitted sequentially one after another with minimum spacing. Any spacing resulting is due to processing times in the Metroring station.

D.2.2 Priority Control

Priority control is implemented in the path layer. Four separate levels of priority are used with a special priority mechanism called the Fast Forward, FF, Bit (described later). The path layer and the above application layer both assign frames their respective priorities.

D.2.3 Addressing

A full address in MR is composed of three parts:
Region-ID.Segment-ID.Node-ID.Port-ID

The region and segment ID's are numbers correspondingly indicating which MR network region and segment within that region the DTE is situated on. The node-ID part of the address is an identifier assigned by the management sublayer to every node on the segment. Every node on a segment has a unique Node-ID. The Port-ID is a number indicating what class of application the frame is intended for. Several applications could be running at once on a node and some frames could also be destined only for layers in the Metroring protocol stack. The port-ID does not correspond to the physical input and output ports of the node.

A special unique identification (UI) number is also available for each node. The UI is made up from the MAC and Network layer addresses as defined in the ISO and IEEE protocol stacks. Although the UI is not used in the addressing scheme it is required for resolving node address allocation problems eg. when two nodes assign themselves the same Node-ID, the node with the lower valued UI takes control of reassigning addresses.

The address table used in addressing and routing out-going frames is constructed from full addresses and contains the output port associated with each DTE, as discussed in section D.4.

D.3 Link Control Protocol

Efficient flow of information is enabled via three types of frame buffers present in the Metroring entity:

- one frame queue per node,
- one receive buffer and
- one transmit buffer for every input/ output port on the node.

The receive buffers simply store correctly received frames until they are processed by the Metroring entity. The frame queue temporarily stores frames on their way through the node, or frames for transmission or frames destined for the application.

The Link Control Protocol uses these buffers to ensure that correct transfer of information takes place. The only connections maintained by the path layer are controlled at this sublayer. The connections maintained between any two adjacent stations are link-to-link connections. The end-to-end connections between source and destination stations required by higher layers, are comprised of these link-to-link connections.

D.3.1 Soft Error Recovery

When frames are discarded at the MRMAC layer due to frame check error (ie. mismatched calculated and received frame check sequences), the rest of the Path layer is not informed. Indication of frame corruption is evident when one or more frames are out of sequence. The Link Control Protocol layer will detect an inconsistency in the sequence numbers contained in the frames it receives.

For every frame correctly received by the destination station the Link Control Protocol returns a positive acknowledgement frame (management frame) to the sender. The sending station then discards its copy of the acknowledged frame. If a consistency in received frames' sequence numbers is detected, then the receiving station sends back a negative acknowledgement frame indicating the sequence number of the missing frames. The sending station will then re-schedule the discarded frames for retransmission. The receiving station simply leaves space for these discarded frames without discarding the correctly received frames in the sequence. Acknowledgements can also be piggybacked on information frames if possible.

D.3.2 Flow Control

Each frame to be transmitted from the application on the DTE is first loaded into the frame queue before it is passed to an output MAC buffer. Whenever an output buffer is empty, the frames destined for that MAC's port are sent to its output buffer.

Frames that are received get contained in the input MAC buffer. These frames may then be passed to the application or to a different outgoing port. When frames destined for a station other than the one they are currently passing through, they are referred to as *through traffic*. These frames are only passed to other links if the corresponding destination DTE's receive buffer is empty otherwise they are passed to the frame queue in the current station. The status of the adjacent station receive buffers is made known through management packets received from the respective adjacent stations.

A special technique called *fast forward* is used to reduce transmission delays for through traffic. As soon as a packet is received, it is immediately transmitted on all outgoing links with empty adjacent DTE receive buffers. While these transmissions are occurring, the current DTE compares the frame's destination address to its own and those within its routing table. If the frame was destined for the current DTE, transmission on all the outgoing paths is halted. If the frame was to be routed to another outgoing path, the retransmissions on those links that are not part of the frame's intended path are halted. The truncated frames, received by adjacent stations due this transmission halting procedure, are simply discarded.

D.4 Route Control Protocol

MR architecture is defined with at least two independent output ports. This implies each DTE requires a routing function for sending frames to the correct output ports. The routing function also ensures the packets traverse the correct paths to their destinations.

D.4.1 Hierarchy Routing

A dynamic routing scheme called hierarchy routing is used. It prioritizes the sending of frames firstly to the target region, then the target segment and then the target node. All that each node, that receives a frame not intended for itself, does, is (1) choose the output port the frame must be transmitted on and (2) transmit it on the link connected to the port. Routing tables are used for this decision process in each node. No information is used in nodes for maintaining paths for a connection, ie. the series of links to traverse to a destination. Each node then makes a routing decision every time a frame passes through en route its destination node.

Routing is done in a hierarchical manner: (1) If the frame does not belong in this region the nodes route it to the bridge connected to the frame's destination region. (2) If the frame is not on the destination segment, 2 port nodes simply relay the frame while SENs route the frame to the correct segment. (3) If the frame is on the correct segment, nodes route it to its destination station on the segment.

One important characteristic of the routing method used, is its ability to take physical link changes in the network and also the congestion of links into account. Routing tables are modified when hard errors in the network occur and when link congestion changes occur.

Congestion is monitored by every node for each of its connected links by taking note of an average frame delay. This average frame delay is the time it takes for a frame to propagate to the adjacent node enroute to its destination node plus the time for the adjacent node's acknowledgement to reach the source node. The delay is calculated for frames using the current node's links for every 10 second interval. For one link's frame delay, if the newly calculated value is different from the original frame delay by a certain threshold, the original frame delay is updated. However, the threshold is not fixed but a varying value. The threshold starts off at 64 msec and decreases by 12,8 msec for every 10 sec measurement period where the frame delay is not updated. This implies that the frame delay will eventually be updated.

SENs and bridges require the average frame propagation delays for every possible destination segment or bridge in the network to construct their routing tables (described in the next section). A special type of frame, Link Delay Notification Control Frame (LDNCF), is used every 10 seconds by bridges and SENs to collect necessary delay information to construct or modify their routing tables. This requires the average frame delay of the link on which the LDNCF is received on, gets added to a special link delay time field in the LDNCF. The modified LDNCF is then transmitted on all other outgoing links. The original LDNCF is transmitted back along its source link while any duplicate LDNCFs received by a node are discarded. When a SEN or bridge receives one of these LDNCFs it updates its routing table.

D.4.2 Routing Tables

Routing tables are maintained by each node to allow choice of which output port a frame must be sent on to reach its destination. However each class of node constructs and uses its routing table differently:

D.4.2.1 Two Port Nodes

A two port node:

- chooses the path with the least number of hops to a 2 port node on the same segment. The 2 port node section of its routing table keeps track of (1) all nodes on the same segment including the source node itself along with (2) the number of hops to each node.

- keeps track of the SEN nearest it along with the SENs address and the output port leading to the SEN.

D.4.2.2 Segment End Nodes

A Segment End Node:

- selects the link with the least number of hops to a 2 port node on one the segments which it terminates. The 2 port node section of its routing table keeps track of all nodes on the segments terminated by this SEN including the number of hops to each of the 2 port nodes.
- keeps track of all the segments in the current region. For each segment it holds the segment number, the output port leading to it and the propagation delay to the segment. Routes are chosen according the shortest propagation delay to the segment.
- selects the route with the shortest propagation delay to the adjoining bridge for a different region. The region number, output port and propagation delay to adjoining bridges are noted for all regions comprising the Metroring network.

D.4.2.3 Bridges

Each bridge:

- keeps track of all the segments in the regions it is attached to. For each segment it holds the (1) segment number, the (2) output port leading to it and the (3) propagation delay to the segment. Routes are chosen according the shortest propagation delay to the segment.
- selects the route with the shortest propagation delay to the adjoining bridge for regions to which it is not attached. The region number, output port and propagation delay to adjoining bridges is kept for all regions comprising the Metroring network.

D.5 Management Protocol

The management protocol informs all nodes when its adjacent nodes power up or down, or when a line becomes faulty. It is also used to inform the node's user or application of the node's and network's status and settings.

D.5.1 Hard Error Recovery

When a link or station fails the two stations adjacent to the failed component do not reconfigure internally. Instead the adjacent stations stop using the port on the side where the problem has occurred until it has been repaired. Special management frames are transmitted by these two stations to the rest of the two port nodes on the segment to inhibit new nodes from attaching to the segment.

After a two port node receives this fault identification management frame, it relays it onto the next node on the segment and simultaneously updates its routing table. The port on which the management frame was received is recorded by each two port node along with the node number of the functional node closest to the fault. This allows all segment nodes to be aware of the fault's boundary for routing table correction.

At the SENs, these management frames are not relayed further. The SENs will modify their routing tables to cause frames destined for the damaged segment to take different paths to their destination nodes if necessary. The SENs will then transmit complementary management frames back along the damaged segment to allow new stations to join again.

After the fault is corrected, the nodes adjacent the fault initiate the procedures for making the segment fault free again. This will entail reassigning node numbers where duplicates were created during the duration of the fault condition. Again these two fault neighbouring stations issue frames that inhibit new stations linking into the segment. These two stations then swap node address entries in their routing tables. The neighbouring station with the lowest UI (unique identifier), reassigns addresses to nodes with duplicate entries. It then recompiles the node section of its routing table and transmits the routing table to all nodes on the segment. This final frame, containing the two port routing table, also signifies that new nodes may join into the network.

APPENDIX E: THE EXPRESS TRANSFER PROTOCOL (XTP)

XTP is a Transfer layer protocol. The Transfer layer results by combining the network and transport layers of the OSI model. It is designed for reliable transmission of data in an internetworked environment.

E.1 Basic Packet Exchanges

E.1.1 Single Receiver Mode

Figure E1 shows how XTP can reliably set up a connection between two users, transmit the data and close the connection. The three packets used in this exchange perform several functions:

(A) In the first packet the source informs the destination that it wishes to establish a connection.

(B) If the destination complies, it establishes a context and returns a control packet to acknowledge receipt of the data and indicate the destination is ready to close the connection.

(C) The source then transmits another control packet before closing its side of the connection. When the receiver receives the packet it closes its side of the connection.

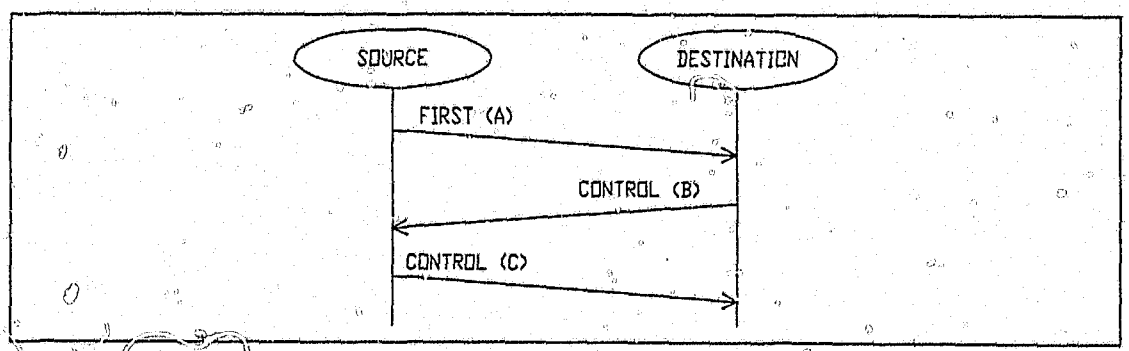


Figure E1 XTP three packet connection mode handshake

Three flags RCLOSE, WCLOSE and END are used in coordinating a connection's close. RCLOSE and WCLOSE are set by a local host when transmitting to the remote host to respectively indicate it is finished reading and writing data. Thus each direction in a full duplex connection can be shut down independently. The END flag is sent in the last packet transmitted by a station to indicate that it has released its end of the connection.

E.1.2 Fast handshakes

Fast handshakes are two packet exchange sequences allowed in XTP. These special exchange modes are less reliable than the three packet connections used and are normally considered a transport level datagram service used for simple request/ response operations.

E.1.3 Multicast Mode

In *multicast mode* one sender targets many destinations. Operation proceeds very similarly to the *single receiver mode.*

Error control must use the *go-back-n* retransmission scheme. *Reject* packets are sent by receivers detecting out of sequence data, but a special precaution is taken to ensure that all receivers do not simultaneously announce an error. Whenever a station detects a *reject* packet passing belonging to the same context, it compares the RSEQ value in the packet to its own value. If its own value is smaller it then transmits its own *reject* packet.

E.2 Frame Format and Structures

XTP uses two types of packets: Information packets and Control packets. Both packet formats contain common header and trailer segments of constant length as shown in figure E2. The type of the variable length segment between the header and trailer segments determines the type of the packet.

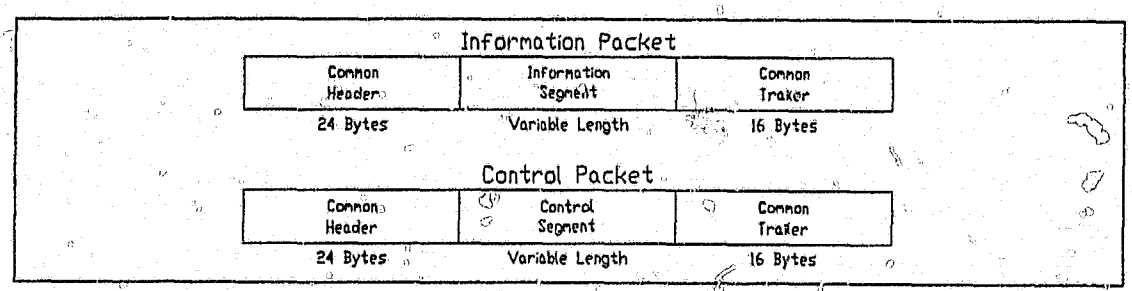


Figure E2 XTP packet types and formats

E.2.1 Common Header

Contains the following fields:

- Packet type
- Description of portion of data stream in information segment (if used)
- Control Flags
- KEY: a special field used for looking up addressing information in the either station of a connection
- LITTLE bits: for determining the byte ordering of the packet
- ROUTE: This is a value used by routers in determining which path a packet must take in internetwork transmissions.
- SEQ: The starting byte sequence number for an information segment of a packet that is part of a stream of data.
- DADDR: Flag for indicating use of direct addressing mode

TTL: time to live field
SORT: for priority control

The control flags are used for specifying:

- Disabling error checking
- Multicast transmission mode
- Beginning Tagged Data (BTAG) flag

E.2.2 Information Segment

This contains user data being transferred. It also passes addresses and other miscellaneous data when appropriate.

Each data packet contains a contiguous portion of the data being transferred. Each of these data packets is limited in length by the underlying data link or MAC layer. The maximum length of the information field is defined as the maximum transmission unit (MTU) for XTP. It is calculated by subtracting the length of the common header and trailer segments from the maximum length allowed by the underlying data link layer.

In packets that are fragmented, the information field is simply broken up into the required smaller sizes with the fragmented packets (also called segments) containing their own header and trailer fields. The header of the first segment and the trailer of the last segment correspond to the header and trailer of the original packet (except of course the checksum values). The headers and trailers in between contain a number of differences to the original header and trailer (like sequence numbers).

E.2.3 Control Segment

Fields for resynchronizing the transmitter and receiver stations are among those used here:

- ALLOC: One greater than sequence number of last byte receiver will accept
- RSEQ: One greater than sequence number of last byte receiver accepted
- RATE: The maximum number of bytes the transmitter of this packet will accept in one second
- BURST: The maximum number of bytes a receiver (transmitter of this packet) will accept per burst
- SYNC: A field used in keeping track of synchronizing frames
- ECHO: A field used in keeping track of echo frames responding to synchronizing frames
- TIME: A field set to the present time when a synchronizing frame is transmitted. It is copied to the echo frame at the acknowledging receiver.

E.2.4 Common Trailer

Contains following fields:

- DCHECK: Checksum for data fields, including information and control segments in respective packet types.
- HTCHECK: Checksum covering header and trailer fields.
- DSEQ: One greater than sequence number of last byte receiver delivered to destination client process
- Flags

The flags field contains:

- End of Message flag (EOM)
- Ending Tagged Data (BTAG) flag
- Immediate Status Request bit (SREQ)
- End of Transmission Request bit (DREQ)
- NODCHECK checksum is present (NODCHECK)
- Checksum calculation disabled for current packet (NOCHECK)

E.3 Addressing

E.3.1 Intra-Network Addressing

Each XTP entity in peer nodes must be able to uniquely identify its peer in a connection. To minimize all addressing data in each packet, all addressing data is passed in the first packet

sent and cached at the source and destination stations. The *KEY* field is then created and used to index into the cache to access all the addressing data needed for a connection. Subsequent packets then only contain the *KEY* field for addressing purposes in the XTP layer.

The initiator of the connection sends all the addressing data plus the *KEY* in the first packet in order to identify the intended receiver. The receiver saves the context identifier (*KEY*), the MAC source address and the route identifier (ROUTE) in a database associated with the context record.

In a two station connection a special key exchange occurs so that context identification occurs faster. The key values used in most data packets are intended for aiding the destination's context look up, but initially the corresponding station's key value is not always known. To clarify what side of the connection a key originates from, the most significant bit in the key's 32 bit wide field signifies whether the key originates from the station receiving the frame (bit value 1) or the station sending the frame. If this bit is set, the key is referred to as a return key and its value may be used in looking up the context. The receiver of the connection returns its own *KEY* value in the *XKEY* field of the first CONTROL packet returned to the initiator. After the keys have been exchanged, subsequent packets use return keys in referencing the context.

The first packet in a connection contains two main fields for addressing: a fixed length descriptor field indicating the address format used and a variable length field containing the list of destination addresses. At initialization the application process specifies to the XTP entity which addresses to connect to. Thus to establish whether a station must take part in a connection the addresses in the packet's address field must be checked against those specified by the application process; if a match is found, the station then partakes in the connection.

A special mode of addressing called *direct addressing mode* is used in the case when the network topology is known. It is indicated by the DADDR flag in the common header. In this mode of addressing, the *KEY* field contains the actual destination address instead of an index used to look up a context.

E.3.2 Inter-Network Addressing

Routers relaying packets between networks must keep track of the routes or paths used for packets. *ROUTE* fields in the packets being routed are used for this. The *ROUTE* value is used by a router to access the necessary addressing information in its cached address translation map for a packet travelling on a specific route. Routers or endpoint nodes may change this *ROUTE* value to change the next host or router taken by the packet on the route. Acknowledgement packets use a *return form* of the *ROUTE* value.

The destination end of a connection may choose to exchange *ROUTE* values with the router. A *XROUTE* field is used for this *ROUTE* identifier exchange between the end station and the router; the router will then use the return form of this route identifier when routing packets to the receiving station in future. A router then uses this *XROUTE* field in acknowledging CONTROL packets whenever it wishes to exchange a *ROUTE* number with a previous router on the current path. As routers can dynamically redirect packets across networks, separate paths can develop for both directions in a two way connection.

E.4 Prioritization

The *SORT* field is for indicating a packet's priority and is 32 bits wide, thus supporting a very large range of priority values. Each context is associated with a particular priority level which is static ie, priority does not change at any point in the network. Several contexts may work at the same priority level simultaneously.

The level of priority is inversely related to the value in the *SORT* field in XTP. Packets are enqueued in receive buffers in the order of priority level.

E.5 Flow Control

XTP keeps track of the amount of information flow between two stations by means of sequential byte numbers instead of the commonly used packet numbers. Three fields found in the control and common trailer segments are used in controlling flow of bytes between stations by keeping track of the receiving station's receive buffer status:

ALLOC is used in control frames by the destination station to prevent the source from sending more bytes than the destination's buffer's can accept.

DSEQ is the sequence number of the next byte in the local queue to be passed to the application process in the local station.

RSEQ is the sequence number of the next byte to be received by the local station.

The size of the buffer used by a context is thus equal to *ALLOC - DSEQ*. Also the number of bytes waiting in the buffer before being sent to the local application process is given by *RSEQ - DSEQ*.

XTP uses 32-bit sequence numbers and *RSEQ*, *DSEQ* and *ALLOC* to ensure that delayed retransmissions are not confused with new frame data having the same sequence numbers. A source will continue to transmit data to the destination until the destination's allocation has been reached - the allocation being specified by ALLOC in an initial acknowledging frame. When the destination acknowledges receipt of the bytes received, it means more space is available for the source to fill.

An alternate allocation mode called *reservation mode* may be used in buffering incoming messages. Here *ALLOC* is set equal to the receiving station's buffer size for the required context to inform the sending station what maximum storage space a message may occupy. The sending station must pause between these message transmissions for the receiver to establish new receive buffers for each message. The buffers reserved in this mode vary greatly in size to those during normal allocation.

E.6 Rate Control

Rate control becomes necessary when different implementations of XTP must deal with the difference in station processing speeds. It is essentially a method of controlling the size of and spacing between bursts of data sent by the sender so that the receiver is not overloaded with data (and will not drop packets due to overloading).

XTP uses the *RATE* and *BURST* fields in the control segment to allow a receiver to tune its data transmission rate to a level acceptable to the local host sending the control packet. The minimum period between incoming bursts which the receiver can effectively deal with is equal to *BURST/RATE*. This pause is equal to the value in a rate timer, *RTIMER*.

Author Dahl B
Name of thesis The development of a method for comparing the operation and performance of Metropolitan area networks 1993

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.