Durham E-Theses

Performance of buffer insertion LANs

Jackson, Christopher

Performance of Buffer Insertion LANs

by

Christopher Jackson, B.Sc.

A Thesis submitted in partial fulfilment of the requirements for the degree of Master of Science

School of Engineering and Computer Science
The University of Durham
1994

Memorandum

The research presented in this Thesis has been carried out at the School of Engineering and Computer Science of the University of Durham between October 1988 and October 1991. It is the original work of the author unless stated otherwise.

Abstract

This thesis presents an evaluation of the performance of some ring local area network protocols, specifically the buffer insertion ring and the token ring. Both dual and single versions of each are considered, and performance statistics are obtained by measurements on real networks, by mathematical analysis, and by computer simulation. New packet routing protocols for dual contrarotating buffer insertion and token ring networks are described and their performance is examined. It is found that, due to the bandwith reuse properties of the routing methods, maximum throughputs of 800% and 200% are achievable, for the dual buffer insertion ring and the dual token ring respectively.

Acknowledgements

Publications

C. Jackson, J.E. Mellor and G.M. Geary *'Hierarchical CIM networking'* Computer-Integrated Manufacturing Systems, Vol 2 No 4 November 1989.

C. Jackson and J.E. Mellor *'Performance evaluation of contrarotating LANs'* Proceedings, IEE 7th UK Teletraffic Symposium, April 1990.

J.E. Mellor and C. Jackson *'Performance evaluation of time critical fault tolerant local area networks'* Proceedings, BCS 6th UK Computer and Telecommunications Performance Engineering Workshop, July 1990.

C. Jackson and J.E. Mellor *'Bandwidth usage optimisation in dual ring LANs'* Proceedings, Bangor Communications Symposium, May 1991.

Contents

Chapter 1

Introduction

In recent years local area network (LAN) technology has advanced significantly and the computing/control industry has seen a new breed of network come into existence. Based on high speed data transmission and fault-tolerant techniques, these new types of LAN are likely to become the next generation standard for local area communications. While useful in the traditional LAN domain of office and campus resource sharing, they are particularly suited to more demanding areas such as factory automation, real-time vehicle and weapons control, and metropolitan area networking. These applications require not only high information throughput and low latency, but also automatic and speedy response to faults in cabling or interface units.

Some examples of such LANs are the ANSI standard Fiber Distributed Data Interface (FDDI), the Society of Automotive Engineers' High Speed

Ring Bus (HSRB), and Nine Tiles Computer Systems' CoRoNet. The first two of these are token ring protocols and are reasonably well-known, whereas the last one employs a buffer insertion protocol and is currently under development. When deciding on the suitability of a particular network for a certain application, some information about its performance must be available, such as reliability, maximum throughput, adaptability to faults, etc. This thesis presents an evaluation the performance of some of these networks, in particular the buffer insertion protocol. As buffer insertion rings have been less widely investigated, it was deemed to be important that various protocol parameters should be investigated fully, in order to evaluate the performance likely to be obtained from products using this protocol. In particular, the buffer insertion protocol exhibits a unique 'spacial reuse' property which can allow more than one station to use the full link capacity simultaneously. Additionally, by coupling this property with novel packet routing strategies, the dual buffer insertion rings can offer average throughputs of well over 100%. The thesis investigates these possibilities and, to illustrate the performance gains which can be achieved, offers a comparison between dual buffer insertion and token rings.

The thesis begins, in Chapter 2, by reviewing the current state of local area networking. LAN protocols and design principles are considered, and the token ring and buffer insertion ring protocols are described in some detail.

Chapter 3 presents a consideration of the meaning of 'performance' as it relates to LANs. Three different techniques of performance evaluation are discussed, namely measurement, mathematical analysis, and simulation, and the relative merits and disadvantages of each are considered.

The performance of the buffer insertion ring protocol is examined in Chapters 4 and 5, by the three means described in Chapter 3. Both single and dual ring versions are included, and some new techniques concerning packet routing in the dual ring LAN are put forward and tested.

Similarly, Chapter 6 deals with the token ring protocol. Again, both single and dual ring versions are investigated, and a method of improving dual ring performance is tested.

Finally, Chapter 7 presents a summary of the main body of the work, in the form of a comparison between some of the protocols studied.

Chapter 2

Review

2.1 Local area networks

A local area network (LAN) is comprised of an interconnected set of communicating elements, which may include computers, printers, programmable logic controllers, disk drives, and so on. The elements (also referred to as stations or nodes) communicate digitally and the entire network spans a limited geographical area, typically less than a few km^2. Transmission speeds are usually high, often tens of megabits per second (Mb/s) [10, 13, 29]. LANs have a substantial range of application. At one end of the scale they may be used in an office building to allow electronic communication between workers and the sharing of resources such as printers, large file stores and databases.

At the other end of the scale they may form the communications background of a large manufacturing plant, in which the entire manufacturing process depends upon the network for intercommunication between machine tools, control programs and operating consoles [18]. LANs share many common properties with wide area networks (WANs), but are different in several important respects. Franta [6] identified these in the early 1980s as follows. LANs

1. Employ high-bandwidth channels (generally in the megabit range), whereas WANs employ low-bandwidth channels (generally in the kilobit range);

2. Exhibit transmission delays that are shorter than those in WANs (primarily because of the use of high-bandwidth channels);

3. Employ channels with very low error rates, not only compared with those of WANs but in an absolute sense as well;

4. Can easily employ state-of-the art technology, as their construction is not subject to the constraints, especially legal, often associated with WANs that use common carriers; and

5. Are more closely and easily controlled (managed), as they are generally owned and operated by a single association.

Recently, with the development of very high speed fibre optic LANs, the distinction between LANs and WANs has begun to blur, as LANs evolve into metropolitan area networks (MANs) and span increasingly large areas, typically entire cities [26, 15].

Figure 2.1: Conceptual model of a local area network

Figure 2.1 shows a conceptual model of a LAN. The heart of the system is the communications subnetwork, which is responsible for the transport of data between the nodes. The devices that wish to communicate are known as the hosts. These include computers, printers and so on. They are connected to the subnetwork via the network interface units (NIUs). Each NIU is

responsible for accepting data from its host and negotiating its transmission across the network, and also for receiving data destined for its host and buffering it until the host can accept it.

This model leaves the communications subnetwork unspecified. Various techniques exist for the actual transport of data between hosts. They can be classified in different ways, but the most common means is to identify the network's *topology* and its *protocol*. Topology refers to the physical layout of the network. Common topologies are star, ring and bus. The protocol is the set of rules that the network uses to govern access to and use of the transmission medium. This is partly determined by the topology in use. Typical protocols include token passing (for bus or ring topologies), buffer insertion (for rings), and carrier sense multiple access with collision detection (CSMA/CD) (for buses). The buffer insertion ring and the token ring are the main focus of this thesis; these will be discussed later.

2.2   The ISO OSI model

Until recently, different LANs from different manufacturers used different topologies and protocols. These were typically incompatible with each other;

one had to buy all ones networking equipment from one vendor. In an effort to standardise the means by which systems communicate with each other, a number of groups within the International Standards Organisation developed the *Reference Model of Open Systems Interconnection* [16], referred to here as simply the OSI model. This defines a standard set of methods for intercommunication, as well as a model for a network architecture based on a layered approach. Briefly, a 'system' (which may be anything from a dumb terminal to an entire WAN) is represented by a hierarchy of layers. Figure 2.2 shows how two such systems may communicate with each other.

A   B

```
      ( application )                    ( application )

  7  |  application  |              7  |  application  |
  6  | presentation  |              6  | presentation  |
  5  |    session    |              5  |    session    |
  4  |   transport   |              4  |   transport   |
  3  |    network    |              3  |    network    |
  2  |   data link   |              2  |   data link   |
  1  |   physical    |              1  |   physical    |

                    PHYSICAL MEDIA
```

Figure 2.2: The OSI model

11

The protocol 'stack' acts as an interface between the application on the host and the physical transmission media. Each layer of the stack performs certain well defined functions. Some layers may run on the host, while some may form part of the network interface unit. Should an application on system A wish to send a message to an application on system B, it must do so via the different layers of the stack. It builds its message, including data, control and address information, and passes it to the application layer interface. The message then propagates down the stack, each time having extra control information added to it and being processed in some way, until it reaches the physical transmission media. It then travels up the OSI stack in system B, layer-by-layer, each time having the control information stripped away, until it reaches the application layer. The application in system B can then make use of the data. The functions of the various layers are as follows:

- o 7. Application layer provides network services to applications programs, typically file transfer, electronic mail, etc.

- o 6. Presentation layer manages the way in which data is presented to the application layer, performing character representation translation, encryption, etc.

- o 5. Session layer sets up, maintains, and closes down sessions with another system.

12

o 4. Transport layer provides a reliable end-to-end data path for the session layer, relieving it of issues which are network dependant.

o 3. Network layer manages the routing of packets through the network. In LANs this layer usually has a reduced functionality compared with its use in WANs.

o 2. Data link layer manages access to the transmission media and performs error control and correction.

o 1. Physical layer is concerned with the specification of the electrical aspects of the communications channel, such as voltage levels and cable types.

LANs conforming to the OSI model make more use of some layers than others. Certain functions of the presentation layer, for instance, may be redundant, and routing techniques within LANs are generally simple enough to require only a minimal network layer.

The two main types of LAN studied in this thesis are the buffer insertion ring and the token ring. These networks are examined in more detail in the following sections.

2.3 The buffer insertion ring protocol

The buffer insertion (or register insertion) LAN protocol originated in 1974. Haffner et al [9] proposed the protocol and it was later developed at Ohio State University [25, 19]. It was subsequently incorporated into the IBM Series 1 [14] and SILK [12] networks. The implementations of buffer insertion studied in this thesis are those produced by Nine Tiles Computer Systems Ltd.

Figure 2.3 shows a schematic diagram of a station on a typical buffer insertion ring.

The operation of the protocol is as follows. When the station has no data to transmit, it simply waits for incoming packets from the ring. If an incoming packet is not destined for the node, it merely passes straight out of the node via the central path. Conversely, if the incoming packet is addressed to the node the packet will be diverted into the receive buffer and thereby removed from the ring. In this case, the address bits may well have passed out of the node and into the next node downstream, but the network incorporates a mechanism to detect this and remove the stray bits from the ring. When all of the packet is in the receive buffer it can be transferred to the host for processing.

Figure 2.3: Buffer insertion ring station

In the case when the host has data to transmit on the network, it must wait until the transmit buffer is empty then transfer its packet into this buffer. The node then waits for either a gap in the incoming traffic or a packet to be copied from the ring into the receive buffer. Also the insertion buffer must be empty. When these conditions are satisfied, the transmit buffer is emptied bit by bit onto the ring. Should any data enter the node from the ring during this process it is stored in the insertion buffer. When all the data has been copied out of the transmit buffer the insertion buffer may then empty itself onto the ring. While this occurs, any data coming in from the ring enters the back of the insertion buffer and must pass through it and back out onto the ring; the insertion buffer will only shrink in size when there is a gap in the incoming data.

Because data packets are removed from the ring at the destination node rather than the source node, there is no facility for implicit acknowledgement signals within the packet (unlike certain token ring protocols). The destination node, having received a packet, must send a separate acknowledgement packet to the source node to indicate that the data has been received correctly.

2.3.1 Proprietary buffer insertion ring implementations

As the work presented in this thesis is based largely on implementations of the buffer insertion protocol by Nine Tiles, some vendor-specific features of their products are detailed here. The two networks considered are Superlink, a single ring LAN, and CoRoNet, a dual contrarotating ring version still under development [7].

Superlink can operate across various different types of media, including twisted pair and coaxial cable. It offers a choice of two data rates, 1.5 Mb/s and 250kb/s, and is available in several different units. The only ones relevant to this thesis are the card version (which plugs into the bus of a standard IBM PC computer) and the RS232 version (which is a stand alone boxed unit featuring and RS232 interface for connection to dumb terminals etc). The PC card version is supplied with appropriate driver software and an industry standard NETBIOS interface. A later version for the PC, known as Swiftlink, was also used, but operationally this is very similar to Superlink. A network file system known as SimpleNet may be used with Superlink-ed PCs. The operation of Superlink is largely the same as the generic protocol described above. Messages are split into packets of variable length (with a maximum of 86 bytes) which are sent individually across the network. Each packet

contains a 48 bit header consisting of control and address information. Each received packet gives rise to an explicit acknowledgement packet to indicate successful reception. The network consists of only one ring and stations are connected in a simple point-to-point fashion.

CoRoNet is the dual contrarotating ring successor to Superlink. It is still under development at Nine Tiles R&D laboratories, but this much can be said about it: It consists of two buffer insertion rings running in parallel. Each station is connected onto both rings, and data flows in one direction on one ring and in the opposite direction on the other ring. Both rings are used simultaneously to carry data and control traffic. A facility exists whereby should a break occur at a point in the ring, the network can reconfigure to form a single ring around the break and continue operating (though perhaps with some performance penalty), thus providing a degree of fault-tolerance. Again, the transmission media and channel data rate are variable, but the maximum data rate will probably be around 25 Mb/s across coaxial cable. The method of routing of packets through the network has not yet been decided, and this thesis investigates various techniques and assesses their suitability in terms of performance. However, the initial idea for a simple routing strategy is as follows. When a station has a message to send, and assuming the message is longer than the maximum size of a packet, then the first packet of the message is sent out on both rings, in opposite directions.

The sending station then awaits the two acknowledgement packets and, based on which arrives first, chooses one of the rings as the 'quickest path' through the network. All subsequent packets of this message are sent out on this ring only, leaving the other ring free for other stations to use. When all packets of this message have been delivered, the station reverts to its initial state, prepared to perform the same routing scheme for the next message it receives from its host. This is a reasonably effective technique, though it does have its problems, as will be discussed in Section 4.3.

2.4   The token ring protocol

The token ring protocol is one of the oldest ring LAN control protocols. It originated in 1969 and was known as the Newhall Ring [5], and was subsequently standardised as one of the IEEE 802 LAN standards [23]. Various manufacturers, including IBM, have adopted it for use in their products.

The basic principle of the token ring protocol involves a short frame, known as the token, which circulates around the ring. A token can be in one of two possible states: free, indicating that a node may acquire it for data transmission, or busy, indicating that it is already in use by another node

and cannot be used for transmission. When a station wishes to transmit it must wait for an incoming free token. It then alters the bit pattern of the token frame to indicate that the token is now busy, then sends a data frame immediately after it. The frame travels all the way around the ring and is removed by the transmitting station. The receiving station makes a copy of the frame as it passes through, and may alter a control bit at the end of the frame to indicate whether or not it was received correctly. After purging the frame from the ring the transmitting station then issues a free token which another station may subsequently use.

The IEEE 802.5 specification builds on these basic principles by adding more specialised features. It is a single-token protocol, in which a station which has completed a transmission may not issue a free token until the busy token returns, and it employs a priority system which regulates access to free tokens based on importance or urgency of certain messages or stations.

At the time of conducting the research certain standards pertaining to high-speed dual token rings were being formulated. As these standards were not yet stable, work presented in this thesis relates to a generic dual token ring protocol which is similar to 802.5 in its operation. This is described in detail in Chapter 6.

Chapter 3

Performance issues and methodology

3.1 Introduction

The term "performance" in relation to local area networks encompasses various criteria concerned with the quality of service a network presents to its users. Typically, it can include such metrics as throughput, message transit time, error rate, lost packet count, and others. Performance issues are of great importance to network users; for a given application, a system designer must choose a network whose performance will be adequate for that application. The designer will specify the quality of service required and will then choose a network which can meet these requirements.

21

The ISO standards do not adequately address these issues. Work is under-way within the Time Critical Communications Architecture (TCCA) activity to address the problem [8].

Three main techniques are available for the performance analysis of local area networks, these being measurement, mathematical analysis, and simulation. No one technique is ideal for all situations, and each one has its own advantages and disadvantages. Their relative merits are discussed below.

3.2   Measurement

Initially measurement may seem the most obvious way of evaluating the performance of a LAN. The technique is reasonably simple: set up the desired configuration of the network to be investigated, introduce some test traffic, and measure the progress of the test traffic through the network. For example, to assess the performance of a campus Ethernet LAN one might set up two workstations as test nodes. One station could send time-stamped messages to the other one; the receiving station would thus be able to calculate the transit time of a message through the network. By making such measurements at various times of the day an accurate assessment of the network's

performance could be obtained.

This technique can be highly accurate and effective when the network to be investigated actually exists and is available for measurements to be made on it, i.e. when a performance index of the network 'as it stands' is required. The shortcomings of this approach become apparent when predictive statistics are required. If one would like to know the effect of increasing the number of nodes in the network, or changing the cable length, or using fibre-optic instead of copper connectors, measurement is not a viable technique. Changing a network's configuration purely for predictive reasons may be highly expensive and may give rise to unacceptable quality-of-service to network users (e.g. poor throughputs and unexpected network failures.) For performance assessment of network configurations which do not actually exist, or, more significantly, of networks which have been designed but not yet implemented, mathematical analysis or simulation are preferred techniques.

3.3   Mathematical analysis

Performance assessment by mathematical analysis involves constructing a mathematical model of a network and then solving it for a given scenario. At

the most basic level a local area network can be seen as a system consisting of queues (of messages to be delivered) and servers (the links between the nodes) [27]. Well established mathematical techniques, such as queuing theory, can be used to analyse these systems and derive expressions which describe their behaviour. The advantage of this is that a network need not actually exist in order for its performance to be evaluated, and provided a formal description of a network's operation is available, a mathematical model of it may be built. Variables, such as data rate, cable length, number of nodes, etc, simply appear as parameters in the expressions; to observe the effect of increasing the data rate one is only required to substitute a different value for one of these parameters.

Because an entire network can be modelled by a set of mathematical expressions, analysis is usually the quickest method of obtaining network performance statistics. However, it is seriously limited by the assumptions required for the validity of the techniques employed. Typically, analysis of a LAN requires that the traffic pattern is symmetric, i.e. that all nodes generate traffic at the same rate and send their messages to a node chosen at random. Clearly this is rarely the case in a practical situation. Additionally, analytic methods are often suited only to the lowest levels of a LAN's protocol stack, usually layers 1, 2 and possibly 3. Higher layer protocols are often implemented as software in the host and can be extremely complex in opera-

tion and subject to the performance of the host itself; constructing a suitable mathematical model can be difficult or even impossible. In all but the most simple cases, then, mathematical analysis provides only an approximate indication of a networks performance; for more complex situations simulation is to be preferred.

3.4   Simulation

Simulation of a LAN involves modelling the operation of the network and its hosts by means (usually) of a computer program. The program, when executed, mimics the network's functions, e.g. generation of messages, sending of packets, with respect to time. This 'working model' may then be used for performance assessment by introducing test traffic as one might in a real network.

This approach is highly flexible, because the level of detail which can be incorporated into the model can be chosen by the programmer. It is therefore possible (theoretically) to build a perfectly accurate model of a network and obtain performance statistics which should be identical to those obtained from measurements on the real network. The main drawback of simulation

is that it can often be slow; even if the model is executed on a fast computer, the time to obtain results can be large, especially when a high level of detail is incorporated into the model. However, if the model is well-constructed and not too detailed, simulation can often be more useful than analytic methods.

3.5   A hybrid approach

The three performance assessment techniques described above each have advantages and disadvantages; some may be more appropriate than others, depending on the network under investigation. Different approaches have been chosen for the research work as appropriate and to allow comparison and verification of results. In Section 4.1, Chapter 5 and Chapter 6, analysis and simulation are used because the networks involved were not available for measurement. In Section 4.2, however, networks were available for measurement. In this case, a hybrid system was built, using both measurement and simulation, in order to obtain an accurate yet efficient network model.

3.6 Published work on buffer insertion

Published material on the performance of buffer insertion tends to employ mathematical analysis exclusively. Early work on single rings was carried out by Liu et al [20] and Thomasian and Kanakia [30]. Bux and Schlatter [3] attempt a more accurate analysis and verify this with simulation, and Hammond and O'Reilly's analysis builds on this. Literature on dual ring buffer insertion is scarce but simulation studies are presented in [28] and [17].

Chapter 4

Analysis and measurement of the buffer insertion ring

4.1 Analysis

4.1.1 Introduction

This section presents a performance analysis of the buffer insertion protocol using analytical techniques. The performance measure considered here is the mean message delivery time. This gives a measure of the performance one might expect under 'typical' operating conditions. The latter are specific to each application but are approximated here using certain assumptions

28

which facilitate simpler mathematical models. In particular, it is assumed that the network is operating under fault-free conditions; should a cable break or a node go down, reconfigurations may occur which are difficult to model with any degree of accuracy. Also, for simplicity, we assume an unacknowledged transport service in which packets are removed from the ring at the destination.

The mean message delivery time analysis comes from Hammond and O'Reilly [11], which itself is partly derived from work by Thomasian and Kanakia [30]. The derivation is not repeated here; their results are quoted and extended to cover the dual ring case. The analysis is based on queuing theory [2], and assumes that message arrival is a Poisson process and that message lengths are exponentially distributed. These assumptions are used as a basis for performance analysis, providing an approximation to the characteristics of real local area network traffic. The accuracy of these approximations is somewhat open to question, but some data has been provided to support them [11].

4.1.2 Single Ring

To calculate the normalised average message delivery time for a single ring, T, we begin with an expression from [11], subject to the following assumptions:

- The arrival processes are Poisson, with the same mean arrival rate for all nodes.

- All nodes have the same distribution of packet lengths (exponential distribution.)

- The latency for each station is τ'/M.

- The ring is *balanced*, i.e. the probability of transmitting between two nodes depends only on the separation of the two nodes.

- The capacity of the both the insertion and transmission buffers is infinite.

- The interarrival times and service times are independent.

$$T = 1 + \frac{(\alpha + 1)a'}{M} + \frac{(1 + \alpha)^2 S/M}{1 - S(1 + \alpha)/M} \qquad (4.1)$$

where:

α = average number of insertion buffers passed through by a packet travelling

from source to destination

M = number of nodes

S = normalised throughput

R = data rate

X = average packet length

τ' = total ring latency

a' = normalised ring latency delay = $\frac{\tau'}{X/R}$

Assuming a symmetric traffic pattern, such that a station transmits to each of the others with equal likelihood, we have

$$\alpha = \frac{M}{2} - 1 \qquad (4.2)$$

4.1.3   Dual ring

For the dual ring, assuming the simple packet routing strategy described in Section 2.3.1, the 'quickest path' to the destination node is chosen when sending a message. The message then travels on one of the two rings. For the chosen ring, it is unclear as to exactly what effect this has on the parameter α. α should be reduced, since the quickest path has been chosen, but the

magnitude of the reduction is not easily determined. For the purpose of an example, we shall assume that, under the symmetric traffic pattern, the average number of nodes traversed by a message is effectively halved. So α then becomes

$$\alpha = \frac{1}{2}\text{round}\left(\frac{M}{2}\right) - 1 \qquad (4.3)$$

(the function round(x) rounds the value of x to the nearest integer)

This value may then be substituted into Equation 4.1. This gives the delivery time / throughput relationship for one of the two rings. However, since we have two rings operating simultaneously, to obtain the total network through-put we must substitute $\frac{1}{2}S$ for S in the equation. As an illustration of the performance improvement to be gained from employing two contrarotating rings, Figure 4.1 shows a graphical comparison of single and dual buffer insertion rings. The ordinate represents the average message delivery time, and the abscissa the total network throughput (normalised to the individual link data rate.) The parameters used were $M = 50$ and $a' = 0.1$.

One may observe that, in the case of the single ring, the maximum throughput expected is double that of the data rate. For the dual ring, this figure increases to 8 times the base data rate. Performance gains achieved by using two contrarotating rings, then, may be considerable. However, the results do depend significantly on the parameter α, as discussed above. One
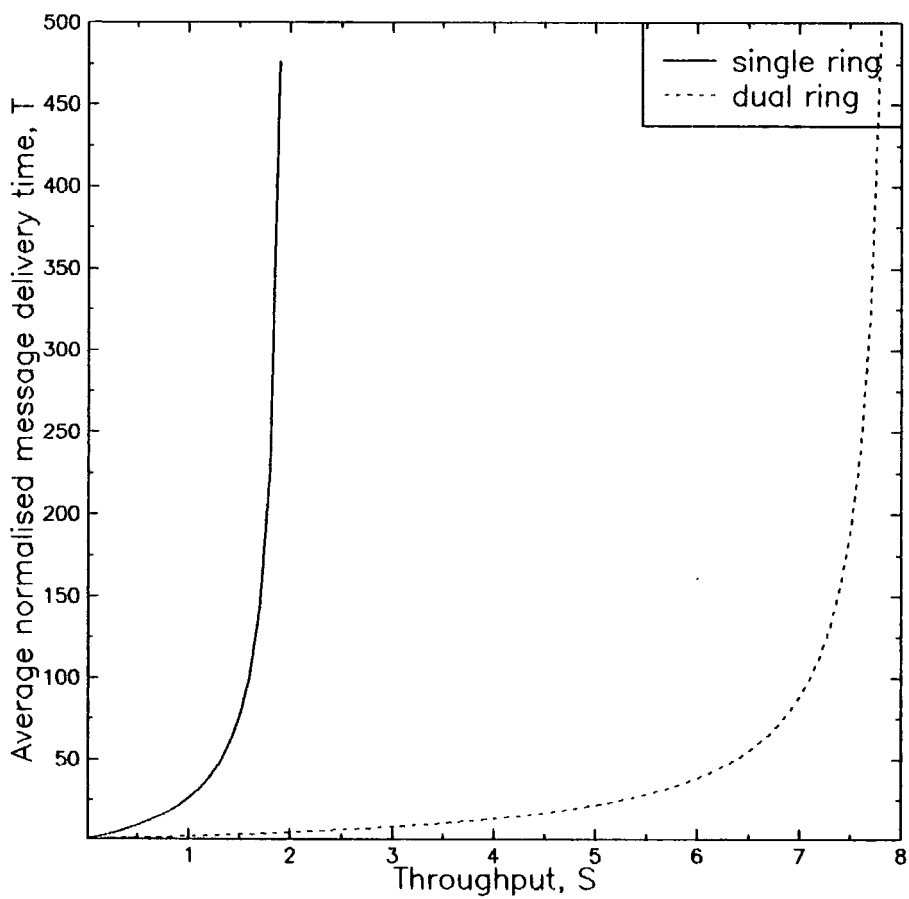
Figure 4.1: Performance of single and dual BI rings

would expect that, by using the quickest path routing strategy, α would be around half its value in the single ring case, or perhaps even less; this may not be the case however, and simulation should provide a clearer illustration of its significance.

4.2 Measurement

4.2.1 Introduction

In this section, actual measurements on real networks were used to evaluate the performance of the buffer insertion protocol. In this case only a single ring LAN was examined, since a dual ring version was not available in the laboratory. The first set of experiments were carried out at the Research and Development Laboratories of Nine Tiles Computer Systems Ltd., Cambridge, with the assistance of Mr J. Grant (Technical Director). They were largely concerned with the low-level details of the network hardware. The second set of experiments took place at the Networks Laboratory at Teesside Polytechnic. These involved measurements at a higher level, typically involving networks of up to 20 nodes.

4.2.2 Low-level experiments on buffer insertion ring LANs

Introduction

This section presents a summary of the results obtained from experiments on the Superlink network, which were performed at Nine Tiles Computer Systems, Cambridge. The work was carried out for two reasons: as an evaluation of the performance of Superlink, and to provide data with which to verify the Superlink simulation program (see the following section.) Each experiment usually involved the measurement of the amount of time to deliver a message or establish a connection. The equipment used comprised of the following:

- A digital counter/timer with a resolution of 10 μs and two inputs, 'start' and 'stop'. A TTL high-to-low transition at the 'start' input caused the timer to begin counting, and a high-to-low transition at the 'stop' input stopped the count, thus displaying the elapsed time between the two events.

- An 8 channel logic analyser with a 16 bit word recogniser probe and an RS232 serial input. This allowed the capture and storage of 8 bits of time-varying parallel data, or a single stream of data through the

RS232 port. A trigger signal was produced at an output when a set
pattern of 24 parallel bits or 16 serial bits was present at the inputs.
Thus a particular piece of data within a network node could trigger the
logic analyser, which would then start or stop the timer.

o Two personal computers fitted with network cards, two stand-alone
RS232 network interface boxes, and a rack-mounted set of 16 RS232
network interface cards. Also available were various cables and a dumb
VDU terminal. These were connected in various combinations to form
the networks used in the experiments.

In the following sections the experiments and their results are described
individually. However, a few general comments can be made. The usual form
of an experiment was to set up a particular configuration, write some software
to perform a particular network function (e.g. send a message from one node
to another), then use a combination of timer and logic analyser to measure
the duration of this function. The measurement was usually repeated so
that a total of 50 samples was taken. This was necessary because the times
involved differ slightly in each repetition of the same experiment, due to the
nature of the processing in the nodes. An average value was taken when
plotting the graphs. In addition, each experiment was performed twice; once
for the 'fast' ring speed (1.5 M bits/sec data rate), and again for the 'slow'
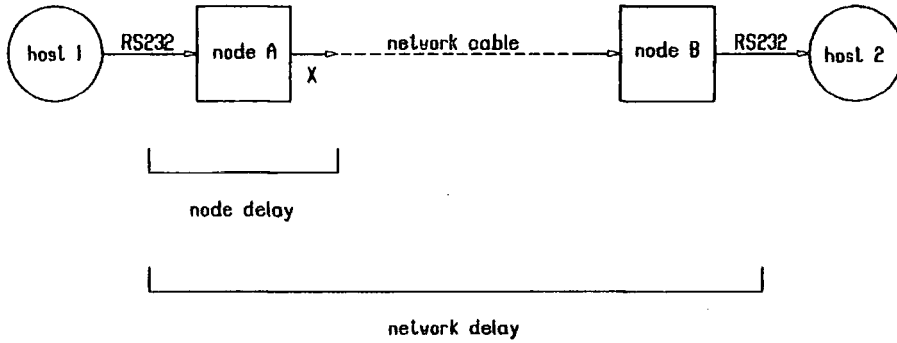
ring speed (250 k bits/sec).

Figure 4.2: Node delay and network delay contrasted

Two terms used throughout the discussion are 'node delay' and 'network delay'. 'Node delay' refers to the processing time in one network interface unit (referred to here as just 'node'), excluding transmission times over the network cables or RS232 links. 'Network delay' is the total network time delay, including processing times in the nodes and delays in the cable. For example, in Figure 4.2 a message is being transmitted from node A to node B. In this situation, the node delay for A would be the time between the start of the message entering A's host/node interface via the RS232 link, and the start of the message being output onto the network cable at point X. The

network delay would be the time between the start of the message entering A's host/node interface via the RS232 link, and the start of the message emerging from B's host/node interface via the other RS232 link.

Each of the eight experiments is described below with a brief explanation of the method and the results. A summary is shown in Figure 4.3. Graphs are referred to which depict the mean values of the samples recorded for each measurement.

Expt no.	Network function	Measured parameter
C1	connection establishment	node and network delay
C2	connection closure	node and network delay
C3	data transfer	node delay
C4	data transfer	network delay
C5	data transfer	throughput
C6	data transfer through gateway	network delay
C7	data transfer through gateway	throughput

Figure 4.3: Experiments performed in Cambridge

Experiment C1: Connection Requests

The aim of this experiment was to measure the delay times involved in establishing a connection between two nodes. In part (a) the node delay was measured for a particular node when a remote host requests a connection to the local host. The apparatus was arranged as in Figure 4.4.
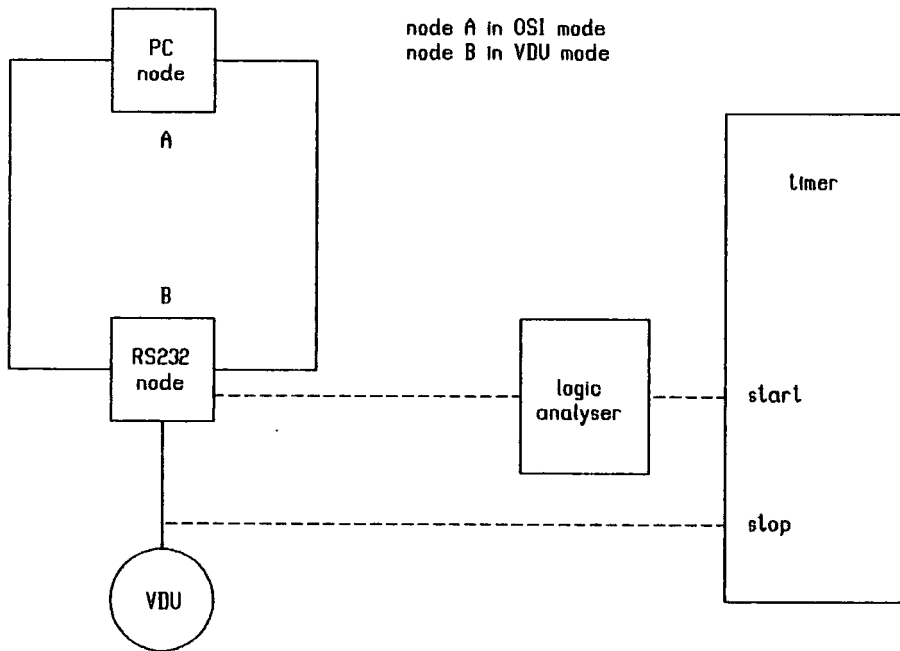
Figure 4.4: Apparatus for connection request experiment (C1)

The logic analyser was connected to node B's internal RAM buffer (data bus) and configured to trigger when the connection request packet was copied from the ring into this buffer. (The analyser detects the last character of B's name, which is enclosed in the connection request packet.) This event starts the timer, which stops counting when the first character of the 'incoming call connected' message is sent to the VDU. A histogram (Figure 4.5) shows the distribution of these times.

Part (b) is the reverse situation of part (a); the VDU attempts to establish a connection with the PC. The timer starts when a signal appears on B's host/node interface, i.e. when the connect command is typed from the VDU (in line-by-line mode). The logic analyser is connected to node B's address bus; when the routine which sends the connection request packet onto the ring is executed, the analyser detects its start address on the address bus and stops the timer. Thus the processing time for the connection request is measured. This includes the time for the host to transmit the connect command to the node; this was determined by calculation (5 characters, in 10 bit frames, at 9600 baud) and subtracted from the measured time, to give the node delay only. These results are displayed in Figure 4.6. It should be pointed out here that the events detected by the logic analyser (i.e. RAM/ROM signal changes) do not correspond exactly to the required network events (i.e. incoming/outgoing data on the ring). Hence the measured

values of node delay are close approximations and not exact values.

Finally, part (c) measured the total network delay involved in establishing a connection. Node B was set to dumb mode, and the timer was started on the connection command being sent from the host. After the connection to node A was established, an ACK (acknowledge) message was sent to the VDU, and this event stopped the timer. This experiment was repeated for different numbers of nodes in between the two nodes attempting connection establishment, and the graph in Figure 4.7 was plotted.

Due to limited space on the abscissa of the histograms, the labelling has been compressed so that the number below each bar refers to the lower limit of the interval. For example, the bar above the number 3.2 (Figure 4.5) represents the number of node delay measurements which were greater than (or equal to) 3.2 and less than 3.3.

Figures 4.5 and 4.6 are rather as expected, apart from certain peaks (e.g. 3.4–3.5 range in Figure 4.5). The delay is produced by the software running on the Z80 processor in the node. This repeatedly executes a loop to check the status of such things as the incoming ring and the host/node interface, and acts accordingly. Some functions have priority over others in order that the ring may operate correctly. Thus the measured times had
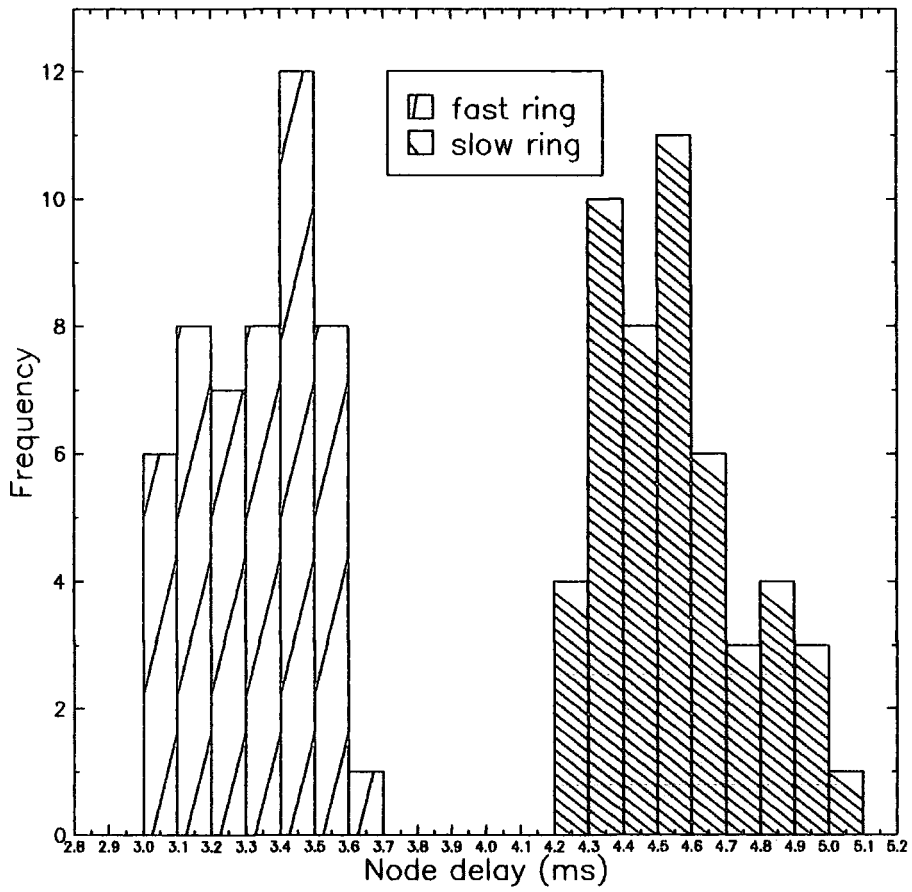
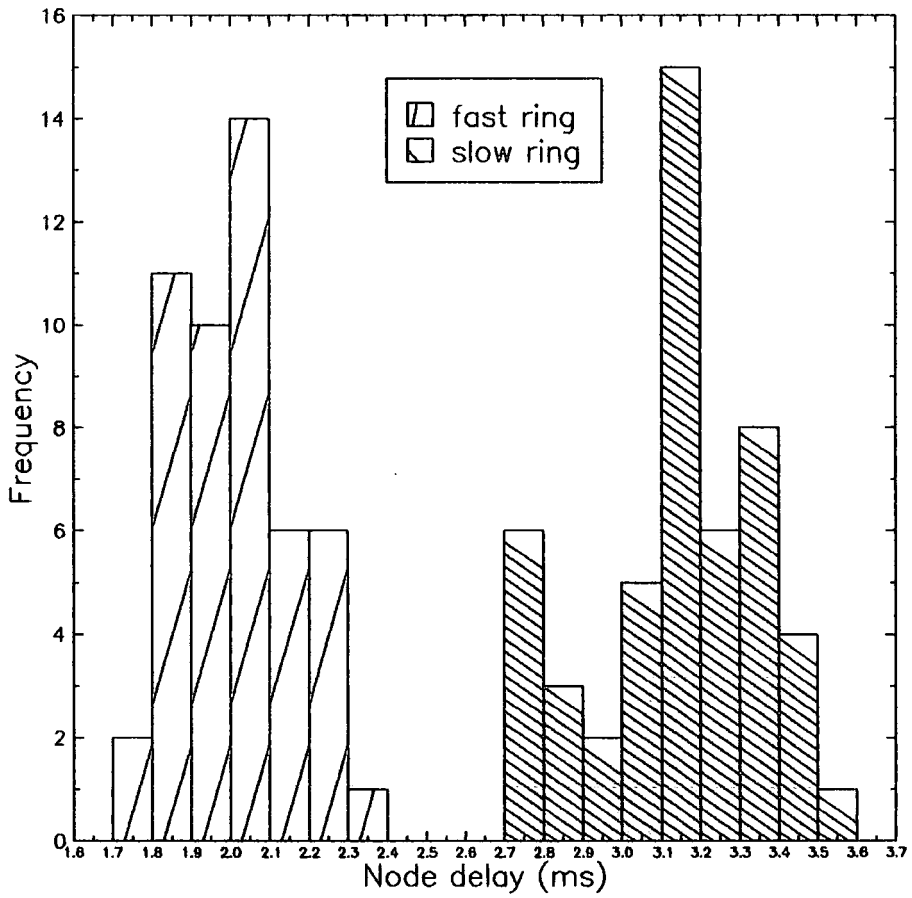Figure 4.5: Histogram of node delay for receipt of connection request

Figure 4.6: Histogram of node delay for transmission of connection request

44

different values because the software was at a different stage in its processing when the connection command was issued. The peaks in the graphs represent the most frequently occurring times. The other smaller bars, especially those representing relatively large values of delay time, may be explained thus: Every so often the node receives housekeeping information from the ring, which it must act upon with greater priority than an incoming call from the host. If the host sends a connection command at this moment, the command will have to wait until the housekeeping information is processed; hence the greater than average node delay time.

Figure 4.7 suggests that the network delay for connection establishment is a linear function of the number of nodes between the two nodes negotiating the connection (i.e. between sender and receiver in a 'downstream' direction). As would be expected, the delay for the slow ring is greater than that for the fast ring.

Experiment C2: Disconnection Requests

Experiment C2 was identical to Experiment C1 except that the delays for disconnection requests were measured (after previously establishing a connection). The results are presented as Figures 4.8, 4.9 and 4.10.
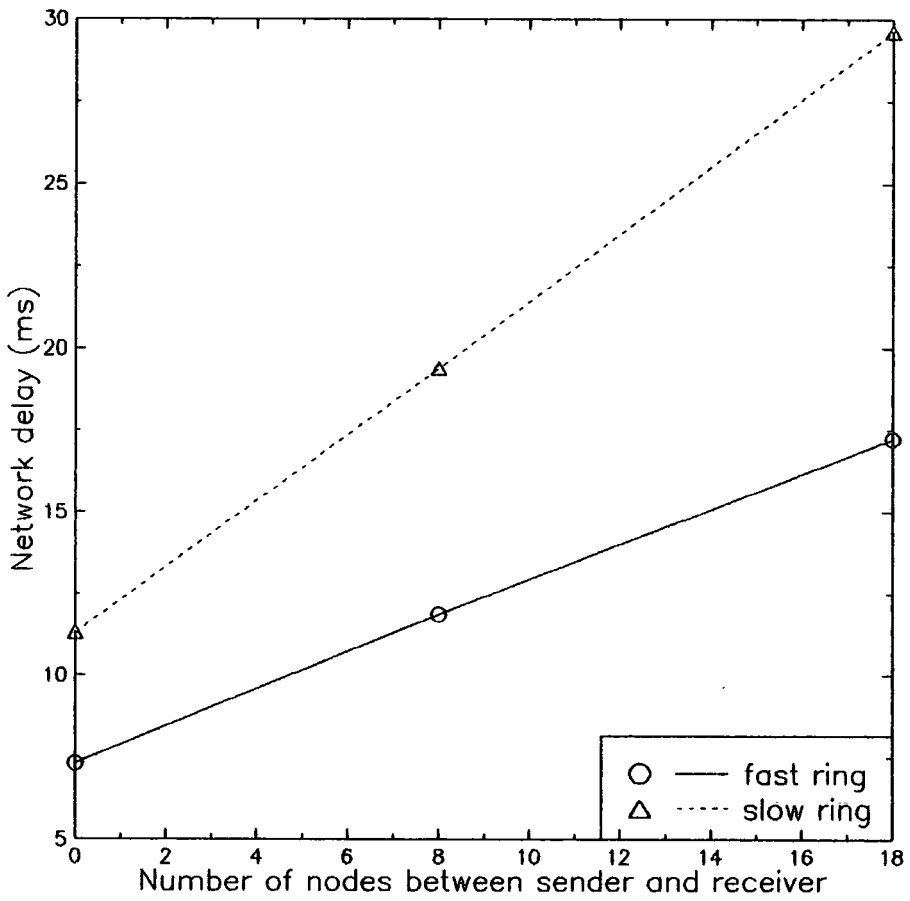
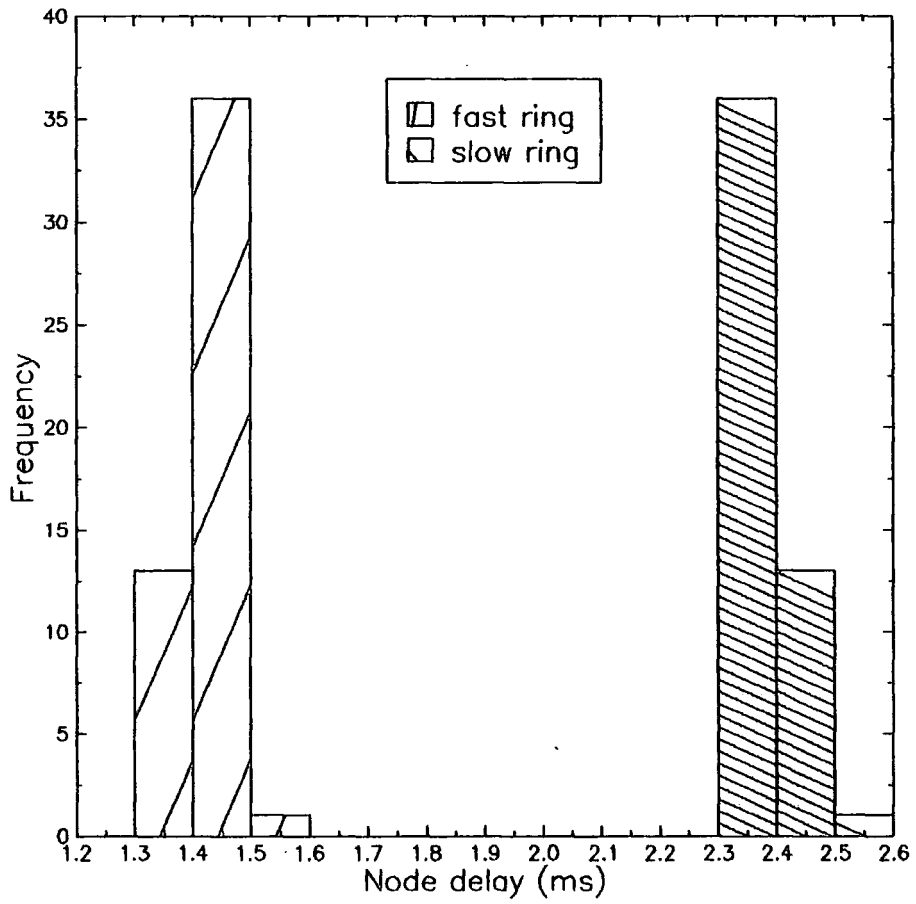Figure 4.7: Network delay (connection request) / number of nodes between sender and receiver

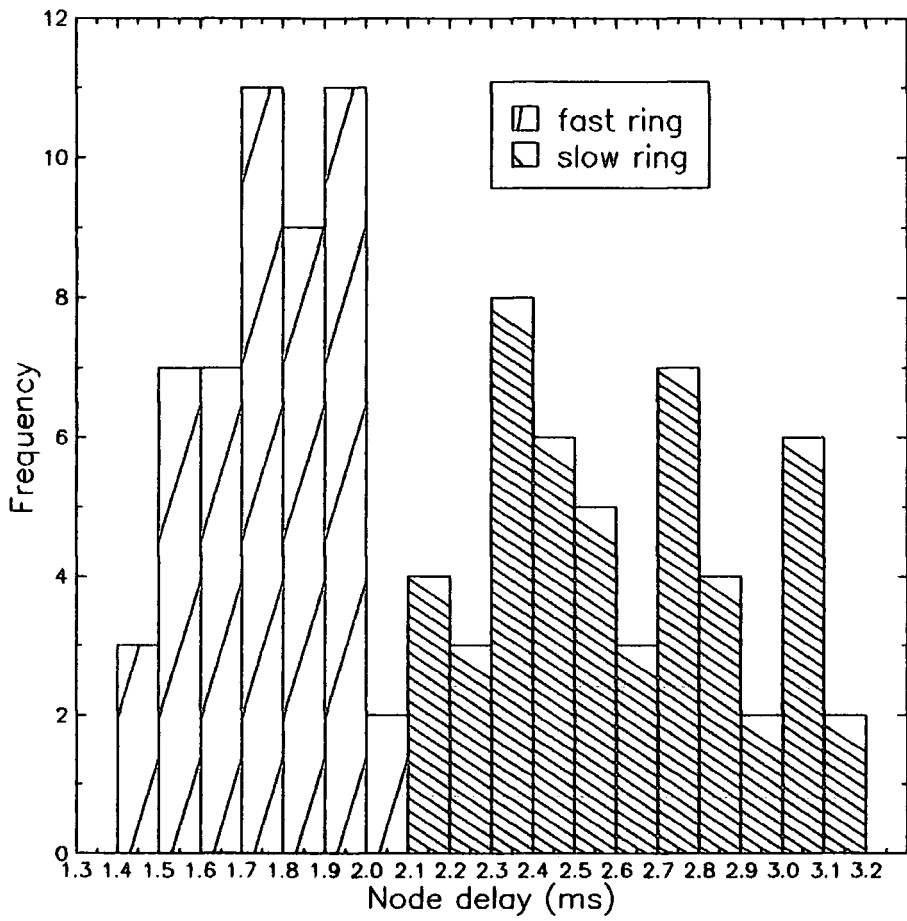Figure 4.8: Histogram of node delay for receipt of disconnection request

Figure 4.9: Histogram of node delay for transmission of disconnection request
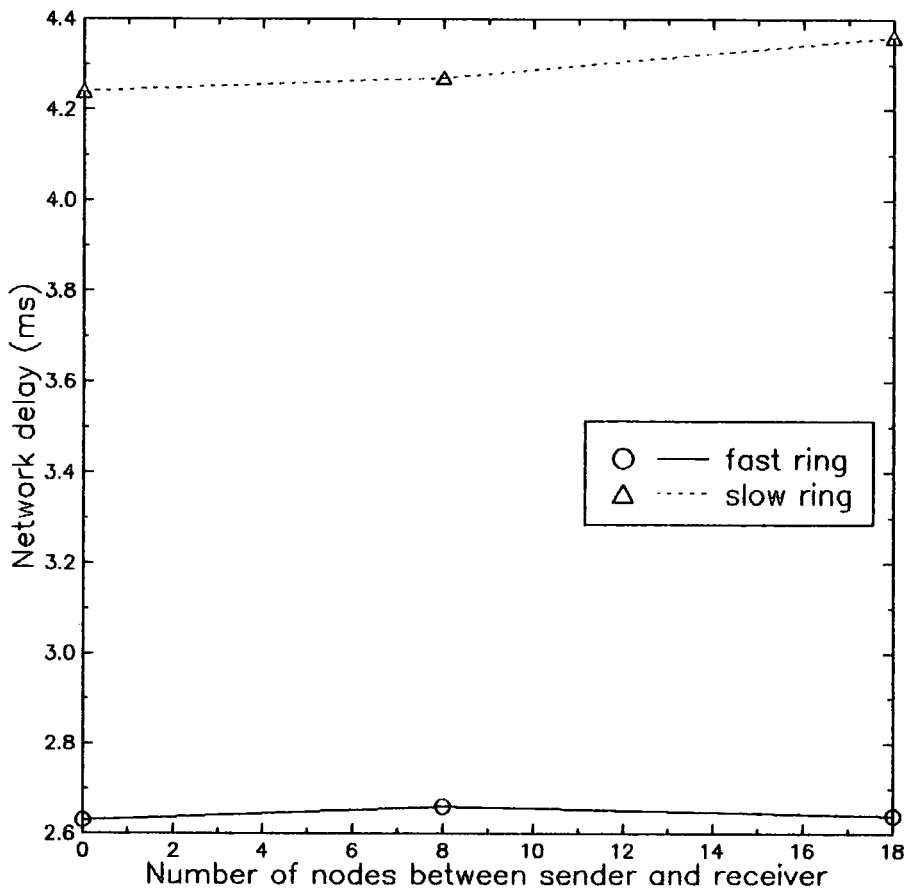
Figure 4.10: Network delay (disconnection request) / number of nodes between sender and receiver

The node delay times for the incoming disconnection requests (Figure 4.8) have a much smaller spread than those for incoming connection requests, yet those for outgoing disconnection requests (Figure 4.9) are not dissimilar to those for outgoing connection requests. The network delay graph (Figure 4.10) shows that, compared to connection requests (Figure 4.7), disconnection times have a much smaller magnitude and increase less rapidly with the number of intervening nodes. Indeed, the latter quantity seems to have little effect on the delay time (at least for under 18 nodes between sender and receiver).

Experiment C3: Node Delay for Data Traffic

An apparatus was used similar to that shown in Figure 4.4. Part (a) involved sending a one-character message from the PC to the VDU. The logic analyser and timer were set up as before, and the analyser was programmed to detect the character in the message (lower case 'A') passing into the node's buffer from the ring. The measured time was the node delay for B when a single character is received. After 50 samples the experiment was repeated but using varying message lengths, each message consisting of up to 79 spaces followed by the lower case 'A'. A graph (Figure 4.11) of node delay versus message length was plotted.

For part (b) the sender and receiver were reversed, so that the VDU sent characters to the PC. The timer was started on the character being sent from the VDU to node B, and stopped when the 'send' routine was called in the node's internal ROM software. The message length was varied by using a PC running a terminal emulator program instead of the VDU.
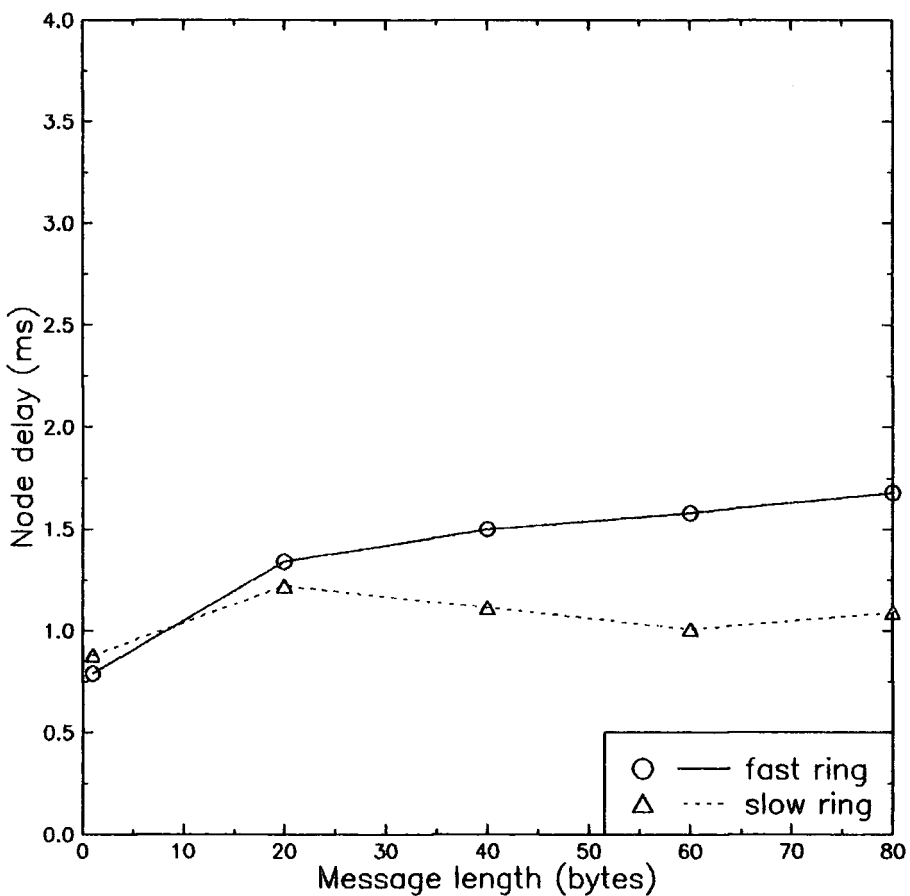

Figure 4.11: Node delay / message length (data from ring to host)

Figure 4.11 shows that, for the fast ring, the node delay increases with message length. This is as expected; the more characters in the message, the
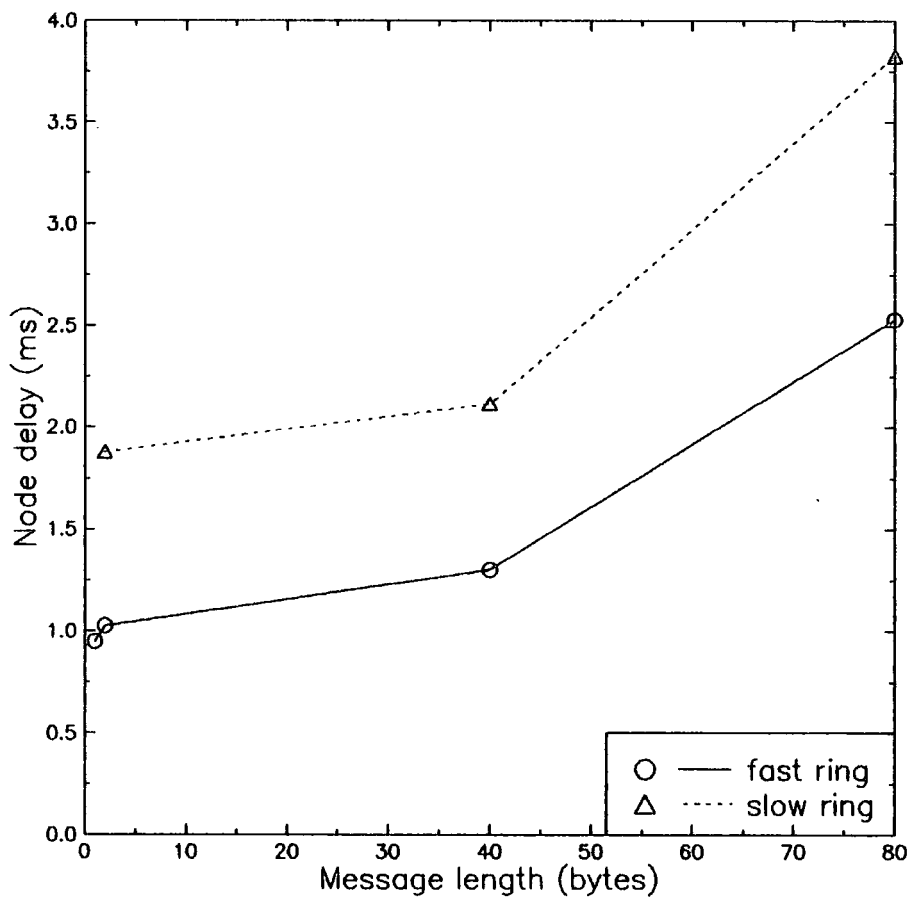
Figure 4.12: Node delay / message length (data from host to ring)

longer it will take to process and copy them around inside the node. One would expect the line for the slow ring to be similarly monotonically increasing; its actual shape is most probably due to variance in the data. Figure 4.12 is rather more readily understood; the delay increases monotonically with the message length.

Experiment C4: Network Delay for Data Traffic

This experiment determined the network delay for a fixed length (1 character) message passing between two hosts. The network activity occurring separately from this message (referred to here as the background activity) was varied in order to observe the network performance under different levels of loading. The apparatus configuration is shown in Figure 4.13.

A set of rack-mounted RS232 nodes was connected in a ring containing 2 PCs and 2 RS232 nodes whose hosts were VDUs. The rack-mounted nodes were configured such that each pair of adjacent nodes had their host/node interfaces connected together. Thus each pair of these nodes formed a gateway back onto the ring, with a baud rate set to 250k. The two VDUs were used to send and receive a test message (the character 'a'), whose network delay was measured by starting the timer when it emerged from the source VDU and stopping the timer when it entered the destination VDU. The two
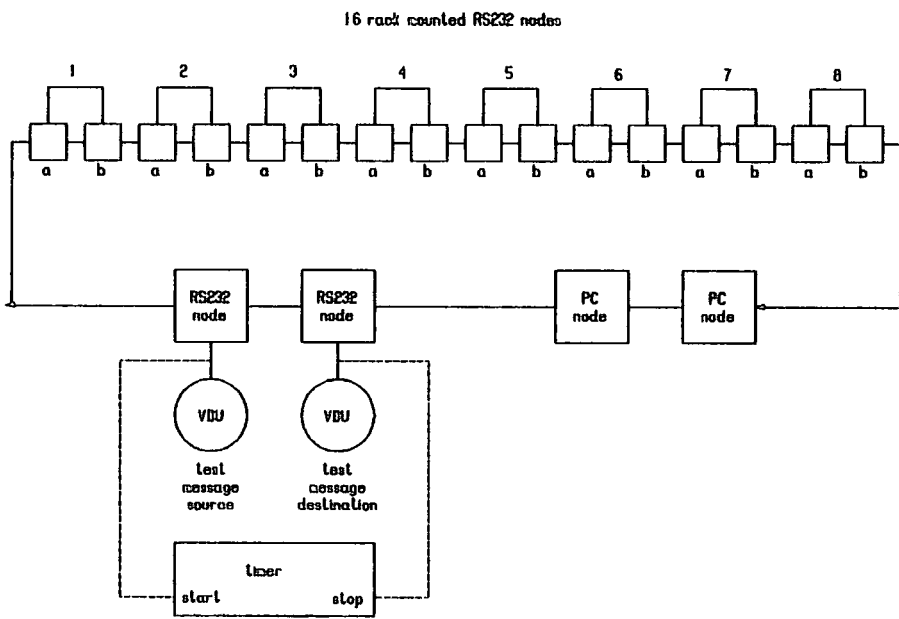
Figure 4.13: Network configuration for Experiment C4

PCs were used to provide the background traffic. The destination PC (network name: 'tandon') ran a program which offered a service called 'sink'; this merely accepts incoming messages and discards them. Using a separate program, the source PC sent 80-character messages to the destination PC. The messages could travel along a user-specified path across the network. Both programs used a 2-buffer method to maximise message throughput.

The amount of background activity was varied by sending the background messages across different network paths. The choice of path affects such factors as how often a message is regenerated through a gateway, how long a message takes to travel around the ring, and so on. Five different levels of background activity were used, referred to by the letters A to E. Corresponding paths, from the source PC to the destination PC (tandon), are shown below in order of increasing background activity. Each background message originates at the 'background traffic source' PC, passes through the list of nodes delimited by '/', from left to right, and is finally removed from the ring by the 'sink' service.

A   none, i.e. no background traffic generated

B   /tandon/sink

C   /1a/2a/3a/4a/5a/6a/7a/8a/tandon/sink

D   /8b/7b/6b/5b/tandon/sink

Part (a) of the experiment measured the network delay for sending the test message across the network shown in Figure 4.13, with varying levels of background activity. A bar chart (Figure 4.14) shows the results. For part (b), the same method was used, but the positions of the test message source and the test message destination were swapped; thus the destination was immediately 'downstream' of the source.

The network delay rises with increasing background activity when 18 nodes are between sender and receiver (Figure 4.14). From Figure 4.15 it would appear that background activity has little effect on the delay when the sender and receiver are adjacent on the ring, at least for the fast ring. Bar charts have been used for these graphs as the background activities used have discrete levels. It is difficult to say exactly how the network delay varies with activity, as the latter involves several components, such as the placement of the nodes, the amount and frequency of traffic generated by each node, the path taken by the messages and the message length. Further work could be done to isolate each of these quantities and perform separate experiments where one quantity is varied while the others are held constant. The results do however provide useful information about the network performance.
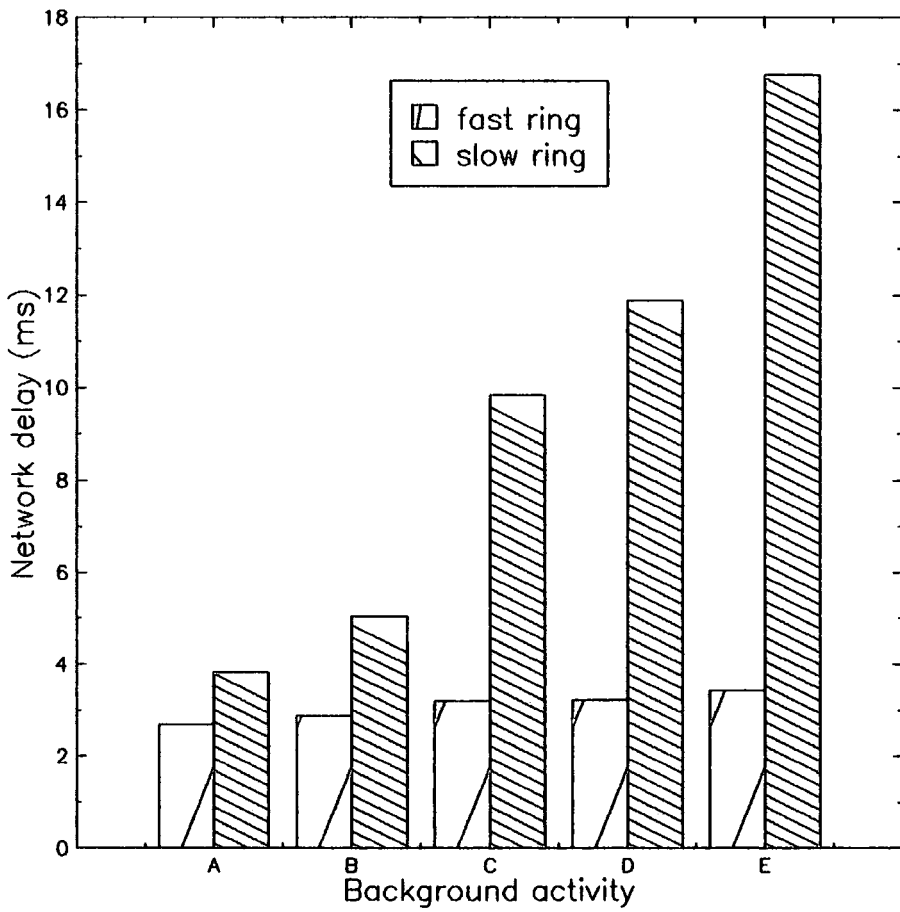
Figure 4.14: Network delay / background activity (18 nodes between sender and receiver)
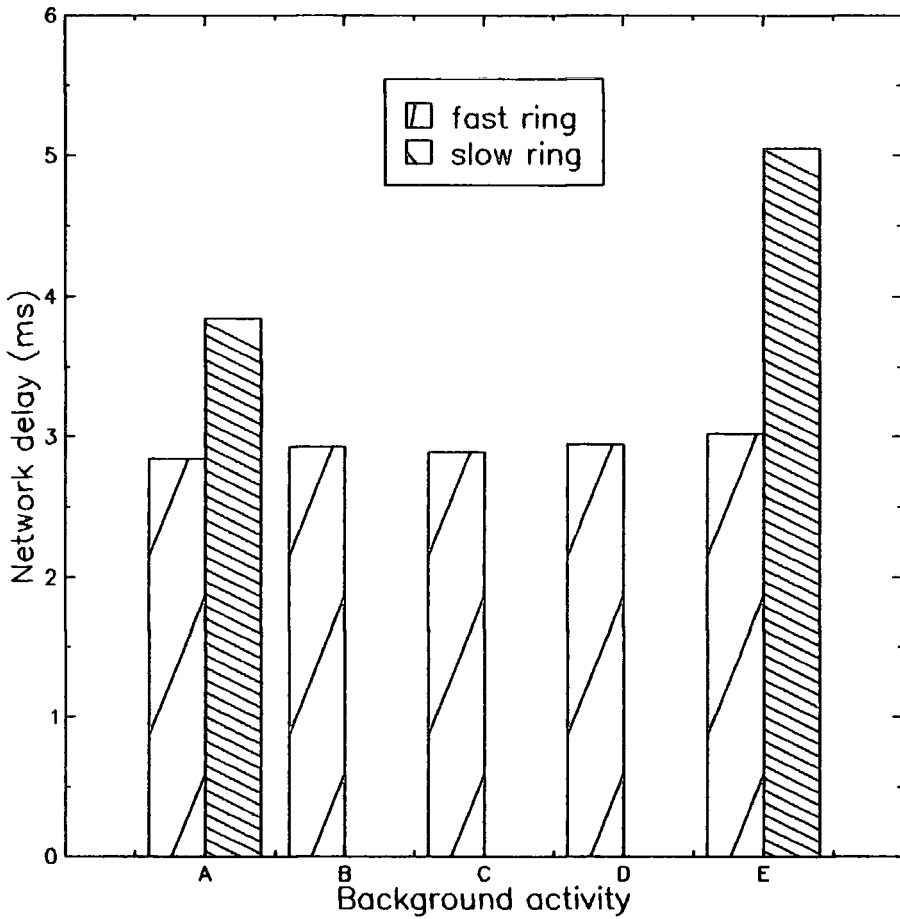
Figure 4.15: Network delay / background activity (0 nodes between sender and receiver)

Experiment C5: Throughput for Data Traffic

This experiment was very similar to Experiment C4, but the test message was not sent; instead, the background traffic alone was used. The time to send a fixed number of these 80-character messages was measured, then the throughput (in messages per second) was calculated by dividing the number of messages by the time. This procedure was performed for each of the different paths (B to E) through the network.

Figure 4.16 shows a bar chart of the throughput versus the path. The paths which corresponded to low background activity in Experiment C4 are those which allow the greatest throughput. The fact that the maximum throughput for a fast ring is much lower than its theoretical value (2344 mes/sec for 80 bytes at 1.5 M bits/sec) indicates the great effect of the node delay and framing information on the network performance.

Experiment C6: Network Delay for Dual Interconnected Rings

This experiment investigated a configuration of two independent networks which were connected via a gateway, as shown in Figure 4.17.

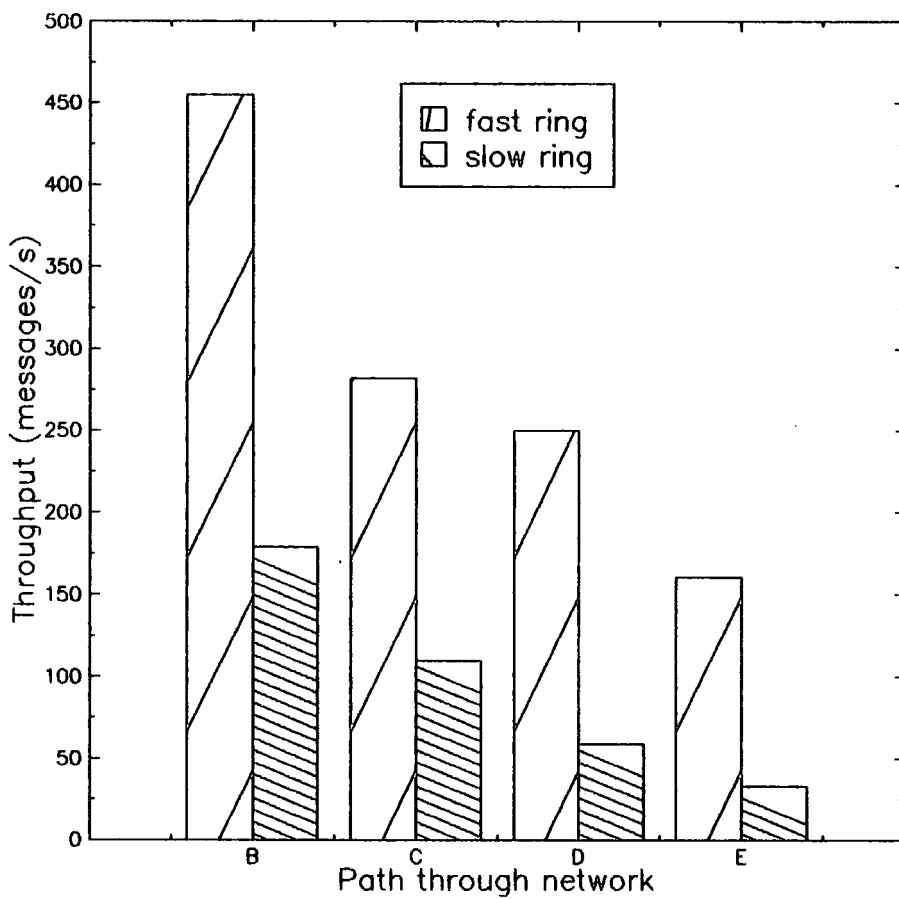The numbered rectangles represent components of the rack of RS232

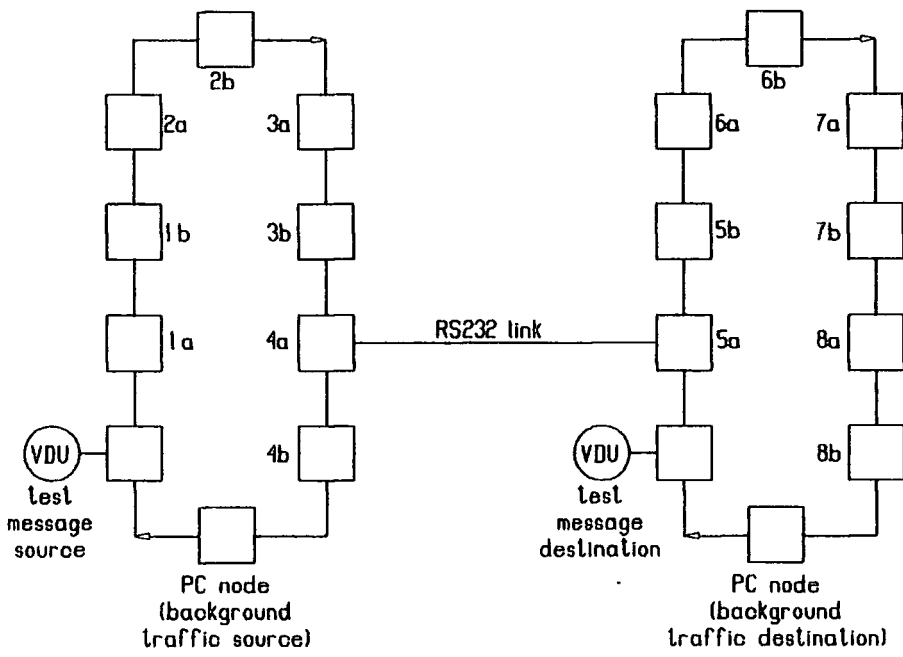Figure 4.16: Throughput / path through network

Figure 4.17: Dual interconnected rings

nodes, which in this case are not used as gateways back onto the ring, but are left inactive. The host/node interfaces of nodes 4a and 5a are connected together such that, operating in OSI mode ('secure' format), these two nodes form a gateway between the two rings. The speed of the gateway was set using a built in configuration service.

The PCs used the program from Experiment C4 to send 80-character messages from the left ring to the right ring via the gateway, as quickly as possible. These messages were used as background traffic. Once again, the test message was one character (letter 'a') sent from the left VDU to the right VDU. The network delay was the time between the test message leaving the source VDU and entering the destination VDU. This was measured with and without the background traffic, and for different gateway speeds.

Figure 4.18 shows how the network delay decreases with increasing gateway speed. The graph shows that the delays are longer if background traffic is present, but the shapes of the graphs are similar in both cases.

Experiment C7: Throughput for Dual Interconnected Rings

The previous experiment was repeated, but instead of using a test message the traffic between the two PCs alone was used. The time for a fixed number

Figure 4.18: Network delay / Gateway speed

of 80-character messages to be delivered across the gateway was measured, and the throughput was calculated from this. The gateway speed was varied and the experiment repeated.

Figure 4.19: Throughput / gateway speed

Figure 4.19 shows how the throughput is limited by the gateway speed. The two lines coincide up to a gateway speed of about 110 k baud, as this speed is much lower than the network data rate. However, as the gateway speed exceeds the slow ring data rate (250 k baud) the lower line begins to

level off due to the data rate becoming the limiting factor. The line for the fast ring keeps increasing, but one could expect it too to level off at about 1500 k baud (fast ring data rate).

Summary

o Node delays associated with data transfer and opening and closing connections are significant. They can be an order of magnitude greater than the time to transmit a typical 80 byte packet.

o Node delay for data tends to increase with the length of the data packet, whereas node delay for connection/disconnection signals has a more random nature.

o Network delay for data on the slow ring varies considerably with the overall load on the network. Typical ranges are 4–16 ms. The fast ring exhibits a much smaller variation.

o Where two rings are connected by a gateway, the speed of the gateway is the limiting factor (unless the gateway speed is greater than the ring data rate).

4.2.3 Experiments on an academic PC network

Introduction

This section presents an account of LAN performance measurement experiments carried out at Teesside Polytechnic Networks Laboratory. The purpose of these experiments was, firstly, to obtain statistical results which would describe a network's performance under certain conditions and, secondly, to compare the performance of different types of network. These experiments are generally on a larger scale than those performed in the Cambridge laboratory, being less concerned with the low-level details of the nodes.

Two types of LAN were investigated: a Nine Tiles Swiftlink (buffer insertion protocol) and a BICC Ethernet (CSMA/CD protocol [22]). Each network consisted of up to 10 nodes, in the form of cards which were fitted into Opus PC5 computers. All the PCs were of the same specification, as were the network cards. The software interface to both LANs was standard NETBIOS, allowing programs written to access one network to be used to access the other with only one minor alteration.

The experiments performed are described in this section, together with a discussion of the results obtained.

To assess the performance of a network one must first choose a parameter to measure. These parameters are different to those used in Section 3.2.1 because the overall network performance is being assessed in this case. This choice is partly influenced by the feasibility of actually making the measurement. For these experiments the chosen parameter was the average maximum throughput (AMT), i.e. the average maximum number of message bits that a PC on the network can send to another networked PC in one second. In most cases the way this was measured is as follows. Each node runs a program which attempts to send messages of a certain fixed length to a certain fixed destination as quickly as possible, by requesting another 'send' command to the NETBIOS as soon as the previous 'send' command has completed; this maximises the throughput. When a given number, n, of messages has been sent, the performance statistics are calculated. A clock internal to the PC, with a resolution of approximately $\frac{1}{18}$ of a second, is used to note the time when the sending of messages begins, and also the time when n messages have been sent. Hence n messages are delivered in a measured time of t seconds. The AMT is then calculated by dividing n by t and multiplying by the message length.

The AMT gives a good representation of performance, because it is inde-

pendent of the message length and it is measured across a reasonably long period of network activity, which tends to minimise the effects of transients or poor clock resolution. It is also a useful parameter for a communications system designer, who, given a certain required throughput for an application, can look at results graphs for the different networks and choose the one with the most appropriate performance. The AMT was also the most practical parameter to measure; to measure each individual message's delivery time would have required a highly accurate clock.

In order to give a fair comparison between the two LANs, the results must be expressed in terms of a parameter which relates performance to link capacity. A suitable parameter is the normalised average maximum throughput (AMT_n), which is simply the AMT divided by the physical data rate. It is necessary to present the results in this way because, as the data rate of Ethernet is over 6 times that of Swiftlink, Ethernet unsurprisingly always has a higher AMT. To compare the relative merits of CSMA/CD and buffer insertion we need to eliminate this difference in data rate. All the graphs shown in this section are thus plotted with AMT_n on the ordinate. To convert back to AMT, one simply multiplies by the data rate (1.5 Mbps for Swiftlink, 10 Mbps for Ethernet).

From the results (except Figure 4.30) it will be seen that the performance

of Swiftlink surpasses that of Ethernet. This is most probably due to the different media access methods, but it could also be affected by the processing speed of the software interface to the network, i.e. NETBIOS; though the user interface appears the same in both systems, the NETBIOS software's internal operation is different in each case. Additionally, some cheaper Ethernet implementations rely on the host processor doing more work. However, it must be remembered that the results are normalised to the data rate; in absolute terms, Ethernet gives the greater throughput.

Figure 4.20 shows a summary of the experiments; details on each are given below. Figure 4.21 shows a brief technical specification of each network.

Expt no.	Network	Variable parameter	Active nodes	Message destination
T1	Swiftlink	Message length	1	successor
T1	Ethernet	Message length	1	successor
T2	Swiftlink	Message length	10	successor
T2	Ethernet	Message length	10	successor
T3	Swiftlink	Message length	10	predecessor
T3	Ethernet	Message length	10	predecessor
T4	Swiftlink	Message length	10	self
T4	Ethernet	Message length	10	self
T5	Swiftlink	Number of nodes	10	successor
T5	Ethernet	Number of nodes	10	successor

Figure 4.20: Experiments performed on Swiftlink and Ethernet

Parameter	Swiftlink	Ethernet
Medium access method	buffer insertion	CSMA/CD
Data rate	1.5 Mbps	10 Mbps
Software interface	NETBIOS	NETBIOS
Maximum packet length	86 bytes	1464 bytes
Total cable length (estimate)	500m	160m
Number of nodes	10	10

Figure 4.21: Network Parameters

Experiment T1

This experiment was performed in order to measure the maximum AMT for one node. Hence only one node is active, sending to its physical successor in the ring; no other nodes generate traffic. (In the case of Ethernet, terms such as 'successor' and 'predecessor' have no physical meaning, as this LAN has a bus topology.) Thus the one active node has the whole of the bandwidth to itself, and its data will not encounter any collisions (in the case of Ethernet) or through-traffic (in the case of Swiftlink). The throughput is a maximum in this situation.

The message length has been used as the variable parameter in several experiments. This is because varying the message length is a convenient way of varying the load on the network, due to the way in which the application program and the NETBIOS software interact. Figure 4.22 shows the main elements in the networking system.
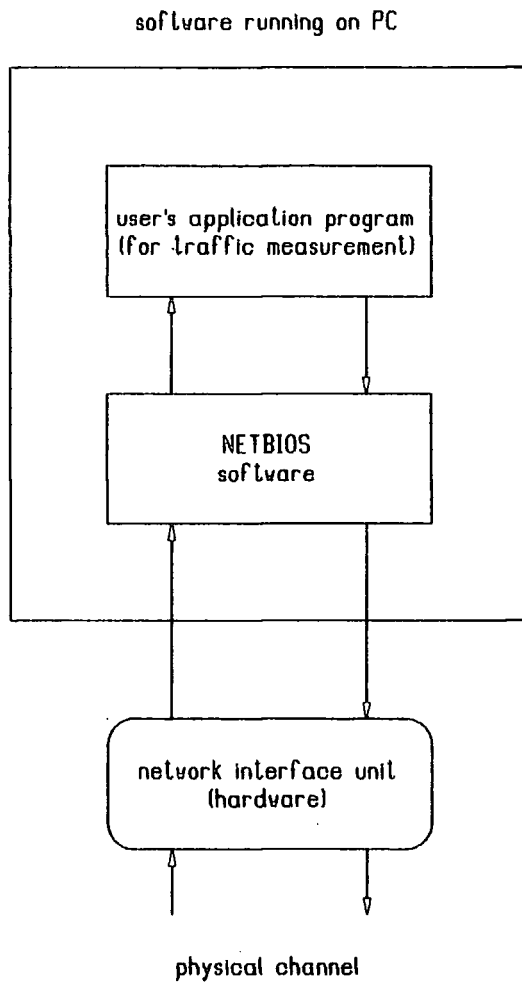
70

software running on PC

```
┌─────────────────────────────────────────────┐
│                                             │
│        ┌─────────────────────────┐          │
│        │ user's application program│         │
│        │ (for traffic measurement) │         │
│        └─────────────────────────┘          │
│              ↑           ↓                   │
│        ┌─────────────────────────┐          │
│        │        NETBIOS          │          │
│        │        software         │          │
│        └─────────────────────────┘          │
│              ↑           ↓                   │
└─────────────────────────────────────────────┘
               ↑           ↓
        ╭─────────────────────────╮
        │  network interface unit │
        │       (hardware)        │
        ╰─────────────────────────╯
               ↑           ↓

          physical channel
```

Figure 4.22: Networking software and hardware in each node

71

With reference to Figure 4.22, if the application program wishes to send a message, it issues a send request to the NETBIOS. It must then wait while the NETBIOS sends the appropriate instructions to the network interface unit and the bits are transmitted across the physical channel. When this is complete the application may issue another send request. Hence after each message is sent control returns to the application program, which must do some processing to prepare the next message for transmission. This processing wastes a certain amount of time, which could have been used to transmit data on the network; in effect, the less time is spent executing the application program, the more efficient is the use of the network and the higher is the throughput. So if the message length is short, the processor will spend relatively large amounts of time running the application program and less time running the NETBIOS, leading to a low throughput (low loading). If the message length is long, little time will be spent on the application program, and more time will be spent running NETBIOS, leading to a high throughput (high loading).

In addition to this effect, the influence of the packet header length on the throughput must also be considered. This can be demonstrated as follows. Using the following definitions

R = data rate

L = data length

h = header length

then the time to send a packet, T_p, is

$$T_p = \frac{L + h}{R} \qquad (4.4)$$

Thus the rate of sending packets, S_p, can be expressed as

$$S_p = \frac{R}{L + h} \qquad (4.5)$$

Multiplying by the data length gives the rate of sending data bits (i.e. the throughput), S_d

$$S_d = \frac{RL}{L + h} \qquad (4.6)$$

$$S_d = \frac{R}{1 + \frac{h}{L}} \qquad (4.7)$$

A sketch graph of this expression is shown in Figure 4.23. This is the form of graph which one would expect from the experimental results. The throughput would never actually reach the data rate, of course – it is limited by the processing speed of the NETBIOS.

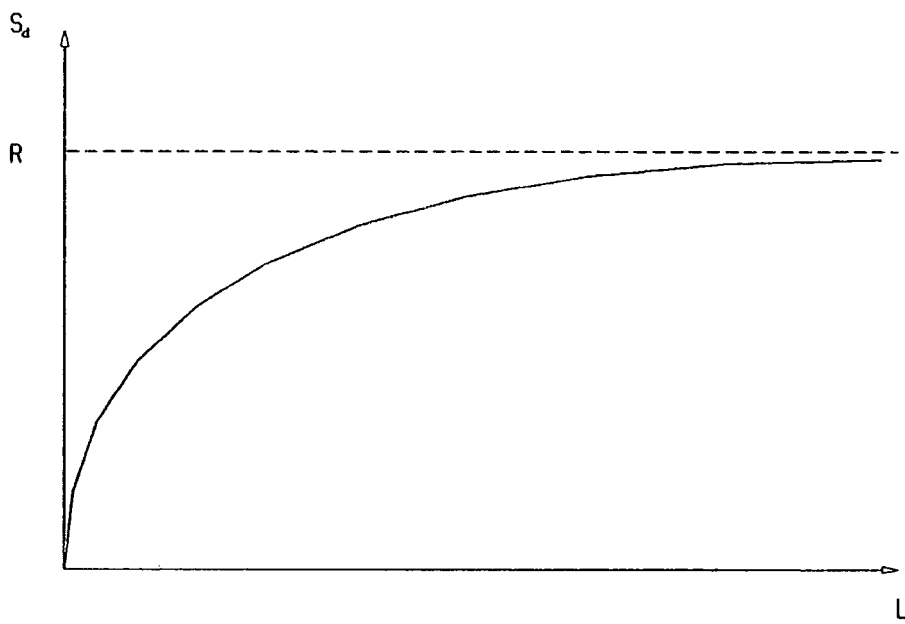From Figures 4.24 and 4.25 it can be seen that, as predicted, the through-

73

Figure 4.23: Theoretical graph of throughput versus message length
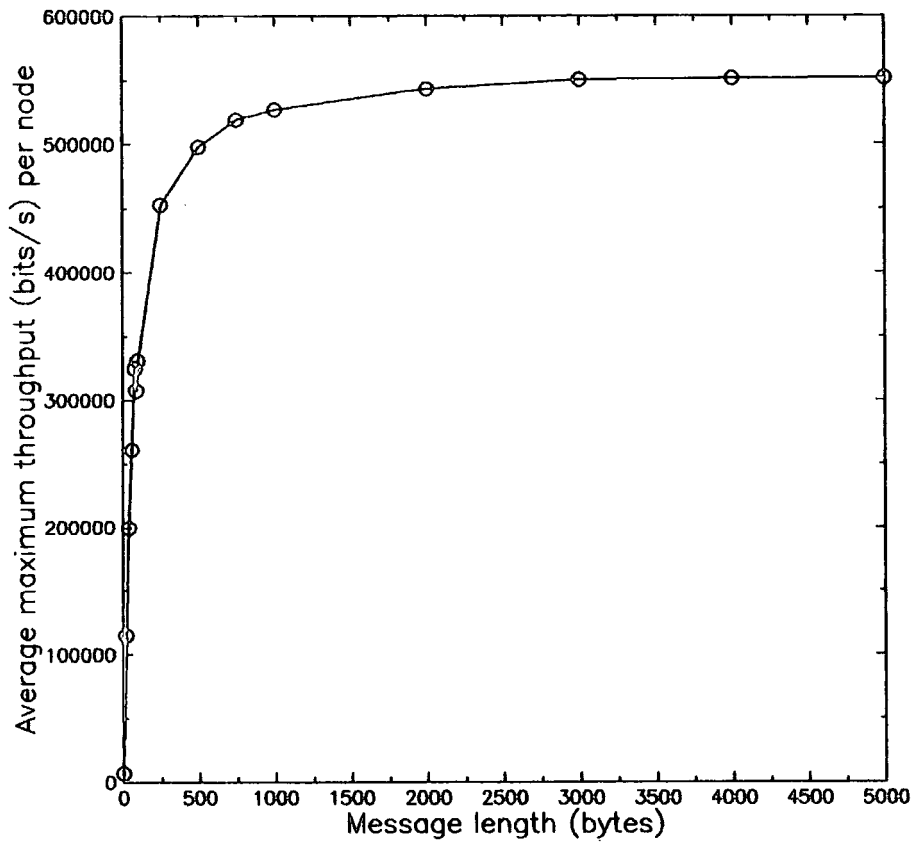
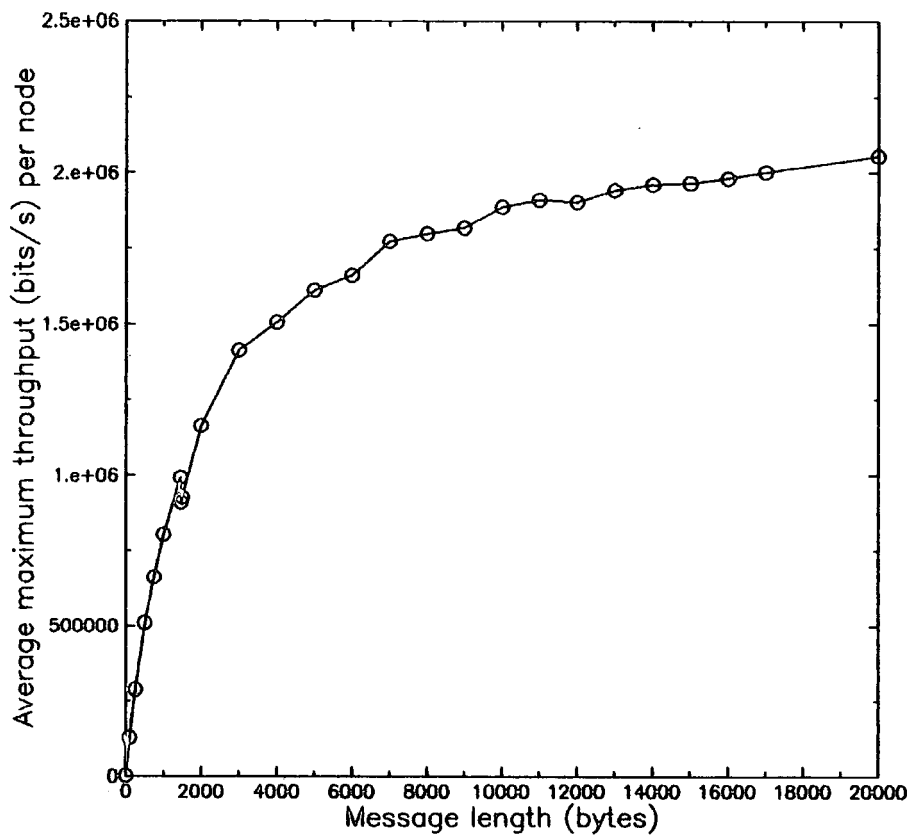Figure 4.24: Swiftlink, 1 active node

Figure 4.25: Ethernet, 1 active node

put does vary with the message length. Taking Figure 4.24 as an illustration, the AMT initially rises quite quickly, until it drops briefly when the message length reaches just over 86 bytes, i.e. the maximum packet length. This drop is due to the loss in efficiency when a message has to be split into more than one packet, e.g. a 87 byte message is split into one 86 byte packet followed by one 1 byte packet. (Messages under 87 bytes long are sent as only one packet.) The AMT then begins to rise again until the message length reaches about 1000 bytes, at which it begins to level off. The limiting factor here is probably the processing efficiency of the NETBIOS, and this represents a physical limit to the network's throughput. Hence the maximum throughput for the Swiftlink node is about 0.55 Mbps (megabits per second), at a message length of about 3000 bytes. As its physical data rate is 1.5 Mbps, the network is operating at approximately $\frac{1}{3}$ of its theoretical capacity.

For the Ethernet (Figure 4.25), we see a similar shaped graph. In this case, the curve is somewhat less smooth, and, after analysing the pattern of points, it seems reasonable to redraw it as Figure 4.26. Here the sudden drop in AMT occurs not only at a message length equal to the maximum packet length, but also at all multiples of this value. This seems to be likely behaviour; every time a message has to be split into n maximum length packets plus a few bytes one would expect a loss in efficiency. It may be that Swiftlink exhibits a similar feature, which would occur at multiples of
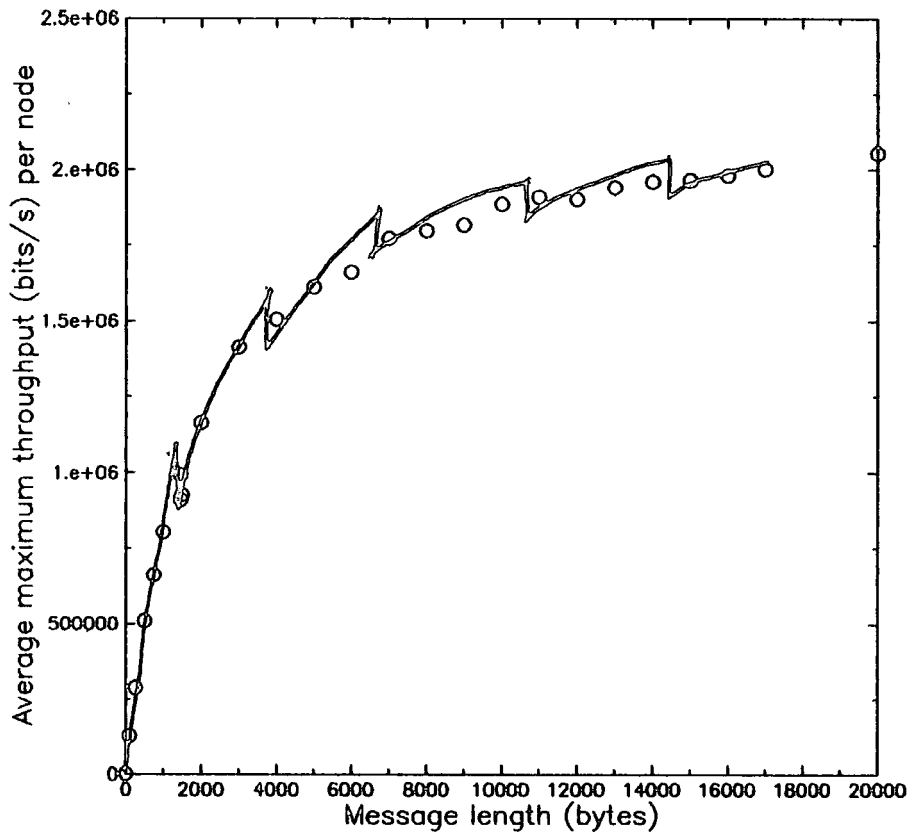
Figure 4.26: Ethernet, one active node

86 bytes, but due to the values of message length used in the experiments it is rather ambiguous; future work may verify this. In any case, the maximum AMT for Ethernet nodes is about 2 Mbps, approximately $\frac{1}{5}$ of its theoretical capacity.

Experiments T2, T3 and T4

In these experiments, all 10 nodes are active (transmitting), but in each case the destination of the messages is different. The purpose of this was to investigate the effects of certain traffic patterns on buffer insertion ring LANs. Theoretical analysis (Section 4.1) has shown that, under certain conditions, LANs using this protocol should be able to achieve a total network throughput which is up to two times greater than the physical data rate. This matter is discussed further in the results section. Traffic destination patterns are less significant for Ethernet due to its bus topology.

For these experiments, the results for Swiftlink will be dealt with first, as traffic destination patterns (should) affect buffer insertion LANs more than CSMA/CD LANs. The theoretical aspect of this work has been covered in Section 4.1, so only a brief description of the key concepts will be given here. Figure 4.27 illustrates how the buffer insertion protocol operates.
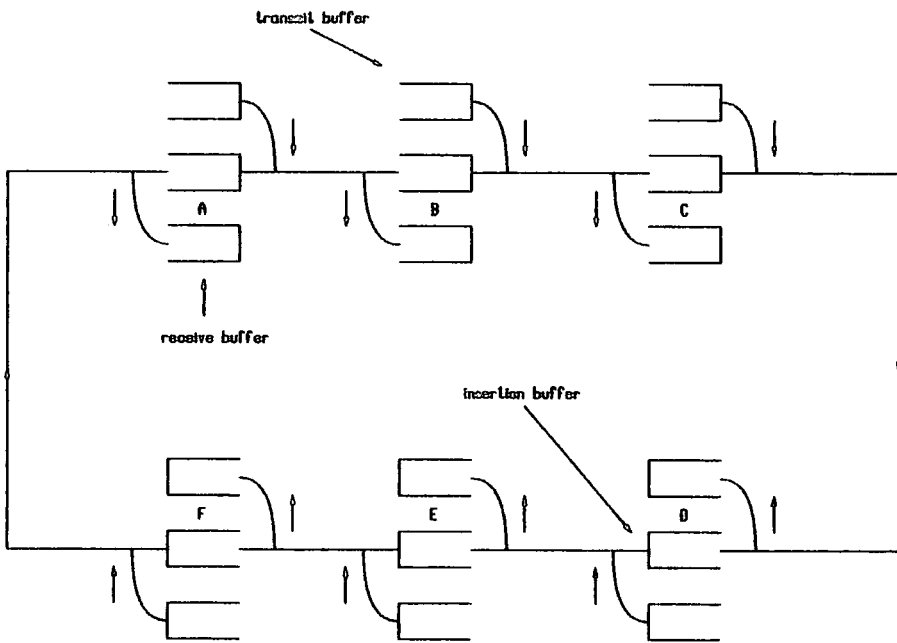
Figure 4.27: Buffer insertion ring protocol operation

Let us assume that all the nodes wish to send data. Each node holds its outgoing data packet in its transmit buffer, and waits until there is a gap in the incoming traffic on the ring. When this gap arrives, the node inserts its insertion buffer into the ring, to 'catch' the incoming data, and starts emptying its transmit buffer onto the ring. When the transmit buffer is empty, the insertion buffer may empty its contents onto the ring, until it is eventually empty also. The node which is to receive the data simply copies the packet off the ring, giving its own transmit buffer the opportunity to send a packet if it so wishes. Because of the fact that the receiving node removes packets from the ring, and that a node may transmit whenever there is no incoming traffic, more than one transmission may take place at the same time. In effect, the total network throughput can be greater than the available bandwidth.

The throughput is, however, limited by the pattern of the message destinations. The maximum throughput should occur when each node sends to its immediate successor in the ring. In this case, a node always receives packets addressed to itself, and can therefore empty its transmit buffer whenever it desires. (This is not strictly true: acknowledgement packets addressed to other nodes may arrive, but as these are very short they may be neglected.) Thus in a n node network, n transmissions can occur simultaneously, giving maximum throughput of n times the data rate.

An atypical scenario giving a lower bound on throughput is the case when each node sends to its predecessor in the ring. A node waiting to send a message must wait for a considerable amount of time before a gap in the incoming traffic occurs, because traffic from $n - 2$ nodes is passing through it (as it is not addressed to this node). The worst case of all is when a node sends messages to itself, all the way around the ring. The waiting time is even longer, because traffic from $n - 1$ nodes must pass through it.

In Figures 4.28, 4.29 and 4.30 we can see how theory compares with the real operation of the network. All these graphs have a similar shape to Figure 4.24, but the maximum throughput is different in each case. In the 'sending to successor' case (Figure 4.28) we observe a maximum AMT of just under 0.3 Mbps for Swiftlink. This is the result for one node; the total network throughput, AMT_{tot}, is obtained by multiplying this by the number of active nodes, i.e. $AMT_{tot} = AMT \times 10$. Hence AMT_{tot} equals just under 3 Mbps. The theoretical value of AMT_{tot}, for this traffic pattern, is the data rate multiplied by the number of nodes, i.e. 10×1.5 Mbps $= 15$ Mbps. The measured value is much lower than the theoretical one. The difference in their values is no doubt due to the fact that the nodes cannot send data at 1.5 Mbps; the speed will be limited by the processing time of the NETBIOS.

Figure 4.29 reveals that, when Swiftlink nodes send to their predecessors, the highest AMT per node is about 0.124 Mbps, which is, as expected, much
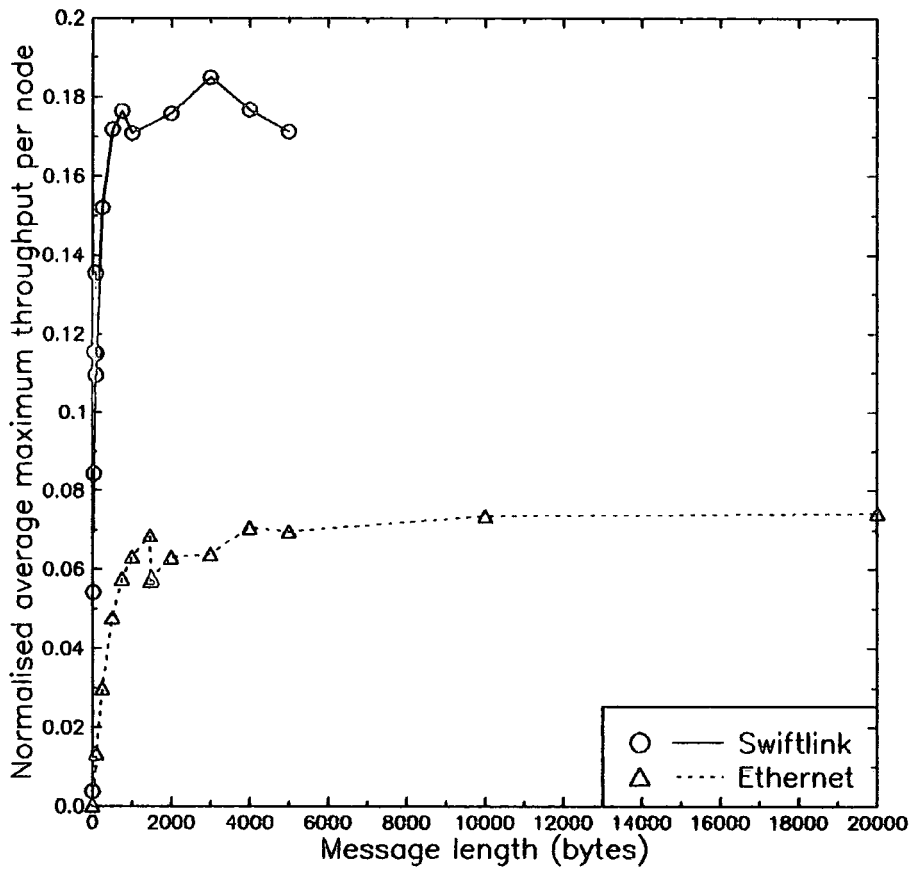
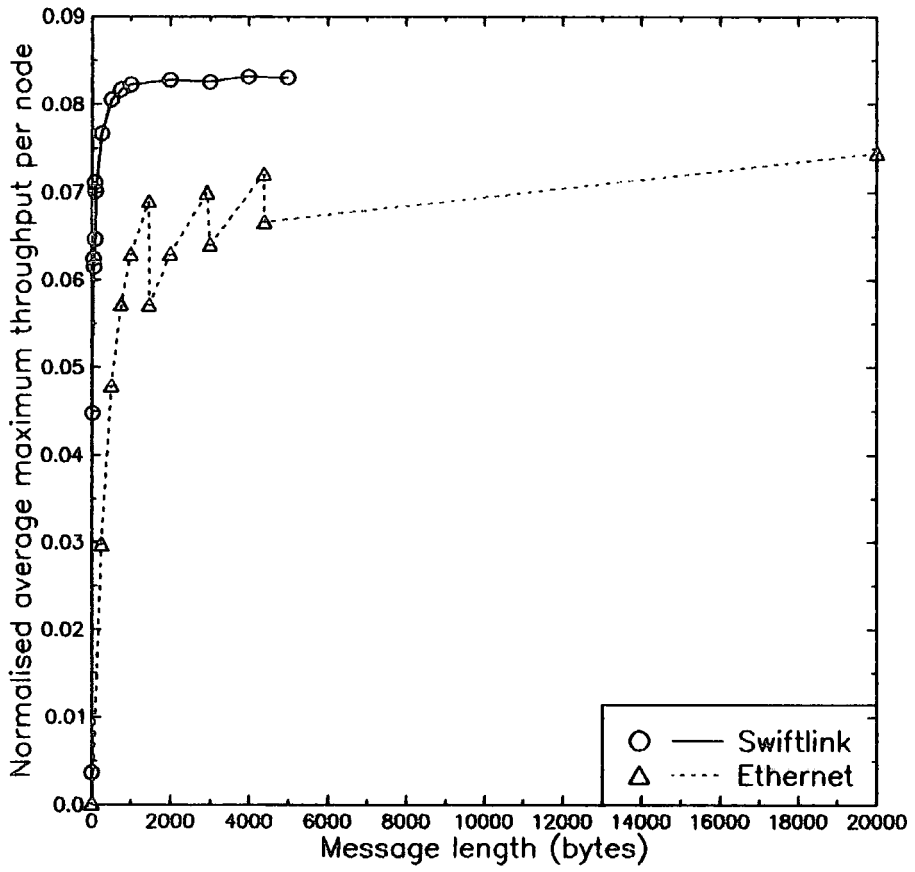Figure 4.28: Swiftlink and Ethernet, sending to successor

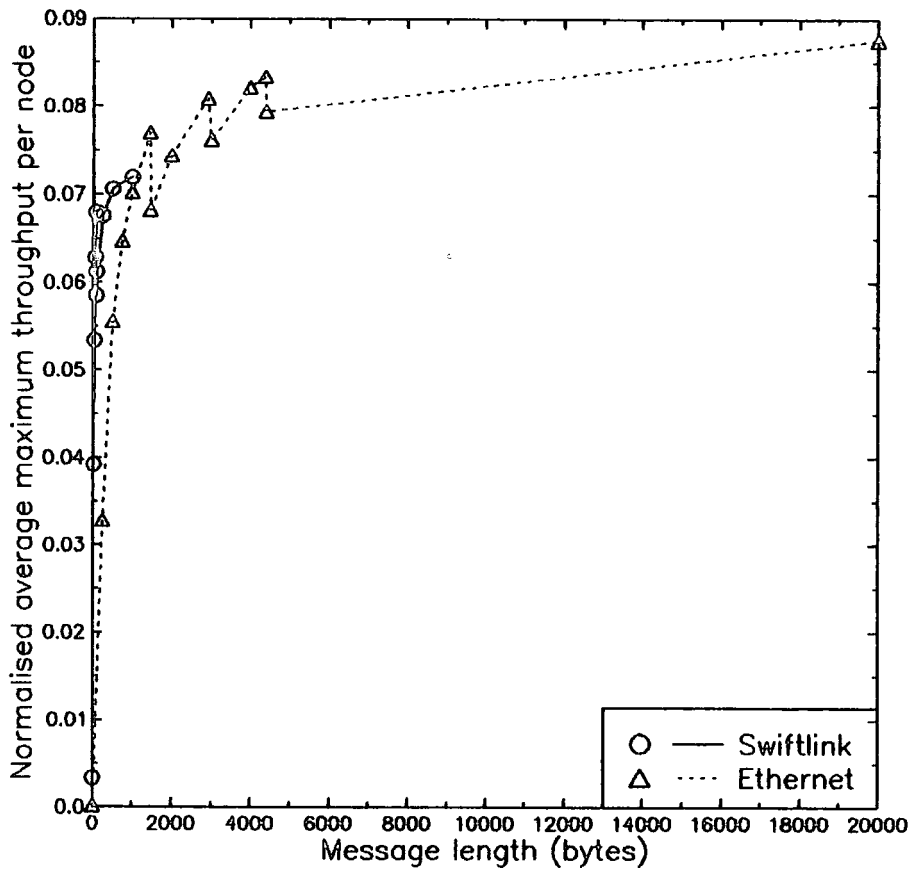Figure 4.29: Swiftlink and Ethernet, sending to predecessor

Figure 4.30: Swiftlink and Ethernet, sending to self

lower than the data rate (1.5 Mbps). In the situation where Swiftlink nodes send to themselves (Figure 4.30), the highest AMT is still lower, at around 0.108 Mbps. The experimental results, then, appear to be consistent with theoretical prediction, at least in a relative sense rather than an absolute one.

The results for Ethernet show that the traffic destination pattern has little effect on the AMT; the curves for 'successor' and 'predecessor' are almost identical. Because Ethernet is a bus network with a 'free' access strategy, there is no concept of ordered access to the network, and hence terms such as 'successor' and 'predecessor' do not refer to the physical order of the nodes. Figure 4.30, while having a similar shape to the other two graphs, displays a slightly greater maximum value of AMT. This is probably due to some function of the NETBIOS rather than the Ethernet protocol itself.

Experiment T5

Here the effect of the number of active nodes in the network is investigated. Each active node sends fixed length messages (1000 bytes for Swiftlink, 3000 bytes for Ethernet) to its successor, while the number of active nodes is varied.
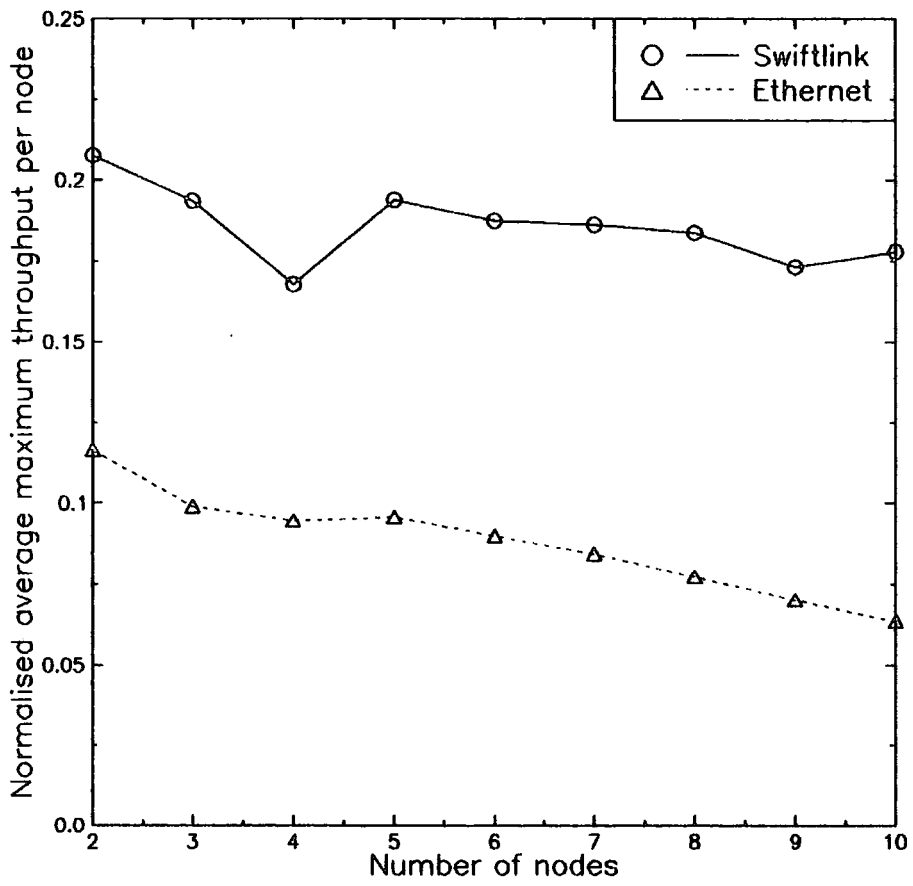
Figure 4.31: Swiftlink and Ethernet, varying number of nodes

Considering Swiftlink first (Figure 4.31), the AMT does not vary greatly with the number of active nodes. However, this is probably because each node was sending to its successor; in this case the messages do not have to pass through any intervening nodes, and so one would expect the throughput to be the same regardless of the number of nodes. It would have been better to conduct this experiment with the nodes sending to their predecessors, i.e. the messages travel a longer distance; the results should then be more interesting.

The Ethernet results, however, show clearly that the AMT falls with increasing number of active nodes. This is as expected; each time a node is added there is more chance of a collision, and hence a longer waiting time and lower throughput.

Summary

⊖ With NETBIOS driven networks the theoretical maximum throughput is impossible to attain; the NETBIOS software itself forms a bottleneck.

⊖ Traffic destination patterns can greatly affect the performance of buffer insertion ring LANs, certain patterns allowing greater use of the network bandwidth than others. CSMA/CD LANs are less affected by this factor, but allow a maximum use of bandwidth of only 100%.

88

○ The number of active nodes in the network tends to affect the performance of Ethernet more than that of Swiftlink.

○ In absolute terms, Ethernet allows a greater network throughput, due to its higher data rate. In relative terms, however, Swiftlink performs better, at least for the implementation of Ethernet used in the experiments.

Chapter 5

Simulation of the buffer insertion ring

5.1 Modelling a proprietary buffer insertion ring LAN

Simulation of the buffer insertion ring protocol was performed using a purpose written program on a Sun 3 workstation. Initially a generic protocol model was constructed and coded in Pascal. After working extensively with Superlink in the laboratory, it was realised that the Pascal simulation program did not incorporate certain features of the network, such as message acknowledgement, delay time in the nodes, and the ability to model gateways. Detailed information was obtained about the features while performing ex-

periments on the Superlink network. This information provided the basis for a new requirements analysis from which improved models were developed. During the modelling process certain decisions had to be taken about which aspects of the real system were to be modelled and how they might be implemented. Message acknowledgement and the recognition of processing times in the nodes were considered essential for the simulator to model Superlink rather than a general buffer insertion ring protocol, so it was decided to include these.

It was thought that the addition of gateway simulation between two rings would be a very time consuming task; hence the gateway function was limited to gateways back onto the same ring, as this was required to simulate some of the experiments performed in Cambridge. The latter consideration also inspired the redesign of the program module which models the hosts connected to the network nodes. Previously this had required all hosts to generate messages of the same average length at the same average rate. As this was rarely the case in 'real-life' experiments, the new version of the module allowed each host to have its own configuration as regards message length, generation rate and so on.

The results of the experiments allowed the testing of the original models. They also provided valuable data for the production of more refined models.

In some of the examples given below, rather than use theoretical algorithms, the results of the experiments were used to derive a model based on practical experience. This is a somewhat different approach to that used in previous simulation software. Originally an algorithm was designed based on the data link and physical layers (layers 1 and 2 of the OSI model; references to layer 2 in fact refer specifically to the MAC layer) of the network protocol, and this algorithm was implemented to form a simulation tool. As layers 1 and 2 are often implemented in hardware and are relatively simple in their operation, it is not difficult to produce an accurate model to simulate them. However, as a real network does not merely consist of two layers of the OSI model, it is insufficient to restrict modelling to these lower two layers; layers 3 to 7 have a very important function and, indeed, often affect performance to a much greater degree, as will be seen later. The question arises, then, as to how these higher layers should be simulated. As they are usually implemented if software, which is considerably more complex in operation than layers 1 and 2, it is impractical to attempt to implement an algorithm based on their precise functionality. Hence the method adopted is based on actual performance measurements made on a real network, rather than pure simulation of a protocol algorithm. Measurements of quantities such as node delays and connection times have been used to derive functions which will represent the quantities in the simulation. For example, the graph in Figure 4.12 shows how node processing delay varies with message length

when data is sent onto the network. A curve derived from this graph has been used to implement a function, whose argument is message length, which returns a value of node delay to be used during simulation.

In essence, then, the final version of the simulator uses a hybrid approach: layers 1 and 2 are modelled by implementation of the protocol algorithm, and the higher layers are modelled by functions based on measured data. Figure 5.1 shows the modified design strategy. This approach affords a great reduction in code complexity and simulation time.

As the structure of Superlink is not clearly defined as regards the OSI model, the layers above the MAC layer have been associated with the processing which occurs between the host connection to the RS232 box and the ring connection, excluding the functioning of the various transmit, receive and pending buffers. The aspects of the simulation which involve these adaptive modifications are enumerated below.

1. Connection/Disconnection Time: Rather than add a large amount of code to model connection and disconnection procedures, data which relates network delay to number of intervening nodes (from Figures 4.7 and 4.10) has been used to implement a function which will give an approximate value for the time taken to execute these requests, based
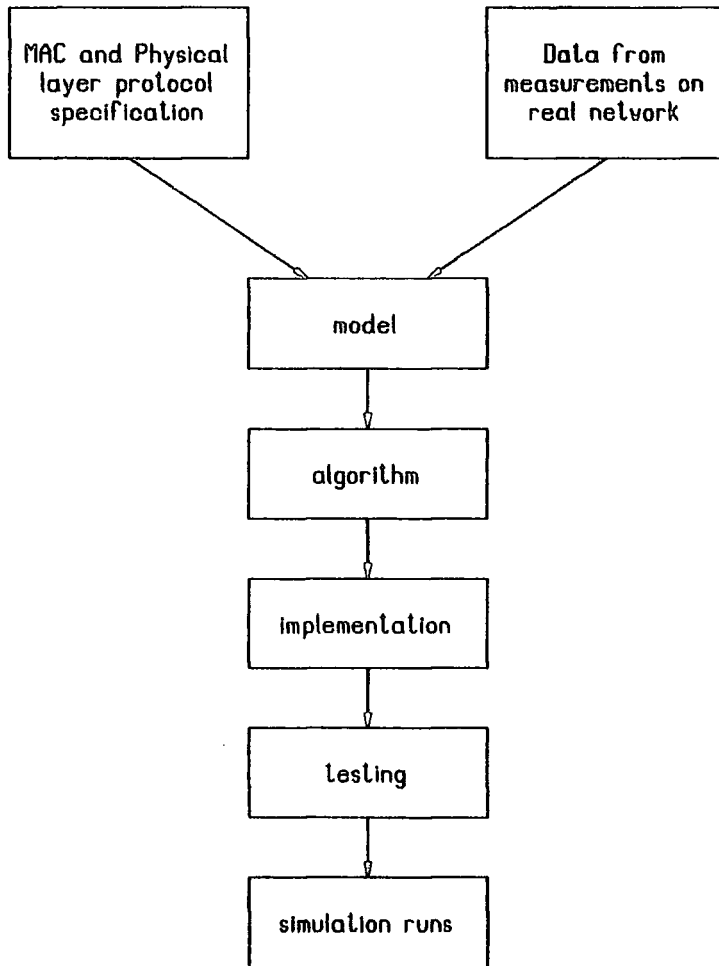
93

Figure 5.1: Modified simulator design strategy

on the number of nodes between sender and receiver of the request. Additionally, an option is available whereby these delays may or may not be included when making performance measurements. This option was included because connection requests may be used in different ways in a real situation. One possibility is to negotiate a connection at the start of the communication, then leave the connection open so that several messages may be sent; in the simulation the connection is left open indefinitely and the connection time is neglected. The other possibility is to open and close a connection for each message sent; in this case the connection/disconnection times are included in the simulation.

2. Node Delay for data passing between host and ring: Similarly, data which relates node delay to message length (from Figures 4.11 and 4.12) has been used to derive a function to represent the node delay for data. These are always included in performance measurements, as they are an inherent part of the Superlink system. A polynomial curve fit was used to give a time based on the message length.

3. Gateways back onto ring and Message Acknowledgement: The gateway model was developed as a special case of the host model. Each gateway can be configured to operate at a certain speed and with an

additional fixed delay included. Gateways contain 2 input buffers, so that after two packets have been received any further packets are rejected until a buffer becomes free again. This, of course, requires some form of message acknowledgement. A system was added so that when a message is received, the receiver may send back an acknowledgement (ACK) or not-acknowledgement (NAK) packet to the sender. The sender will continue to repeat the message until it is acknowledged. Thus a gateway will only send back an ACK if it has a free buffer.

Simulation Results

After making the above mentioned modifications to the software, several simulation runs were performed. The first runs were intended to give a general impression of the performance of the Superlink network; the second batch of simulations corresponded to some of the experiments performed in Cambridge, which were described in Section 4.2.1.

In the first simulations, one network parameter was varied while the others were held constant. These constant values are shown in Figure 5.2.

Experiment S1

Network Parameters
Data rate = 1.5 Mb/s
Number of nodes = 10
Message length = 80 bytes (fixed)
Data packet header length = 48 bits (fixed)
Message generation rate = 5 messages / second (Poisson distributed)
(a low load)
All nodes send traffic to random destinations.
Simulations run until 100 messages have been sent.
Node delays always included in simulations (except where stated otherwise).
Connection/disconnection times included only where stated.
Cable propagation delays neglected.

Figure 5.2: Network parameters

Figures 5.3 to 5.5 show how the throughput varies with the data rate of the network. These simulations are intended to show how the buffer insertion protocol would perform at higher data rates. Figure 5.3 shows this relationship when the node delay for data is included in the simulation and the time for connections and disconnections is excluded; i.e. a connection is made between each node and all others and left open throughout the simulation. This should be compared with Figure 5.5, which shows the same relationship but with the node delay excluded from the measurements. Figure 5.5 exhibits the expected straight line; the throughput rises in proportion to the data rate. The former is always slightly lower than the latter because of the packet headers and acknowledgements which use up bandwidth which would otherwise be available for user data. Figure 5.3, however, has a considerably different shape and magnitude. The throughput is so much lower because
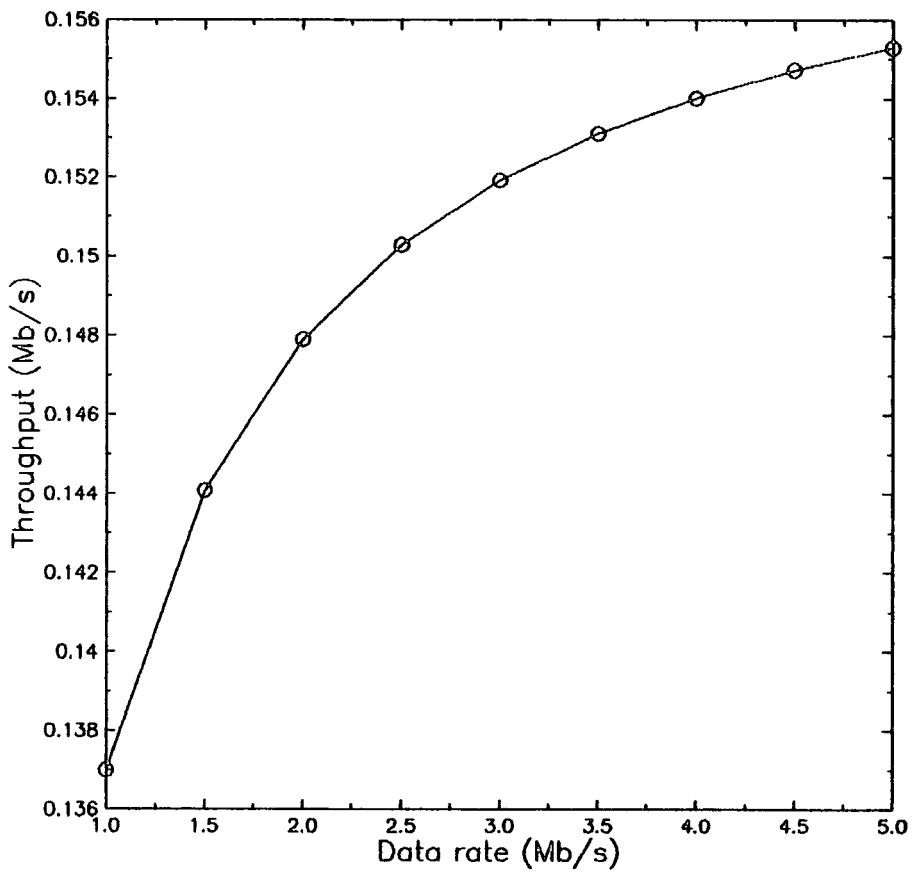
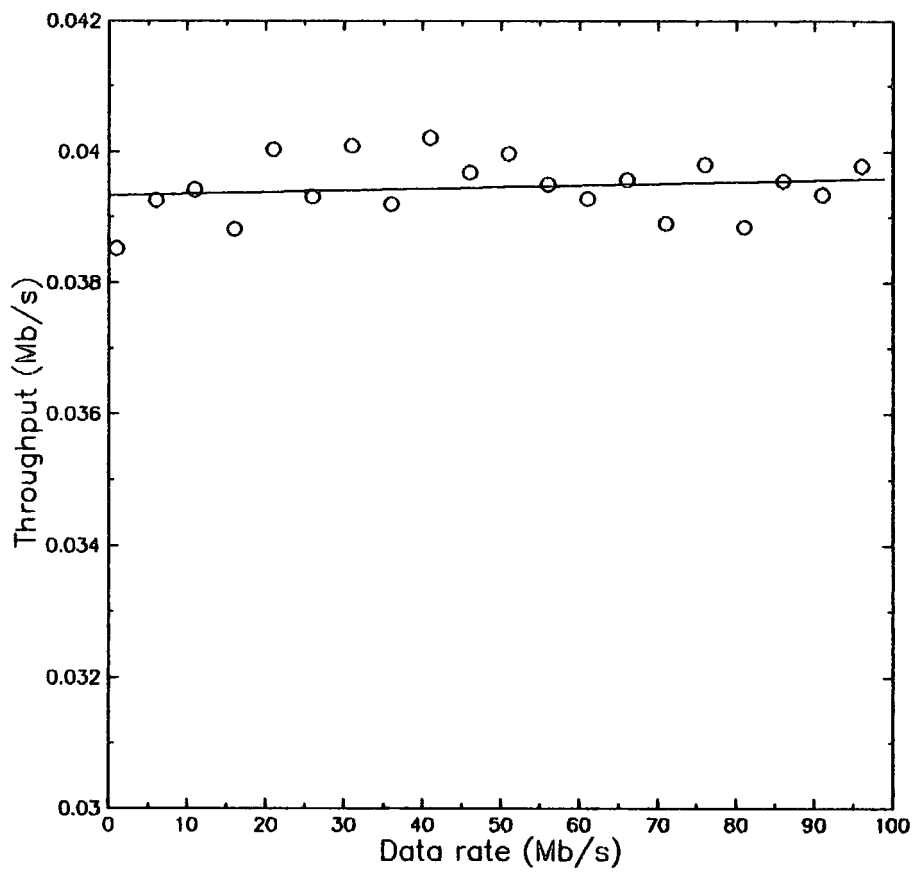Figure 5.3: Throughput / data rate (including node delay)

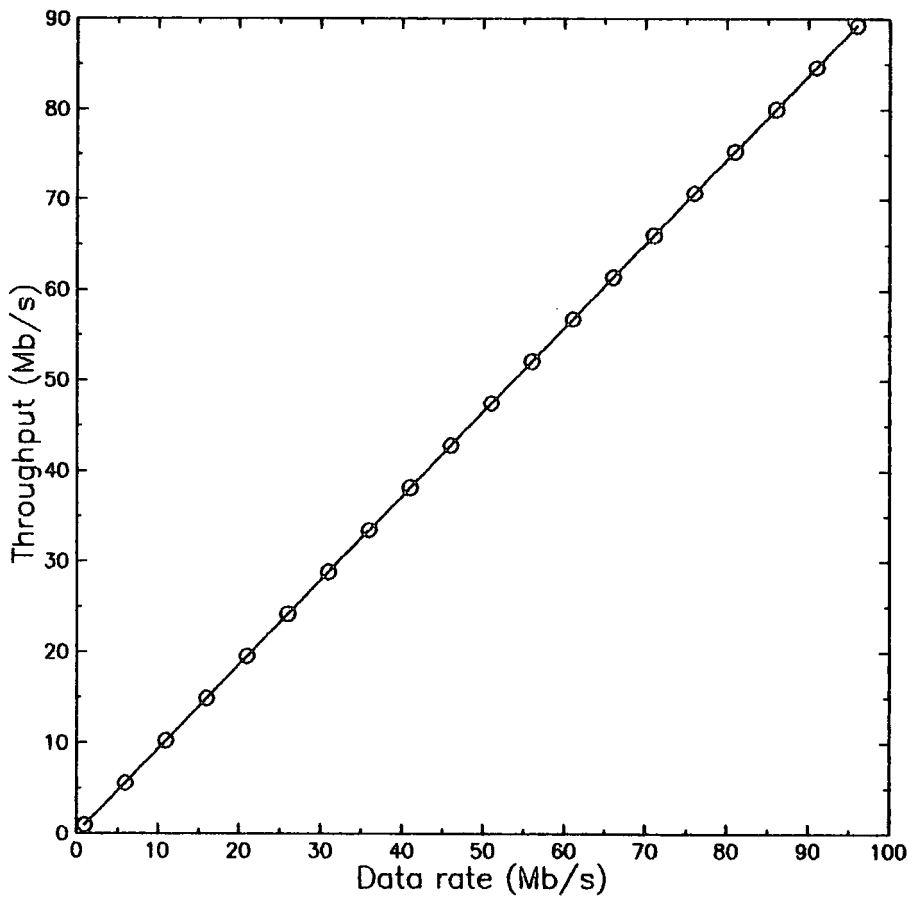Figure 5.4: Throughput / data rate (including node delay and connection time)

Figure 5.5: Throughput / data rate (no delays included)

of the processing delay in the nodes, which is large in comparison with the transmission time. This node delay limits the throughput; it can no longer increase linearly with the data rate. In Figure 5.4 a connection has been established (and later closed) for each message sent; thus in addition to the data transmission time, an extra delay was incurred due to negotiation of the connection. In this case the throughput oscillates about a fixed value of approximately 0.0393 Mb/s. The connection/disconnection time is so large in comparison with the transmission time that an increase in data rate causes very little change in throughput.

Experiment S2

Figure 5.6 shows that the message delivery time remains reasonably constant regardless of the number of nodes in the network (for the range of nodes used - limited to 60 by memory and processing time constraints). Again, one suspects that the node delay has swamped the actual transmission time.

Experiment S3

The results shown in Figures 5.7 and 5.8 were produced by simulation of the network configuration from Experiment C4 in Section 4.2.1. These results are to be compared to Figures 4.14 and 4.15 in Section 4.2.1, i.e. the simulation is to be compared with the real measurements. The configuration for Experiment S4a is shown in Figure 4.13; for Experiment S4b, the test
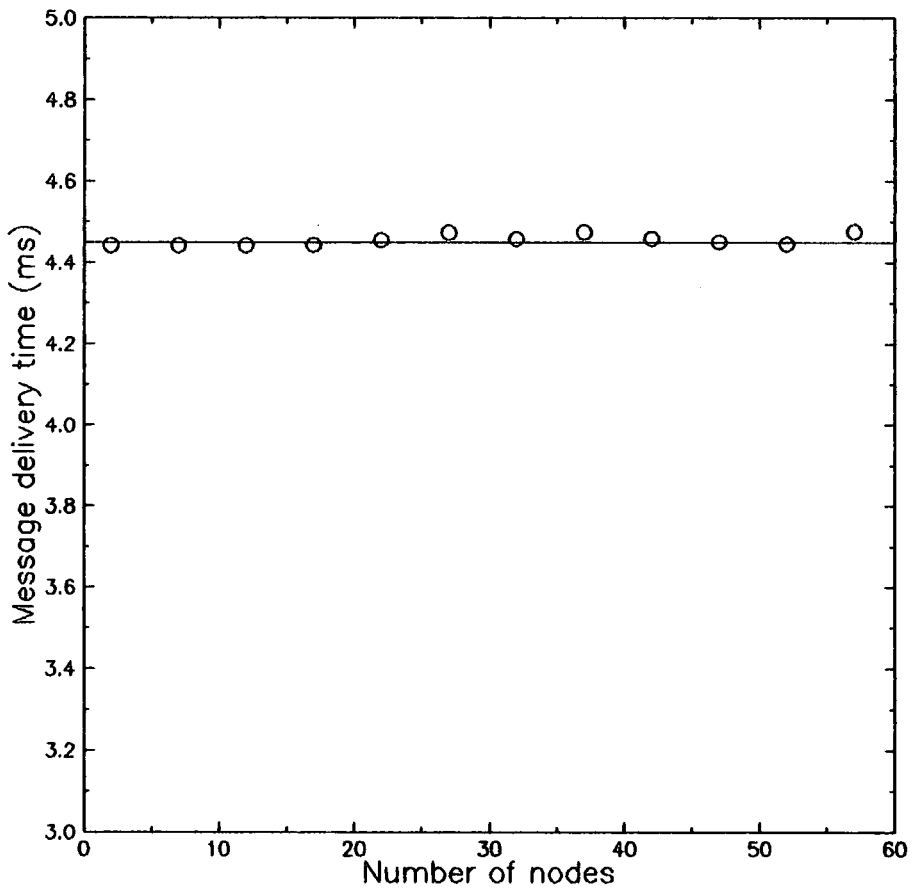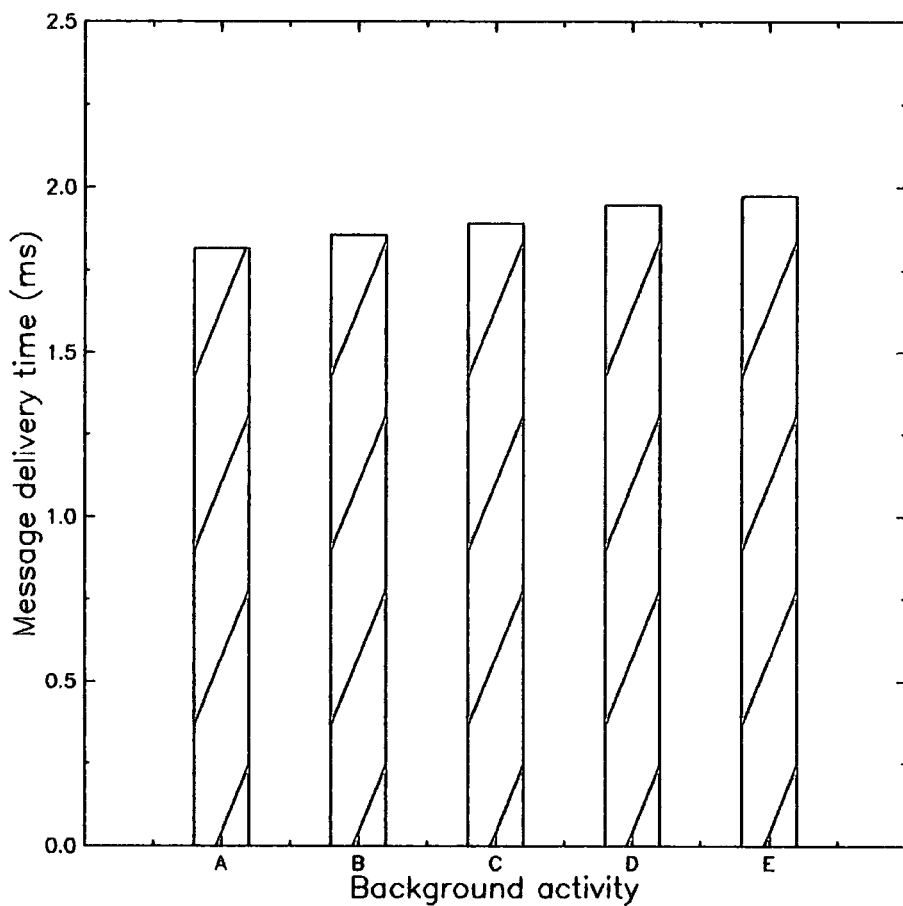
Figure 5.6: Message delivery time / number of nodes

Figure 5.7: Message delivery time / background activity (18 nodes between sender and receiver)
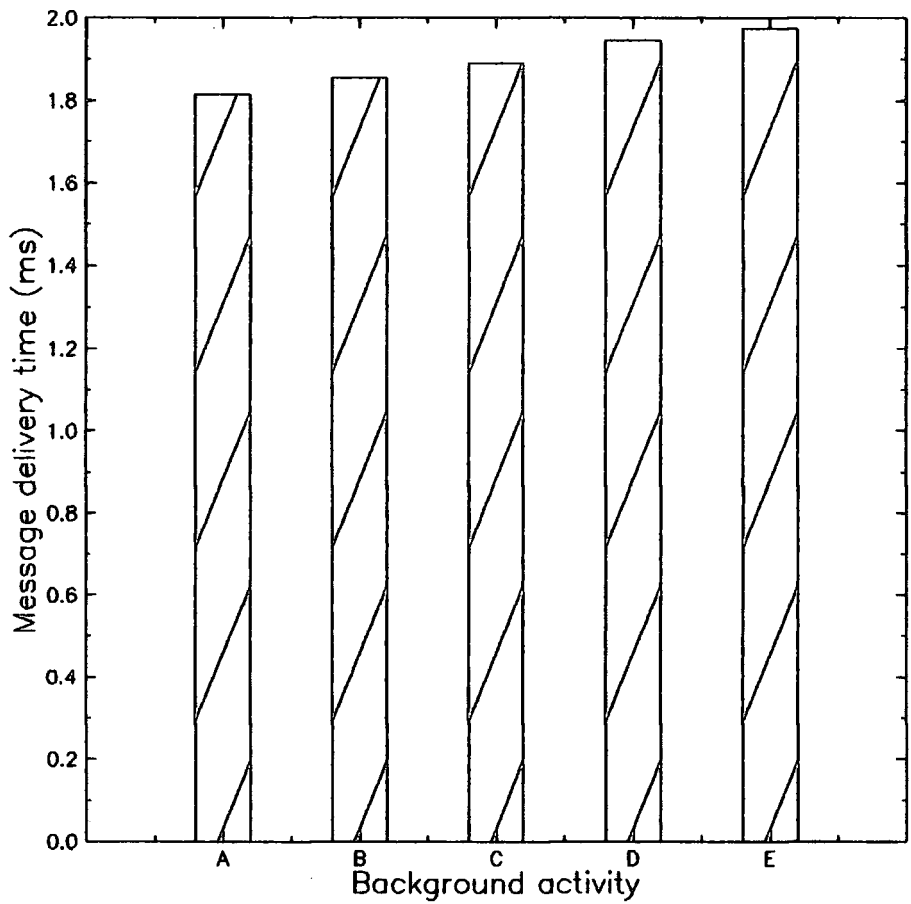
Figure 5.8: Message delivery time / background activity (0 nodes between sender and receiver)

message source and destination were swapped in their positions on the network. The message lengths and generation rates were as used in the original experiment. However, the background traffic generation rate was derived from Experiment C5 in Section 4.2.1 - the measured throughput was used as the generation rate.

Examination of the graphs reveals that, though the bars have a slightly different magnitude than those derived from the measurements, their relative heights are similar. A likely explanation for the simulation results being smaller than the measured results is that the background message throughput is lower than it should be. A subsequent simulation run revealed that the actual background throughput did not equal the generation rate of the background source; in fact it was considerably less. Consequently the message delivery times are slightly lower due to lower background activity.

Summary

- Processing delays in the nodes greatly limit the performance of the network. An increase in data rate after a certain value will not bring about an increase in throughput.

- Incorporating results from measurements into the simulation can pro-

vide a much greater degree of realism.

5.2 Generic single and dual buffer insertion rings compared

5.2.1 Simple routing strategy

The single ring buffer insertion simulator described in the previous section was developed into a version which modelled a dual buffer insertion ring. In this model data flows on both rings simultaneously, in opposite directions. The physical and data link protocols used are the same as in the single ring simulator but the network layer protocol is necessarily different. With two independent rings a choice of routing strategies is available. The main purpose of this simulation, then, is to investigate the use of two contrarotating rings, and also to compare the performance of different routing strategies.

The theoretical aspects of the dual buffer insertion ring were considered in Section 4.1.2. The simulation work here is intended to support this analytical approach. Simulations have been performed of both single and dual ring

networks, under the same operating environment for ease of comparison. Messages are routed through the dual ring using the following algorithm. The first packet of each message is sent on both rings, and the destination node, having received a packet on a certain ring, sends out the acknowledgement on the same ring. When the source node has received an acknowledgement on both rings, it chooses the ring on which the first acknowledgement arrived and uses this ring to send all the remaining packets of the message; effectively the quickest route is chosen.

The simulation parameters which remained constant were set up as shown in Figure 5.9. The variable simulation parameter was always the average total network throughput, and the measured quantity was the average message delivery time.

Network Parameters
Data rate = 1.5 Mb/s
Number of nodes = 10
Data packet header length = 48 bits
Acknowledged connectionless service
All nodes send traffic to random destinations
Node delay = 0
Bit delay through node = 0
Cable propagation delays neglected

Figure 5.9: Constant simulation parameters

Figure 5.10 shows the experiments performed.

Experiment Number	Ring Type	Number of Active Nodes	Packets per Message
B1	single ring	1	1
B2	single ring	1	10
B3	single ring	10	1
B4	single ring	10	10
B5	dual ring	1	1
B6	dual ring	1	10
B7	dual ring	10	1
B8	dual ring	10	10

Figure 5.10: Experiments performed

Simulation Results

As the theoretical aspect of this work has been considered in Section 4.1, only

a brief resumé will be given here. From the theory it has been found that for

a single buffer insertion ring, using destination removal of packets, the maxi-

mum normalised throughput (MNT) achievable is 2 (normalised throughput

= actual throughput / data rate of ring). This assumes a symmetric traffic

pattern, i.e. the destination node for a message is chosen at random from all

nodes on the network. Effectively, the network can handle total loads of up

to double the data rate (total load = sum of loads generated by each node),

given this traffic pattern. This is a significant improvement over other types

of LAN (e.g. token ring), which generally have a MNT of 1.

Under the same operating conditions, the dual buffer insertion ring's

MNT was not so easy to predict. In Section 4.1 it was estimated at 8 for the

'quickest path' method of message routing. If this prediction is accurate, the

dual ring should handle total loads of up to 8 times the data rate without saturation occurring. These characteristics are illustrated in Figure 5.11. In the examination of these results one may observe how those from the simulation compare with those predicted by analysis.
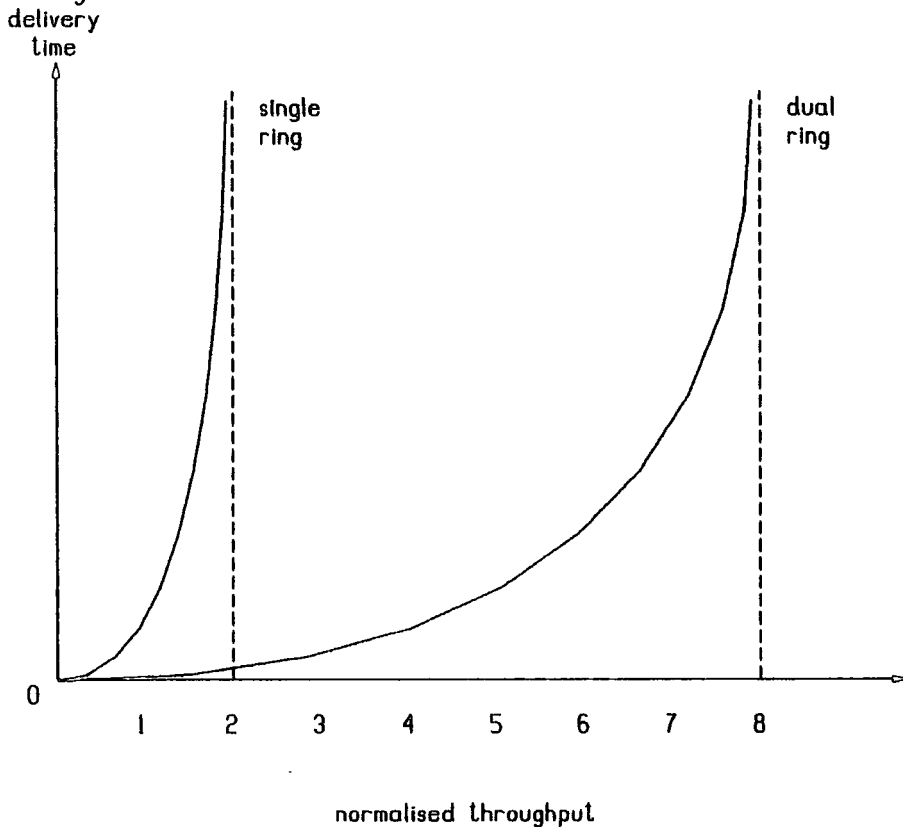

Figure 5.11: Theoretical results for single and dual rings

The first 2 graphs show results for the single buffer insertion ring. From Figure 5.12 it can be seen that the MNT with only one node active is only 1, not 2 as predicted (recall that the data rate is 1.5 Mb/s). The MNT value of 2 derives from the fact that, with destination removal of packets, at any one time different segments of the ring can be transmitting different messages.
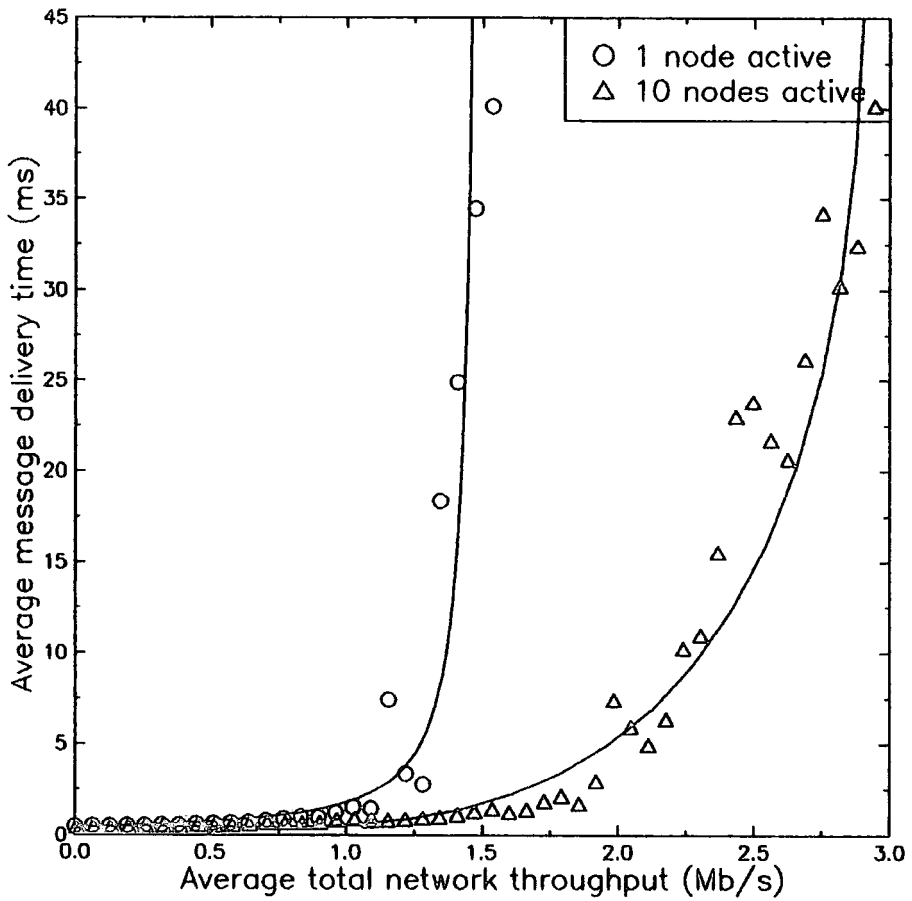
109

Figure 5.12: Single ring, 1 packet per message

Figure 5.13: Single ring, 10 packets per message

Since only one node is transmitting here, this benefit cannot be achieved, hence the low MNT. When 10 nodes are transmitting one sees the expected result: a MNT of 2. Figure 5.13 exhibits a similar phenomenon; though the message length is now 10 packets, with only 1 node active the network saturates with a MNT of (almost) 1. Once again, when 10 nodes are active, we see the expected MNT of 2. These results, therefore, confirm the validity of the theoretical analysis.

Figure 5.14: Dual ring, 1 packet per message

Figure 5.15: Dual ring, 10 packets per message

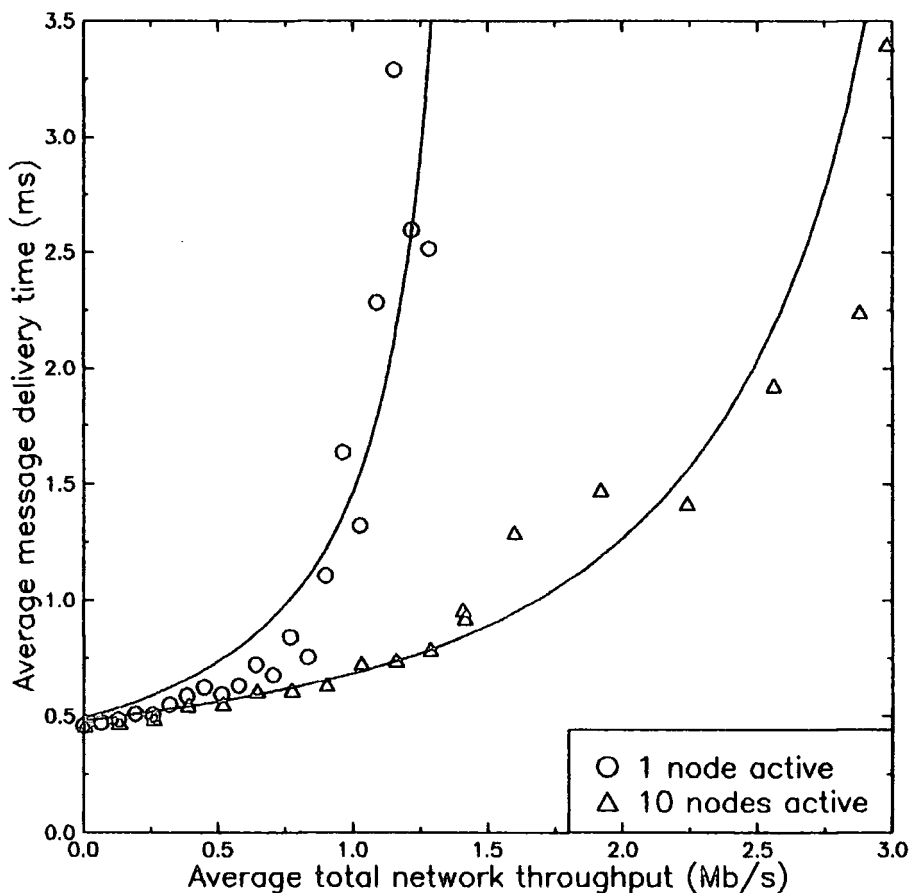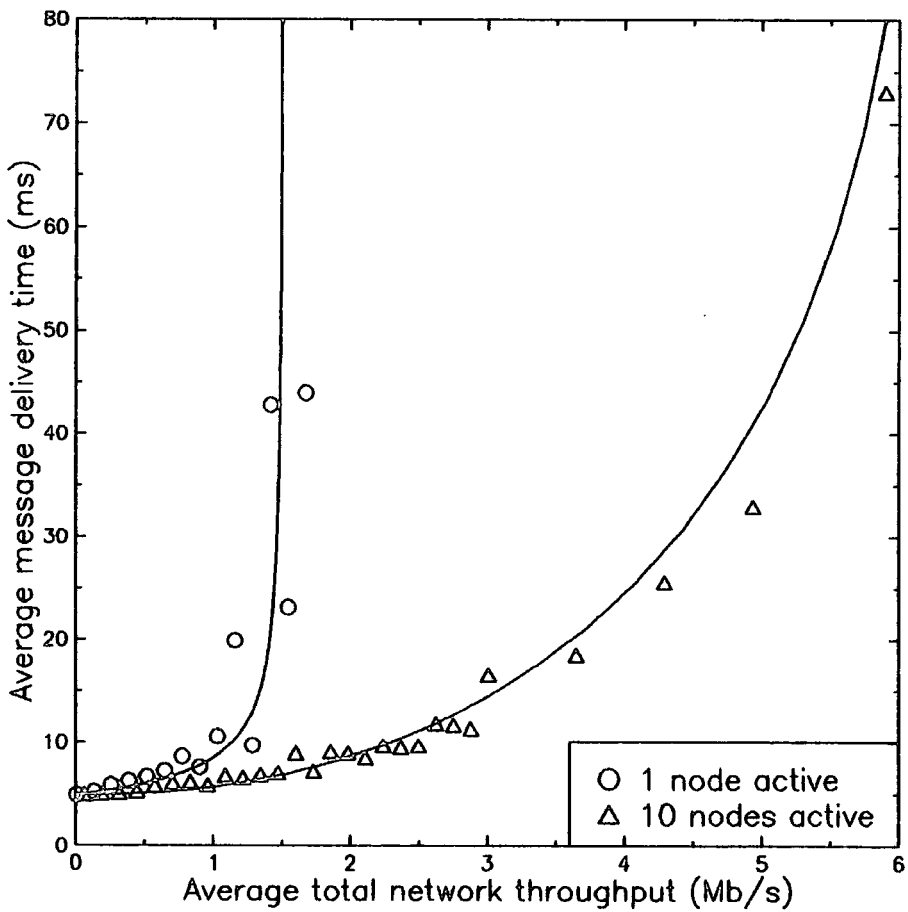In the next 2 graphs the dual ring is considered. The line in Figure 5.14 where only one node is active shows the same characteristic as the '1 node active' line in Figure 5.12. The MNT is 1, because of the reason stated above, and also because when a message is only 1 packet long the 'quickest path' technique will not bring about an improvement in performance (the message has to go around both rings). Considering the other line in Figure 5.14, though all 10 nodes are active, the messages are all only 1 packet long, so the MNT is 2. This is the same as for the single ring; the benefit due to the routing strategy is lost because the messages are short and are transmitted on both rings, wasting bandwidth.

In Figure 5.15, where each message is 10 packets long, we observe a similar result when only one node is active: use of the bandwidth cannot exceed 100%. The other line in this figure, however, better illustrates the advantages of the dual ring. In this case, because all the nodes are active and the messages are longer than 1 packet, the benefit of the 'quickest path' technique is apparent. The MNT is 4 here, better than for the single ring but rather lower than the expected value (estimated at 8 in Section 4.1). This is due to the fact that the simulation uses an acknowledged connectionless service, whereas the mathematical analysis required neglecting the effects of acknowledgement packets altogether. In the simulation, when a node transmits a message it must wait for that message to be acknowledged by

114

the recipient before transmitting again, due to the window size of 1. Under high offered loads, this will limit the maximum throughput through one node, and thus be responsible for the lower than expected MNT.

Summary

o The single buffer insertion ring can provide an MNT of 2, provided that all nodes transmit with a symmetric traffic pattern. Variations in the traffic pattern can give rise to throughputs between the extremes of 1 and n, where n is the number of nodes.

o Under the routing strategy tested, the dual buffer insertion ring can provide an MNT of 2 for single packet messages and 4 for multi-packet messages, again assuming the traffic conditions described above and that an acknowledged connectionless service is employed. A higher value than 4 is likely to be possible with an unacknowledged service.

5.2.2 Alternative routing strategies

The above simulations of the dual buffer insertion ring employed a rather basic strategy to route packets around the network. In this section differ-

ent methods are considered in an attempt to improve overall performance. In a dual contrarotating ring LAN the designer of the network layer protocol, which involves the routing of packets through the system, must make careful decisions in order that use of the available bandwidth is maximised and the average message transit time is minimised. Two components of the network layer protocol are discussed here, and their effects on the system's performance are investigated by simulation.

Routing of acknowledgement packets

In any communications network it is important to have some means by which a node, having sent a message to another node, can be informed as to whether or not the message arrived at its destination. This usually involves the destination node sending a special message back to the source node to indicate whether the incoming message was received correctly. The special message is known as an acknowledgement signal (or ACK) if it indicates correct reception, or a non-acknowledgement (or NAK) if it signifies that some error occurred in the reception of the message. In LANs which employ source removal of packets, such as 802.5 token ring, the acknowledgement signal often forms part of the data message itself; acknowledgement is indicated by setting a status bit in the message before it returns to its source. LANs which use destination removal of packets, (e.g. Nine Tiles Superlink) however, cannot do this; the acknowledgement signal must be sent in an 'ACK packet',

116

as a message in its own right, after the data message has been received. For a single ring this is a reasonably trivial task, but for a dual ring the question arises as to the routing of the ACK packet through the network.

In the case of a dual contrarotating buffer insertion ring, such as Nine Tiles CoRoNet, the ACK may travel on the same ring as the data message, in the same direction, or it may travel on the other ring, in the opposite direction. In the experiments in the previous section, the ACK packet travelled on the same ring as the data message. It has been suggested that, if the ACK packets were to travel on the opposite ring, they would tend to arrive more quickly. This would allow the sending node to transmit its next data packet sooner, and thus improve the overall throughput of the network.

In order to test this hypothesis, simulations of a dual contrarotating buffer insertion ring, using the same software as in the previous section, were performed as follows. A 10 node network was used, with a data rate of 1.5 Mb/s. The average total network throughput was varied and the average message delivery time (neglecting connection times and software delays for clarity) was measured. Four scenarios were investigated, as shown in Figure 5.16.

The graphs (Figures 5.17 and 5.18) exhibit the characteristic curve shapes, whose interpretation was described previously. For the single packet message

Experiment	Message length (packets of 640 bits)	ACK packet path
R1	1	same ring
R2	10	same ring
R3	1	opposite ring
R4	10	opposite ring

Figure 5.16: ACK packet routing experiments

case (Figure 5.17) the maximum throughput is 3 Mb/s, i.e. double the data

rate, as expected. It can be seen that, for single packet messages, sending

the acknowledgement packet on the opposite ring does not induce a great

change in performance. The maximum throughput is slightly, though not

significantly, higher than for the case where the ACK packet travels on the

same ring.

When the messages consist of 10 packets (allowing the selective data

packet routing system to optimise the transit path) the results (Figure 5.18)

are markedly different. When the ACK packet travels on the same ring,

the maximum throughput is 6 Mb/s, i.e., as expected from the previous

section, 4 times the data rate. However, when the ACK packet travels on

the opposite ring, the maximum throughput increases to around 12 Mb/s,

which is 8 times the data rate. It is clear that the transit time for ACK

packets has been significantly reduced, reducing the time a node must wait

before sending another packet and hence increasing the overall throughput.
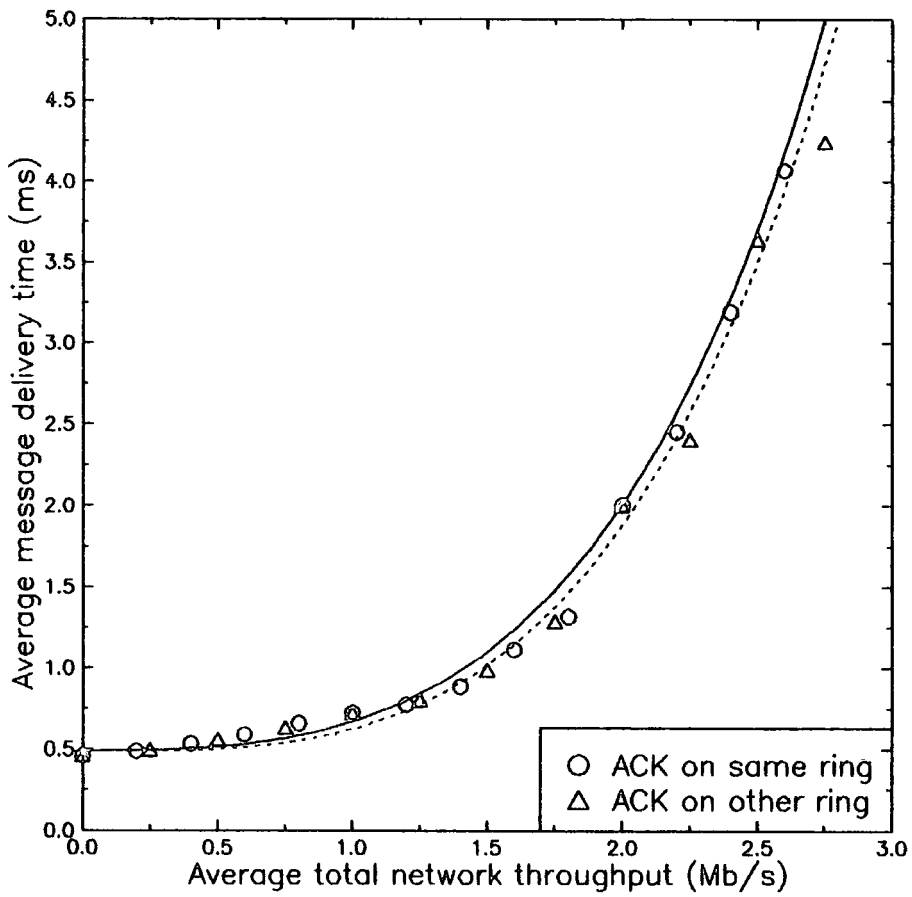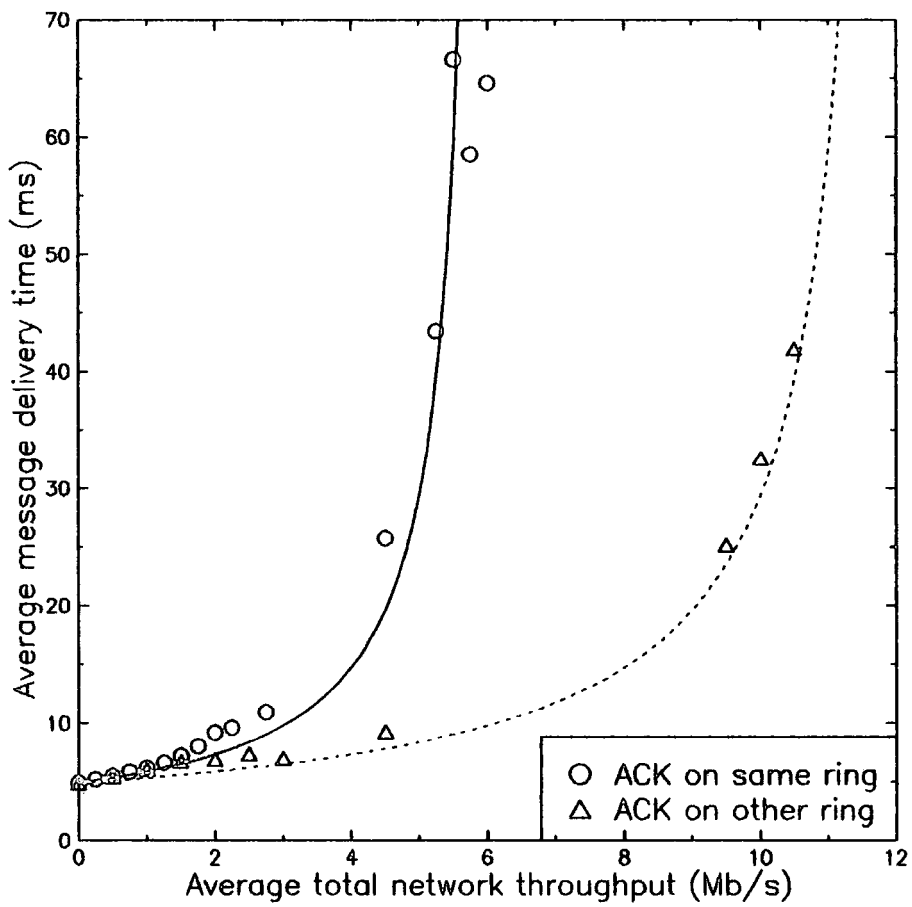
Figure 5.17: Dual RI ring, 1 packet messages

Figure 5.18: Dual RI ring, 10 packet messages

To summarise then, it can be said that the transmission of the acknowledgement packet on the opposite ring to that on which the data message arrived offers considerable performance improvements on the case where the ACK packet travels on the same ring. These improvements are particularly significant when messages consist of more than one packet.

Routing of data messages

Ignoring the effects of ACK packets now, the routing scheme for data packets is reassessed. The scheme that has been employed previously is summarised as follows. The first packet of each message is sent on both rings and the sender awaits the two acknowledgement packets. Based on which ACK packet arrives first, the sender chooses one of the rings for transmission of all the remaining packets of the message. This protocol aims to maximise use of available network bandwidth by minimising the length (in terms of bit delay) of the path between message source and destination. An assumption crucial to the efficacy of this method is that the traffic on the network should be in a reasonably steady state during the transmission of an entire message; if the traffic pattern changes halfway through a message transmission, the path which was selected at the commencement of the transmission may no longer be the 'shortest' one. This assumption holds for reasonably short messages, because the transmission time is relatively small, but it may not be reasonable for longer messages. In the latter case network bandwidth

121

could be wasted if the overall traffic pattern has a high rate of change.

One possible way to overcome this problem is to re-evaluate the chosen path during the transmission of a message. The path is recalculated every n packets using exactly the same technique as for the first packet. With this protocol the path is dynamic for each message and should be able to adapt to changes in the overall traffic pattern. Some simulation work was performed in order to investigate this technique. An example is shown in Figure 5.19. This message delivery time / throughput graph results from a 10 node, 1.5 Mb/s dual ring, with all message lengths being fixed at 10 packets. Three lines are shown, for values of n of 2, 5 and 8. The length of the simulation appears to have prevented the network from saturating at high loads. It is apparent that the more frequently the path is recalculated, the worse the network performance. The reason for this is that all nodes generate messages of the same length at the same rate, giving rise to a highly stable and symmetric, but unrealistic, traffic pattern. In this situation the overhead required to re-evaluate the path is more significant than the advantages to be gained by changing the path mid-message.

Figure 5.19: Recalculating path during message transmission

Chapter 6

Performance of the token ring

6.1 Analysis

6.1.1 Introduction

This section presents an analysis of the single and dual token ring protocols using analytical techniques. It is included for the purpose of comparison with the results obtained for buffer insertion. Analysis and simulation were used to investigate both protocols. The performance measure considered here is the mean message delivery time, again derived from work by Hammond and O'Reilly [11]. The assumptions made for the analysis of the buffer insertion

ring in Section 4.1 also apply in the case of the token ring. Once again,
the analysis has its foundations in queuing theory and results from [11] are
quoted here as a starting point for further work.

6.1.2   Single ring

We begin with an expression for the average normalised packet delivery time
T_{av} for the single token ring, operating in single token mode [11].

$$T_{av} = 1 + \frac{a'}{2} + \frac{a'[1 - S(e^{-a'} + a')/M]}{2[1 - S(e^{-a'} + a')]} + \frac{S[(a')^2 + 2(1 + a')e^{-a'}]}{2[1 - S(e^{-a'} + a')]} \quad (6.1)$$

where

M = number of nodes

a' = normalised ring latency delay

S = throughput

This equation will form the basis of the dual token ring analysis.

6.1.3   Dual ring

Early work on dual token rings suggested the employment of one ring for data transfer and a secondary 'backup' ring to be used only in the case of a fault developing on the primary ring [31]. This is the basic dual token ring protocol considered here. The performance of this protocol is, of course, the same as that of the single token ring, under normal operating conditions. It is easily observed that under such conditions the entire bandwidth of the secondary ring is wasted. In an attempt to improve upon this inefficient method, an enhanced version of the protocol has been developed [21], which operates as follows. When a station has a message to send across the network, it first waits for a token to arrive on either of the rings. It seizes the first token to arrive and sends as many packets of the message as possible (determined by the token holding time) using that token, all on the same ring. The other ring remains free for use by another node. When the token holding time has expired it releases the token and repeats the same procedure over again, until all packets of the message have been transmitted. Note that different packets of one message may travel on either ring. Should a ring break occur, the network can still reconfigure to form a single ring and continue operating, albeit with some performance degradation. Hence this enhanced protocol manages to boost performance while still retaining the fault-tolerant features of the basic dual ring protocol.

126

So, for the suggested improved protocol where both rings are used simultaneously, we have the equivalent of a second server in the queuing theory model. This effectively doubles the service rate, leading to an expression identical to Equation 6.1, but with S replaced by $S/2$.

6.1.4   Conclusion (Token ring analysis)

Figure 6.1 shows a comparison of single, FDDI-style dual, and improved dual token ring protocols. The following values for the network parameters have been used

o Exponentially distributed packet lengths

o Poisson generation

$M = 50$

$a' = 0.1$

Data Rate = 100 Mb/s

One may observe that the results for the single ring and the basic dual ring network are identical; no performance gain is obtained by using dual data paths, only better fault-tolerance. The enhanced version of the protocol, however, shows a potential doubling of the throughput, as would be expected
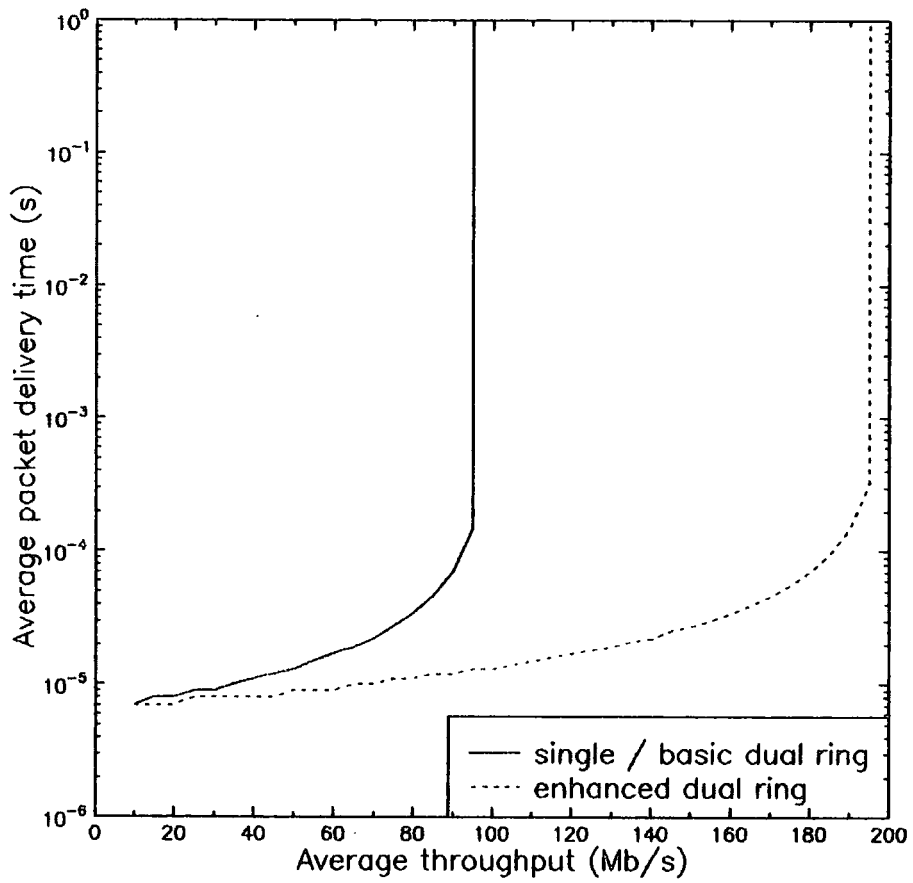
127

Figure 6.1: Comparison of three different token ring protocols

when dual data paths are used.

6.2   Simulation

6.2.1   Introduction

In order to investigate the performance of the token ring protocols a computer program was designed and implemented in SIMULA [24]. The DEMOS discrete event simulation package was used to simplify the implementation [1]. The simulation system was implemented on an Amdahl mainframe. Both single and dual rings were modelled; the dual ring version used the enhanced protocol, as the basic protocol is essentially the same as the single ring for performance evaluation purposes.

6.2.2   Simulations

Two types of 10 node ring were simulated: single ring and dual ring. The single ring is a typical 802.5 style LAN, whereas the dual ring incorporates

two single rings plus a higher level control system to manage message routing. The strategy used for the latter is as described in the previous section. As the token ring employs source removal of data packets, the acknowledgement signal can be added to the data message itself as it passes through its destination; a separate ACK packet is not required. The data packet header length and data rate were chosen to be 48 bits and 1.5 Mb/s respectively, for ease of comparison with the buffer insertion ring simulation. The data messages were all a fixed length of 10 packets of 640 bits. Delays in the nodes have been neglected for simplicity, and the message generation follows a Poisson distribution.

Figure 6.2 shows the curve for the single ring. The shape is as expected, but the throughput does not exceed the data rate as it does for the single buffer insertion ring. This is because the token ring protocol only allows the transmission of one message at a time, which is removed by the source node. Hence the network cannot take advantage of certain destination traffic patterns.

Considering the dual token ring now (Figure 6.3), it can be seen that the maximum throughput is double the data rate. Because the first available ring is chosen for each message, the other ring is free for use by another node; thus two independent transmissions may occur simultaneously, giving
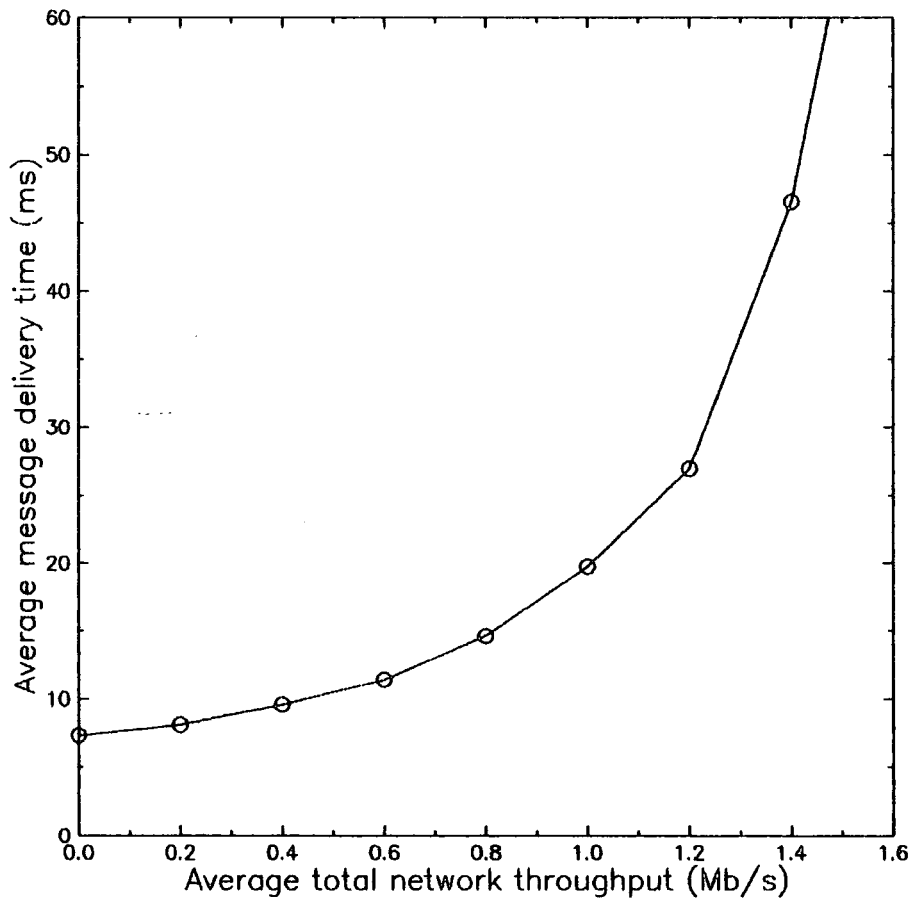
Figure 6.2: Single token ring

Figure 6.3: Dual token ring

a maximum (normalised) throughput of 2.

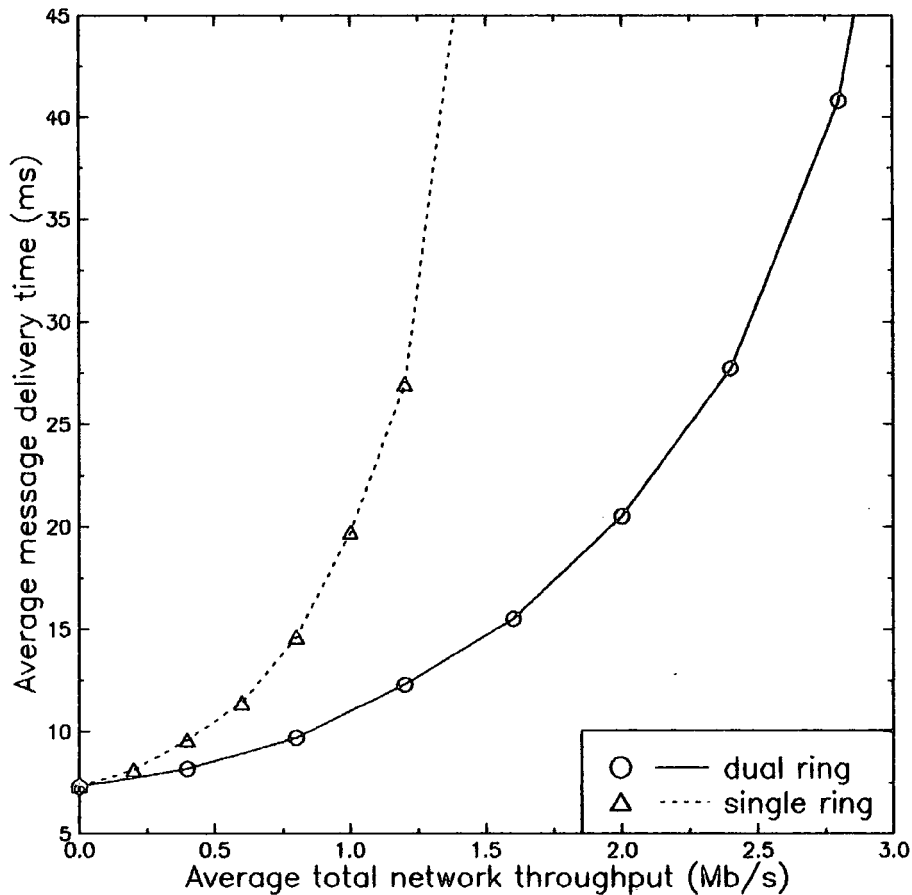Figure 6.4: Single and dual token rings

Comparing single and dual rings (Figure 6.4) it can be seen that the performance of the dual ring is considerably better than that of the single ring. However, in this respect, the performance of the token ring protocol is poorer than that of the dual buffer insertion ring, which can provide maximum throughputs of 4 and even 8 times the channel data rate.

Chapter 7

Conclusions

This chapter presents a summary of the conclusions that can be drawn from the results presented previously.

The quality of service offered by a LAN, usually measured in terms of delay time or throughput, is typically assessed by mathematical analysis, simulation or measurements on networks *in situ*. Analysis has the advantage of providing immediate results, but it relies on a strict set of simplifying assumptions for its correctness, e.g. uniform loading on the network and identical behaviour of all nodes. It is thus at its most useful when great accuracy is not required and results must be obtained quickly. Where the parameters are different for each node and when the network load is uneven analytical solutions are not easy to derive. To obtain a more detailed and flexible model of a LAN, simulation is preferred. It is a much slower process

134

than analysis, particularly when the network consists of a large number of nodes, but it has the advantage of being able to represent the network to any required degree of detail. Traffic patterns are user-definable, as are all the parameters of the individual nodes. Simulation may also be used to verify the model derived by analytic methods, and to test its accuracy. Measurements on real networks can produce meaningful results when they are carefully designed and analysed.

Experimentation with a real network allowed detailed evaluation and study of the operation of two different LANs. The unexpected results that were obtained in some cases required careful consideration. It is clear that the measured values differ markedly from those predicted by analysis and simulation. Due to the complexity of today's networking hardware/software, mathematical or simulation models can only reasonably represent the lower-level functions of LAN protocols, typically the data link and physical layers. It is generally not specified how the packets arrive at the data link layer for transmission; similarly, the destiny of packets arriving from the physical media is rarely considered beyond layer 2. The processing of packets between this layer and the application program may be carried out in hardware or software, and data may flow across other (e.g. RS232) communication channels as well as on the bus of the network interface unit's microprocessor. This processing is subject to a variety of factors, many of which are not eas-

ily quantifiable, and can give rise to time delays and throughput bottlenecks which may be orders of magnitude larger than the time involved to transmit packets on the physical media. Hence mathematical analysis and simulation may be of use in the comparison of different medium access protocols, but are severely limited in their ability to model networks as they are used in 'real' applications unless these higher level delays are taken into account. In this respect, the technique of integrating simulation with parameters that have been obtained from measured data has proved to be more useful. This 'hybrid' approach has resulted in the production of a simulation model which provides a more refined representation of the actual network under consideration.

Despite the fact that, in real terms, a network's performance may be limited by factors other than its medium access method, the lower-level protocols in the OSI stack do have significance, particularly when fault-tolerant behaviour is desirable, or when the network is likely to be subjected to high offered loads. In the latter situation, the buffer insertion protocol performs particularly well. It has been shown how the unique simultaneous use of data links between nodes can give rise to total network throughputs greater than 100%. In the experiments performed in the laboratory, LANs of different data rates were used. The raw performance data was normalised to allow comparison. However, it is necessary to be cautious as not all features of

LAN performance can be scaled in this way. Using the raw data, an Ethernet LAN performed better than a buffer insertion based network, but this was due to its higher data rate; when statistics were presented relative to the data rate, the buffer insertion LAN gave a better all-round performance. However, this study was based on two particular implementations of these protocols and, for reasons given in the paragraph above, may not be valid for *all* implementations of Ethernet and buffer insertion. Where the operation of the higher layers can be considered a constant it provides a basis for performance comparisons between different LANs.

When structured as a pair of contrarotating rings, a buffer insertion LAN achieves even greater throughputs due to a doubling of the available bandwidth and also to strategies which aim to optimise packet routing on the rings. Such LANs offer strong competition to the currently popular dual token ring networks, even those which employ simultaneous use of both physical channels. As an illustration, consider the following example.

Figure 7.1 compares the performance of the buffer insertion and token ring protocols studied in this thesis. All graphs are derived from results from previous sections. For ease of comparison, all the results assume a data rate of 1.5 Mb/s. In terms of maximum throughput, the worst performance is clearly exhibited by the single token ring, which achieves a maximum
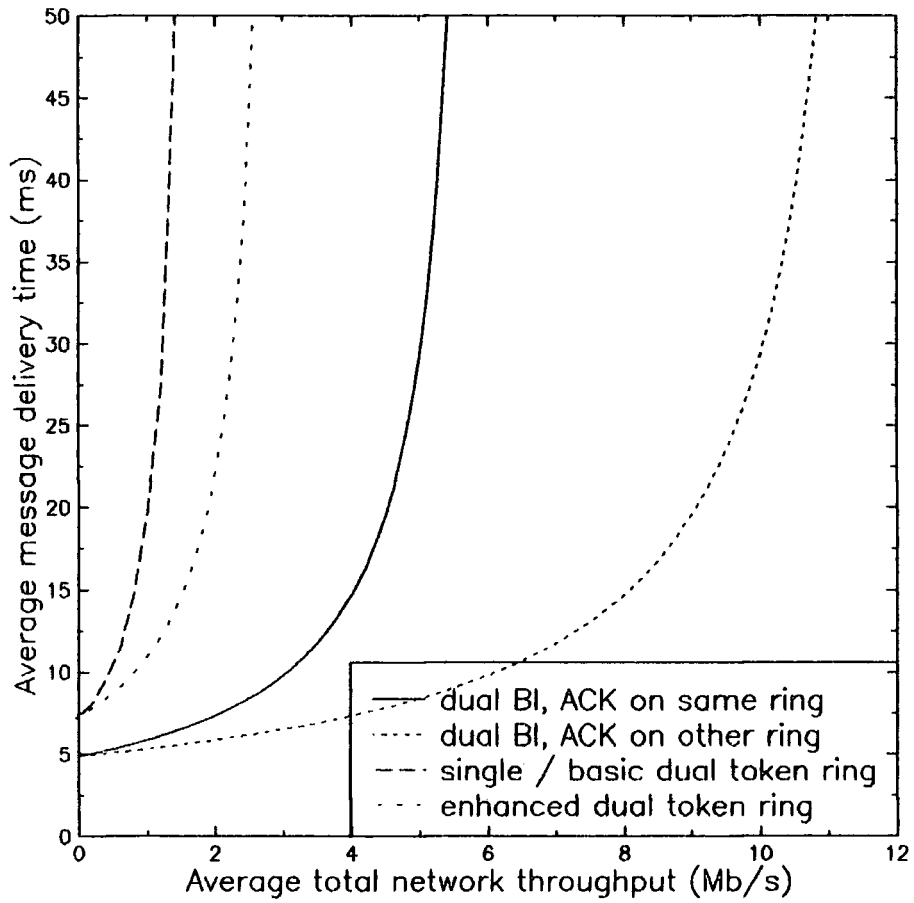
Figure 7.1: Some LAN protocols compared

throughput of 1.5 Mb/s (1.0 normalised), the same as the channel data rate. This limitation is inherent in the single token protocol, which only allows one transmission to take place at any one time. The basic dual token ring has a performance identical to the single token ring, because it is in effect a single ring with a backup ring that is unused under normal operation; its advantage over the single ring is that it provides a measure of fault-tolerance in its ability to react to cable breaks and node failures.

The enhanced dual token ring protocol manages to achieve a maximum throughput of 3 Mb/s (2.0 normalised). This is due to its use of both rings for non fault condition data traffic, effectively doubling the network bandwidth. It also retains the fault-tolerant features of the basic dual token ring.

Considering the dual buffer insertion rings, one observes that in the case where the acknowledgement packet travels on the same ring as the data message, the maximum throughput is 6 Mb/s (4.0 normalised). This figure derives from the fact that a single buffer insertion ring may achieve a maximum throughput of twice the data rate, due to its destination removal of packets policy, and from the dual ring routing strategy which attempts to use the 'quickest path' for data transmission. That is, for a single ring a message on average only travels half way around the ring, leaving the other half free for simultaneous traffic. This spacial reuse produces a maximum

throughput of two times the data rate. With two rings this results in four times the data rate of one ring. (Note the bandwidth for dual/multiple rings could be considered as being the sum of the bandwidths of the individual rings, but for consistency all results are related to a single ring bandwidth.) A fault-tolerance scheme is achieved by ring reconfiguration techniques.

Finally, the dual buffer insertion ring with acknowledgement packets being sent on the opposite ring achieves a maximum throughput of 12 Mb/s (8.0 normalised). This substantial improvement in throughput derives partly from the reasons mentioned in the previous paragraph, and partly from the further decreased transmission waiting delays obtained by optimal routing of acknowledgement packets.

The significance of these results becomes apparent when one considers the cost / performance ratio of the technologies involved. A typical commercial dual token ring LAN is based on optical fibres and transceivers in order to provide a fast (100 Mb/s) data rate, while employing a backup ring to recover from certain types of fault. A dual buffer insertion ring, by using the techniques described, may achieve a 100 Mb/s maximum throughput while using a data rate as low as 12.25 Mb/s, and still provide some degree of fault-tolerance. (The transit time for individual messages will of course be greater, however.) Data rates of this magnitude can easily be accommo-

dated on twisted pair cable; at 12.25 Mb/s the electronics of the physical layer are 'lower tech' and hence lower cost. Line drivers for twisted pair are readily available and can be used for distances of the order of 100 to 500m. Additionally, a very large installed base of such cabling exists, whereas the switch to fibre can involve a huge investment in terms of installation and maintenance costs. When one considers such aspects, the advantages of the buffer insertion scheme are apparent.

Buffer insertion would therefore seem to be an excellent choice of protocol for use in high-performance local or metropolitan area networks of the future, as evidenced by IBM's employment of it in a recent product [4]. The routing strategy presented in this thesis facilitates high throughputs with a reasonably low overhead, particularly for long messages. It may be worth investigating alternatives, such as the adaptive strategy described in Section 5.2.2 or a more straightforward 'shortest physical path' technique. The latter has a lower overhead than the 'quickest path' method, but it may not always find the fastest route to the destination. Additionally, the packet acknowledgement protocol may be subject to further experimentation. The throughput is maximised when the acknowledgement packet travels on the opposite ring to the associated data packet. However, this performance increase could be made even greater by increasing the data link window size, allowing further packets to be transmitted without waiting for an immediate

acknowledgement. Even more significant benefits should result if all data packets went unacknowledged at the data link layer. The technology and transmission media used in modern LANs afford a very low error rate, leading to a low incidence of packet loss or corruption. Hence the data link layer could offer an unacknowledged datagram service, and higher layers could perform their own error detection and correction if required.

References

[1] G.M. Birtwistle. *DEMOS - Discrete Event Modelling on Simula.* Macmillan, 1979.

[2] B.D. Bunday. *Basic Queueing Theory.* Arnold, 1986.

[3] W. Bux and M. Schlatter. "An Approximate Method for the Performance Analysis of Buffer Insertion Rings". *IEEE Transactions on Communications*, COM-31(1):50–55, January 1983.

[4] I. Cidon and Y. Ofek. "Metaring - A Full-Duplex Ring with Fairness and Spatial Reuse". Technical report, IBM Research Division, September 1989.

[5] W.D. Farmer and E.E. Newhall. "An Experimental Digital Switching System to Handle Bursty Computer Traffic". In *Proceedings, ACM Symposium on Problems in the Optimization of Data Communications*, 1969.

[6] W.R. Franta. *Local Networks*. D.C. Heath and Company, 1981.

[7] J.S. Grant and C.R. Grant. "A Buffer-Insertion Ring Suitable For Time-Critical Applications". In *Proceedings, EFOC-LAN 89*, pages 71–75, 1989.

[8] K. Grant. "Interim report of the TCCA rapporteurs' group of ISO/TC184/SC5/WG2 on time-critical communications architecture and systems". Technical report, International Standards Organisation, 1989.

[9] E.R. Haffner, Z. Nenadal, and M. Tschavz. "A Digital Loop Communications System". *IEEE Transactions on Communications*, COM-22(6):877–881, June 1974.

[10] F. Halsall. *Data Communications, Computer Networks and Open Systems*. Addison-Wesley, 1992.

[11] J.L. Hammond and P.J.P. O'Reilly. *Performance Analysis of Local Computer Networks*. Addison Wesley, Reading, MA, USA, 1986.

[12] D. Huber, W. Steinlin, and P. Wild. "SILK: An Implementation of a Buffer Insertion Ring". *IEEE Journal on Selected Areas in Communications*, pages 766–774, November 1983.

[13] D. Hutchison. *Local Area Network Architectures*. Addison-Wesley, 1988.

[14] IBM. "IBM Series/Local Communications Controller Feature Description GA34-0142-2". Technical report, IBM Corporation, September 1982.

[15] IEEE. "Data Communications Special Guide". *IEEE Spectrum*, August 1991.

[16] ISO/TC97 International Standards Organisation. *Open Systems Interconnection - Basic Reference Model*. Draft International Standard ISO/DIS 7498, 1982.

[17] C. Jackson and J.E. Mellor. "Bandwidth usage optimisation in dual ring LANs". In *Bangor Communications Symposium*, pages 271–274, May 1991.

[18] C. Jackson, J.E. Mellor, and G.M. Geary. "Hierarchical CIM Networking". *Computer-Integrated Manufacturing Systems*, 2(4):207–213, November 1989.

[19] M.T. Liu. "Distributed Loop Computer Networks". *Advances in Computers*, 17:163–221, 1978.

[20] M.T. Liu, G. Babic, and R. Pardo. "Traffic analysis of the distributed loop computer network (DLCN)". In *National Telecommunications Conference*, pages 31:5-1–31:5-7, December 1977.

[21] J.E. Mellor and C. Jackson. "Performance Evaluation of Time Critical Fault Tolerant Local Area Networks". In *Proceedings, BCS 6th UK Computer and Telecommunications Performance Engineering Workshop*, pages 9/1-9/9, July 1990.

[22] R.M. Metcalfe and D.R. Boggs. "Ethernet: Distributed Packet Switching for Local Computer Networks". *Communications of the ACM*, 19(7):395-404, July 1976.

[23] The Institute of Electrical and Electronics Engineers. "Token Ring Access Method and Physical Layer Specifications". *American National Standard ANSI/IEEE Std 802.5-1985*, 1985.

[24] R.J. Pooley. *An Introduction to Programming in SIMULA*. Blackwell Scientific, 1987.

[25] C.C. Reames and M.T. Liu. "A Loop Network for Simultaneous Transmission of Variable-Length Messages". In *Proceedings, Second Annual Symposium on Computer Architectures*, pages 7-12, 1975.

[26] J. Salter and P. Evans. "The IEEE 802.6 Standard for MANs, its Scope and Purpose". In *EFOC/LAN 90, the Eighth Annual European Fibre Optic Communications and Local Area Networks Conference*, pages 151-163, June 1990.

[27] M. Schwartz. *Telecommunication Networks: Protocols, Modeling and Analysis*. Addison-Wesley, 1987.

[28] R. Simha and Y. Ofek. "A starvation-free access protocol for a full-duplex buffer insertion ring local area network". *Computer Networks and ISDN Systems*, 21:109–120, 1991.

[29] W. Stallings. *Local and Metropolitan Area Networks*. Macmillan, 1993.

[30] A. Thomasian and H. Kanakia. "Performance study of loop networks using buffer insertion". *Computer Networks*, 3:419–425, December 1979.

[31] E.A. Zurfluh and H.R. Muller. "Automatic Configuration for Fiber-Based LANs". In *EFOC/LAN 88, the Sixth European Fibre Optic Communications and Local Area Networks Exposition*, pages 408–413, June 1988.