Performance Measurement Methodology for Integrated Services Networks

by

Mahboob-ul-Haq Siddiqui, BSc (Eng.)

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy of the University of Technology, Loughborough.

December, 1989
Supervisor: Dr. D.J. Parish

Department of Electronic and Electrical Engineering
University of Technology, Loughborough
England.

In Memory of

My Beloved Parents

ABSTRACT

With the emergence of advanced integrated services networks, the need for effective performance analysis techniques has become extremely important. Further advancements in these networks can only be possible if the practical performance issues of the existing networks are clearly understood. This thesis is concerned with the design and development of a measurement system which has been implemented on a large experimental network.

The measurement system is based on dedicated traffic generators which have been designed and implemented on the Project Unison network. The Unison project is a multisite networking experiment for conducting research into the interconnection and interworking of local area network based multi-media application systems. The traffic generators were first developed for the Cambridge Ring based Unison network. Once their usefulness and effectiveness was proven, high performance traffic generators using transputer technology were built for the Cambridge Fast Ring based Unison network. The measurement system is capable of measuring the conventional performance parameters such as throughput and packet delay, and is able to characterise the operational performance of network bridging components under various loading conditions. In particular, the measurement system has been used in a 'measure and tune' fashion in order to improve the performance of a complex bridging device.

Accurate measurement of packet delay in wide area networks is a recognised problem. The problem is associated with the synchronisation of the clocks between the distant machines. A chronological timestamping technique has been introduced in which the clocks are synchronised using a broadcast synchronisation technique. Rugby time clock receivers have been interfaced to each generator for the purpose of synchronisation.

In order to design network applications, an accurate knowledge of the expected network performance under different loading conditions is essential. Using the measurement system, this has been achieved by examining the network characteristics at the network/user interface. Also, the generators are capable of emulating a variety of application traffic which can be injected into the network along with the traffic

from real applications, thus enabling user oriented performance parameters to be evaluated in a mixed traffic environment.

A number of performance measurement experiments have been conducted using the measurement system. Experimental results obtained from the Unison network serve to emphasise the power and effectiveness of the measurement methodology.

ACKNOWLEDGEMENTS

I would like to acknowledge the financial support provided by the Ministry of Science and Technology, Government of Pakistan, which made this research possible.

I would like to thank Dr. D.J. Parish for his guidance and encouragement throughout the research. Special thanks must also extend to Professor C.J. Adams and Professor J.W.R. Griffiths for their invaluable advice and discussion during the course of this research. I wish to thank the many members of the Project Unison for helpful discussions, and for their comments and feedback while the research was in progress.

I would also like to thank Dr. H.S. Chin and B.J. Murphy for reading and commenting on the final draft of this thesis, to Mrs. C. Clarson and to all my friends in the Signal Processing Laboratory of Loughborough University for making this an enjoyable and memorable time.

My heartiest gratitude goes to my family, in particular to my brother Azhar-ul-Haq, for their constant encouragement and moral support.

M.H. Siddiqui

December, 1989.

SYMBOLS AND ACRONYMS

ARPANET	Advanced Research Project Agency Network
CCITT	Consultative committee on International Telephone and Telegraph
CLK	Clock
CFR	Cambridge Fast Ring
CR	Cambridge Ring
CRC	Cyclic Redundancy Checksum
EPROM	Erasable Programmable Read Only Memory
FIFO	First In First Out
IEEE	Institute of Electronic and Electrical Engineering
IMP	Interface Message Processor
ISDN	Integrated Services Digital Network
K	Kilo
LAN	Local Area Network
LUT	Loughborough University of Technology
M	Mega
MAC	Medium Access Control
MDS	Multicast Data Server
NBSNET	National Bureau of Standards Network
NPL	National Physical Laboratory
SATNET	Satellite Network
SBC	Single Board Computer
SSP	Single Shot Protocol
SSPREQ	SSP Request
SSPREPLY	SSP Reply
RAL	Rutherford Appleton Laboratory
RAM	Random Access Memory
RCR	Rugby Clock Receiver
RPC	Remote Procedure Call
TDS	Transputer Development System

TG	Traffic Generator
TR	Traffic Receiver
TGR	Traffic Generator and Receiver
TOG	Thrown On Ground
WAN	Wide Area Network

DEFINITIONS AND TERMINOLOGY

Client LAN

It refers to a LAN to which application components are attached. The application components can be classified as "clients", which require services from the LAN, or "servers", which provide services to the clients.

Exchange LAN (backbone LAN)

It refers to a LAN which interconnects client LANs in order to support inter-client network communications.

Light-weight Virtual Circuit

It is a logical channel between source and destination stations on a packet-switched network. A virtual circuit may require some form of "set up" which may or may not be visible to the user. Packets sent on a light-weight virtual circuit are delivered in the order sent, but without any guarantee about their delivery. No flow control or error recovery are provided by the network.

Minipacket

It is the unit of information transmitted between ring stations controlled by the ring access mechanism.

One-way Packet Delay

It refers to the elapsed time between a packet generation and its arrival at the destination station. It includes the packet launching time at the network/user interface, delay experienced at the packet switching components, and propagation delay.

Packet Switch (bridging Components)

It is a collection of software and hardware resources which implement inter-network procedures such as forwarding of traffic between network segments, and traffic routing in order to support inter-network communication. Protocol conversion is not a function of the packet switches.

Protocol

A protocol is a set of communication conventions, including formats and procedures, which allow two or more end-points to communicate. The end-points may be packet switches, clients or servers.

Rugby Clock

It is a broadcast time transmission facility used to provide accurate absolute timing information over a limited geographical area. The time transmitter is situated near Rugby, covering a radial distance of around 1000 kilometer.

Throughput

This metric is defined as the sum of packets successfully received (without duplication) at the destination in a unit time.

Traffic Generator and Receiver (TGR)

It is a collection of software and hardware which are used to generate and receive data traffic on the network for the purpose of performance measurements.

U-Channel

It refers to the aggregation of ISDN slots in order to provide variable bandwidth between peer sites for inter-site communications. One ISDN slot is equal to 64 kbits/sec.

TABLE OF CONTENTS

Chapter 1

Introduction

1.1 Background

1.1.1 Performance Issues in Networks

Ever since the advent of digital computers, there have been growing technological advancements in computer networking. The last two decades have witnessed several impressive developments in the forms of Wide Area Networks (WANs), Local Area Networks (LANs), and more recently, integrated services networks.

Packet-switched WANs are basically made up of packet switches and communication lines which interconnect geographically dispersed packet switches. The packet switches, often based on minicomputers, perform store-and-forward functions and make routing decisions. The network transports computer data in the form of packets. A number of public and private WANs now exist which allow computer users to access remote machines for file transfer, electronic mail, and remote job entry. The performance issues in these networks primarily relate to the performance characteristics of the switches and performance of the communication protocol used between switches. The maximum traffic handling capacity and packet transit delay across switches would depend on the efficiency of the routing algorithms, queueing and buffering strategies, and processing speeds of the switches. The communication protocols used in WANs are quite complex in order to deal with the large transmission times, low bandwidths, and high error rates associated with the interconnecting lines.

The performance of the protocol reflects directly on the efficient and fair use of the network resources.

The last decade has seen an increase in the use of LANs to provide improved communication performance over a limited geographical area. Taking the advantage of limited transmission distances, LANs offer a high bandwidth, low transit delay, and low error rate communication media. These characteristics contradict with those of the WANs. LANs, while supporting conventional computing applications, can also be used to allow the distribution of computing systems. Distributed computing systems offer benefits in terms of performance, reliability and flexibility. In a local area network context, performance issues primarily relate to the throughput-delay characteristics of the medium access layer. There are a few disadvantages associated with the LAN technology. These include lack of management and diagnostic facilities. Some networks are vulnerable to a single failure.

Recently, there has been a growing interest in integrated services networks. These are networks that aim to support a variety of services including voice, video, graphics, and distributed computing applications. Such networks promise to meet all the communication requirements of the future. A number of networking architectures are emerging for integrated services wide area networks. The LAN based architectures, sometimes referred to as Linked Local Area Networks (LLAN) are currently receiving increasing attention. LLANs aim to extend the functionality of LANs to a wider area. However, in so doing, a number of challenges have to be met. The performance issues in integrated services networks are wide-ranging. In an integrated services environment, the network should meet the differing performance criteria imposed by each application. The performance criterion of one application may conflict entirely with that of another. The extent to which the differing performance criteria are met by the network is a primary issue in any integrated services network. Another issue is concerned with the performance of the bridging components. With the advances in VLSI technology, these components provide faster data switching in comparison with their predecessors. However, most current applications require real time response as well as high throughput for communications, the provision of which makes these devices extremely complex and unpredictable in their behaviour. There are many other complex issues which at the moment are not properly understood and need more investigation. For instance, the issue of fairness, i.e., does the network provide fair sharing of network resources among mixed users.

1.1.2 Performance Evaluation Techniques

The performance evaluation of communication networks has been, and will remain an important aspect because it helps to investigate the various performance issues which arise while designing, developing, operating, and using networks. There are three performance evaluation techniques commonly used in studying a network : analytical techniques, simulation techniques and measurements. Analytical and simulation techniques see wide spread use in network planning and designing whilst measurements involve physically measuring a real network.

Over the past years, a large body of research in the general areas of networks has been analytical and theoretical. A number of models (analytical and simulation) have been developed which provide immense help to the network designers. While theoretical investigations of networks are of great importance, experiments in real-world situations have to be carried out to verify and possibly extend these theoretical results, and pinpoint actual problems.

A number of experimental techniques have been developed which allow investigations to be carried into various performance issues with real networks. The measurement and monitoring work has been pioneered by one of the early WAN, ARPANET [Tobagi 78]. This network uses a variety of experimental techniques. For instance, cumulative statistics, trace statistics, and snapshot statistics collection techniques have been used in a number of performance measurement experiments. These techniques have been used for network accounting (e.g. collecting statistics regarding the user traffic characteristics). Furthermore, these techniques provide statistics on utilisation and availability of network resources, and allow an instantaneous status of the network components to be looked at. The message generation facility provided in ARPANET had been used for network debugging and protocol verification. Similar monitoring facilities exist in other experimental networks. In SATNET [Treadwell 80], a timestamping technique had been used to evaluate packet transit delay across the network. In this technique, special packets are timestamped at the entry and exit points of the packet switches, thus enabling packet transit delay to be measured between two network end-points. In SATNET, the end-points were the packet switches themselves. Unfortunately, very little work has been done in observing the network performance from a user perspective.

Most of the performance evaluation work on LANs has been theoretical and usually revolves around the topics of network efficiency and upper bounds on the time delay. User throughput performance under dynamic loading conditions has rarely been discussed. Analytical and simulation models have been developed for various LAN access protocols which provide valuable insight into the overall LAN efficiency. There also exists some monitoring work carried out over certain LANs. In NBSNET [Amer 82], a broadcast local area network, a number of performance experiments had been carried out to determine LAN performance under typical loading conditions. The investigations were mainly concerned with the verification of the medium access protocol used in the LAN under study. Also, there has been some monitoring work on slotted ring LANs [Hopper 86]. A monitoring station was developed for the ring LAN to monitor the traffic on a service ring. The idea was to determine the user traffic characteristics and to evaluate the utilisation of the LAN.

Integrated services networks are relatively new to the networking arena. Within this decade, a few experimental networks have already emerged. In the UK., a project called Universe [Burren 89a], had developed a networking architecture in order to support integrated services over a wide geographical area. The Universe network incorporated a number of performance monitoring tools. The monitoring tools were very similar to the ones used in the early WANs. Similarly, in other experimental networks, there has been a general tendency towards using the same monitoring tools developed for the early WANs. These tools provide network gross statistics which are more useful to the network manager than to the network user. Experience with the Universe project has shown that it is not enough to monitor network performance with gross figures, measurements of those traffic parameters which have some correlation with the application traffic are also required.

The measurement system for integrated services networks must address the issues which are important to the network users. For instance, it is important to determine how the network behaves with various user streams in a mixed traffic environment. Such issues are particularly important to real time applications. Real time services require guaranteed bandwidth after the establishment of a connection and cannot tolerate throughput degradations and loss of efficiency due to network overload. The measurement techniques should address these issues in greater depth and should practically determine how such problems affect time-critical services. Unfortunately,

very little attention has been given so far to developing such measurement tools which can effectively address these issues.

1.1.3 Performance Measures

The traditional performance measures used in networks are the average delay and average throughput. These are the parameters which are normally used to characterise and quantify the performance of a real network.

The throughputs of network components are perhaps easy to measure but some complications can arise in measuring the delay. Delay, although conceptually simple to define, is not always straightforward to measure. For example, the end-to-end delay cannot in general be measured due to lack of clock synchronisation between the distant nodes. The most common substitute for end-to-end delay is the round-trip delay, i.e., the interval between transmission of a test packet and the acknowledgment from the destination. The round-trip delay divided by two provides the one-way delay measurement assuming that the network characteristics are the same in both the outward and return paths. Often this is not the case. Being the easiest and cheapest solution, the round-trip delay measurement is the most commonly used method in WANs.

There are other more accurate delay measurement techniques which involve using intermediate network nodes to measure the transit delay. The test packets are stamped with the entry and exit time at each intermediate node. The total delay is then the sum of the times spent at each node. Some measurement errors may still exist between the actual arrival/departure of a packet and the timestamp. Most importantly, such a measurement mechanism disturbs the node operation. In an integrated services environment, such a disturbance cannot be tolerated.

An important requirement for a measurement system is the possession of a suitable mechanism which can measure one-way delay accurately, and at the same time, does not have any adverse effect on the network nodes. This can only be made possible if the clocks of the distant machines are synchronised. Maintaining synchronised clocks in a distributed system itself is a complex problem.

1.2 Project Unison

Project Unison [Clarke 86], a collaborative research project, was established to conduct research into local and wide area networking techniques for office applications. The objectives of Unison were to build an experimental network testbed and multi-media office application components in order to carry out experiments on the suitability of the network for office applications. The office applications included voice, video, graphics, and conventional computing applications [Murphy 88].

The Unison network was composed of a number of geographically dispersed LANs interlinked by an Integrated Services Digital Network (ISDN). The office components are connected to the 'client' LANs which in turn are connected to the Unison Exchange [Tennenhouse 87]. Offices may be established on the same client network, on different client networks on the same site, or on remote client networks at different sites. Communications between different offices then occur via different parts of the Unison internetwork structure. Unfortunately, the different parts of the network have different characteristics and hence communication performance between offices depends upon their locations within the Unison internet. The performance of the network obviously has a critical effect upon the operation of the office components. An accurate knowledge of the expected network performance is essential when designing office applications.

The Unison network has incorporated a number of LAN-LAN and LAN-ISDN bridging components. Under certain loading conditions, these components can exhibit highly non-linear characteristics. These characteristics are difficult to predict, and almost impossible to analyse when a number of such components are involved in a communication chain. A solution to the above problem would be to measure the actual performance of the network when carrying the required traffic load. It would be possible, by varying this load, to determine experimentally the operating characteristics of the network bridging components. The problem would be one of being able to monitor the network traffic whilst providing realistic traffic loads.

1.3 Research Aims

This thesis is concerned with the development of a performance measurement methodology for integrated services networks. The performance measurement strategy is difficult to realise without a physical network, so in developing the measurement strategy, consideration was given to the requirements for the Unison network. However, the scope of the technique is extendible to other similar networks. The research work has the following aims

- To develop a practical measurement system with the following objectives in mind:
    - The measurement system should be able to characterise the operational performance of the network bridging components under various loading conditions.
    - It should be able to pinpoint weak links within the internet and provide an on-line source of performance statistics which can be used by the network designer to optimise the performance of the network components.
    - It should be able to measure the network performance as seen by the "user", i.e. as accurately as possible over real network routes. This involves the ability to measure the one-way packet transit delay accurately between remote users.
- To implement the measurement system on the network
- To assess the performance of the measurement strategy, and perform measurements and practical analysis on the network.

1.4 Thesis Organisation

Chapter 2 reviews the existing performance evaluation techniques with particular emphasis to network measurement techniques. An overview of the various measurement tools employed in other networks is provided. Finally, a measurement methodology is proposed.

Chapter 3 describes the various design considerations in developing the measurement system. The main building blocks of the measurement system are described. In particular, a chronological timestamping technique is introduced. This technique is used to measure one-way packet delays across the network.

Chapter 4 describes an implementation of the measurement system on the Cambridge Ring (CR) based Unison network. The CR based Unison network is explained which is essential to understand the measurement system completely. Using the measurement system, a number of performance experiments were carried out over the network. The results of these experiments are included.

Chapter 5 starts with an introduction to the Cambridge Fast Ring (CFR) based Unison network. The measurement system for the CFR based network is then described. The details of the performance experiments with various network configurations are presented along with some typical test results.

Chapter 6 describes how the measurement system was used to improve the performance of a typical LAN-LAN bridging device. The traffic switching algorithm is studied in greater depth. Some typical results with general conclusions are presented.

Chapter 7 contains a discussion of the work and the results, and presents the conclusions drawn from the research.

Chapter 2

Performance Measurement Methods

2.1 Introduction

Performance measurements of communication networks are needed in order to compare and characterise the important network characteristics. Although theoretical analysis of a communication network is essential for its first implementation, once the network is built, measurements allow insight into the network behaviour to be gained and allow real practical problems to be tackled effectively. The results of network measurement experiments enable the network design flaws and inefficiencies to be detected, and that leads to improvement in the network design. Although considerable advances have been made in the area of theoretical network analysis (a pre-requisite to developing a physical network), the performance studies of real networks using measurements are scarce [Reiser 82].

This chapter reviews existing measurement methods employed in various networks. The capabilities and limitations of these measurement methods are discussed. Other than measurement methods, there are theoretical techniques used for network performance evaluation which are briefly reviewed at the start of the chapter. A measurement methodology is proposed and a discussion section is included to describe the advantages and main features of the proposed methodology.

2.2 Performance Evaluation Techniques

Performance evaluation of communication networks is essential for sustaining the network evolution process. Not only does it help in predicting the network performance in quantitative and qualitative terms but it also helps in designing the next generation of advanced networks which will fulfill our current communication needs. Performance evaluation techniques can be broadly classified into two groups.

a. Predictive Modelling

b. Measurements

Predictive modelling is a theoretical technique which is mainly employed in the designing and planning stage of the network, whilst measurements involve physically measuring a real network. Employing techniques of predictive modelling can often be cheaper and far simpler than performing real measurements. Also, there are circumstances, such as at the design stage of a network, where real measurements are quite impossible. However, even though modelling methods are extremely powerful, they have their limitations. Most of the modelling work is based on simplifying assumptions without which the analysis becomes intractable. With these assumptions, the results from the models may not conform to the real situations. Furthermore, most models produce gross performance statistics that apply to the network as a whole and these models cannot be used to predict the performance that each user will encounter.

2.2.1 Predictive Modelling

In designing or developing a new network, it is difficult to predict its performance unless a detailed analysis of a similar network has been carried out. Suitable information on a similar network is not usually available so predictive modelling is employed which helps in understanding the expected performance characteristics before practically building the real network. Three forms of predictive modelling are used which are discussed below.

2.2.1.1 Analytical Modelling

Analytical models are mathematical representations which relate the system outputs to the inputs by defining functional relationships between system variables. Many mathematical tools have been developed over the years to analyse network design and operation. For example, queueing theory has a close relationship with wide area network technology. It is being used to model the behaviour of traffic switching components, when connected arbitrarily in a wide area network architecture. In particular, it has been used to evaluate transit delays in store-and-forward networks. The single server queue model has been widely used in this regard. In this model, packets are seen to arrive at the switching node, wait their turn for service, and then depart onwards. The probability distribution of packet inter arrival rate and service time are usually assumed to follow an exponential distribution. Such queueing models are referred to as M/M/1 systems [Tanenbaum 88]. These models have been used in various networks, such as ARPANET [Kleinrock 76], to predict transit delays, buffer requirements, buffer strategies and node processing time.

Local area networks are complex multi-layer systems, their analytical analysis has to be performed at various levels of detail. Primarily, the performance investigations of the local networks are related to the medium access layer, e.g., their throughput-delay characteristics. Investigation of the access protocol can provide valuable insights into the overall efficiency of the local network. A number of analytical models have been developed for various access protocols, particularly for Carrier Sense Multiple Access with Collision detection (CSMA-CD) and slotted ring access protocols.

2.2.1.2 Simulation Modelling

Simulation models are computer programs which use software routines to model the relationships between system inputs and outputs. The working of a network can be modelled to any desired level of detail if the necessary system relationships are known. However, the time and effort required to model the system is directly proportional to the level of detail required. The time required to run simulation models on a computer is also directly related to the level of detail and this makes detailed simulation slow and expensive. However, the spectacular drop in computer

hardware costs over the last few years has made this much less of a problem than it used to be.

Simulation is used when analytical solution is too complicated and experimental implementation too costly. Simulation modelling can be relatively easy and an extensive mathematical background is not essential to develop a reasonably accurate network model. The usefulness of any model lies in the accuracy of its predictions. This accuracy can be enhanced by performing extensive experiments on the model in order to validate the results.

Several theoretical studies of networks using simulations have been carried out in recent years. In particular, these studies have been used to compare the performance of various local networks [Blair 82]. A few general purpose simulation software packages are commercially available which can provide useful performance statistics about certain networks.

2.2.1.3 Hybrid Modelling

The hybrid model is a compromise between analytical and simulation models, and thus offers the flexibility and speed of simulation models with the accuracy available from analytical models. Information employed by the hybrid models may be detailed empirical data when available, or statistical approximations when they are not available.

Also, this approach is attractive if the frequency of the state transitions of some portions of the modelled system is much higher than those of the other portions. Then the high frequency events are accounted for in a computationally efficient analytical submodel while the relatively infrequent and more complex events are accurately simulated.

2.3 Measurements

The main objective of making measurements on a network is to gather statistics on various events, interpret them in terms of network performance, and tune the network components to optimise performance.

Over the years, a number of measurement methods have been developed to investigate various network performance issues. Early communication networks were well equipped with measurement and monitoring facilities. This section presents case studies of existing measurement methods in important experimental networks which have used measurements as a major performance analysis tool. Three major experimental networks (ARPANET [Tobagi 78], SATNET [Jacobs 78], UNIVERSE [Burren 89a]) have been selected and the measurement tools incorporated in them reviewed. Some measurement tools are common in each network and essentially perform similar functions. In order to avoid repetition, the details of some common tools have been intentionally omitted.

2.3.1 ARPANET Measurement Tools

Some 20 years ago, the Advanced Research Projects Agency (ARPA) set up a 4-node experimental network for computer communications. The network has since evolved covering a number of countries providing mainly computing applications such as electronic mail, remote terminal sessions, file transfers, etc.

ARPANET comprises two main parts : a subnetwork which consists of dedicated processing computers called Interface Message Processors (IMP) interlinked by digital communication lines and hosts which run application programs. The subnet provides the transport media to carry messages from host to host in the form of packets. The traffic originated from the hosts is carried via the IMP's and communication lines in a store-and-forward fashion. The subnetwork possesses multiple routing facilities between some hosts. The performance issues concerning ARPANET are primarily related to the traffic switching performance of the IMP's and effectiveness of the routing and communication protocol between the IMP's.

A number of performance measurement tools were developed to investigate various performance aspects of the network. A Network Control Center (NCC) was used to control and coordinate these measurement tools.

2.3.1.1 Cumulative Statistics

Cumulative statistics consist of data regarding a variety of events gathered over a given period of time. The statistics are in the form of sums, frequencies, and

histograms. Each IMP is equipped with this statistics collection package and records events such as the size of messages entering and exiting the IMP, the number of control messages, etc. The statistics collected by the IMP's are periodically sent to the NCC for off-line analysis.

These statistics are used to determine the user traffic characteristics. Such a measurement facility is useful to the network manager since the measurement results would also indicate the utilisation of the network resources i.e. whether the resources are under-utilised or over-utilised. This facility is particularly useful in operational networks where the network managers need such statistics to plan for network growth.

2.3.1.2 Snapshot Statistics

Snapshot statistics are the instantaneous status of the internal working of the IMPs. The statistics normally include the length of the internal traffic queues and buffer allocations. Each IMP of ARPANET contains a parameter table whose contents indicate the type of statistics to be collected and the rate at which the statistics are to be sent to the experimenter. If the parameter for routing table statistics is set in the parameter table, the IMP periodically sends a snapshot of the routing table. The routing table is modified dynamically, enabling the traffic to be routed to its destination along the shortest path. The routing table statistics provide the means to investigate the performance of the dynamic routing algorithms.

The snapshot statistics, along with cumulative statistics, are very useful for the network operator to determine how congested the IMPs are, and the queue length statistics indicate the traffic volumes passing through various parts of the network.

2.3.1.3 Trace Statistics

The trace mechanism in ARPANET allows the progress of selected packets to be monitored as they traverse through the network. Selected IMP's whose trace parameter is set (enabled) send statistical data (collected within the IMP) to the experimenter when they detect that a tracing packet has been received. The statistical data are put into a new packet called a reporting packet which contains information such as timestamp information relating to the time of arrival of the traced packet, the time of transmission of the traced packet, and the time that the acknowledgment of the

traced packet was received from the next IMP. There is one bit in the traced packet which if set, indicates to the IMP to generate a reporting packet.

The tracing facility is extremely useful in determining the internal queue lengths of the IMP and in investigating the performance of the dynamic routing algorithm. However, this technique has some serious drawbacks. There is a lot of overhead in the IMP in collecting the statistical data, forming a reporting packet, and sending the packet to the experimenter. Furthermore, the tracing feature is very difficult to manage. To trace a packet travelling across a network, every node that the packet may pass has to be enabled for tracing. In networks with dynamic routing, all the nodes have to be enabled since the path taken by the traced packet would be unpredictable. In a large network like ARPANET, this is not a simple task.

2.3.1.4 Message Generator

In ARPANET, each IMP is equipped with a simple traffic generation facility. The parameter table contains a flag, which when enabled, causes the IMP to generate traffic to a pre-specified destination. The traffic generation rate is also specified in the parameter table. The message generator would keep generating traffic at the given rate until the flag is disabled or until some specified time has elapsed.

This facility is very useful for experimental networks because it helps in debugging the network components at the development stage. Furthermore, the use of this facility enables protocol efficiency to be investigated.

2.3.1.5 User-oriented Performance Measurements

In ARPANET, a prototype data communication performance measurement system was implemented to assess the performance provided to a pair of ARPANET end users [Seitz 83]. A pair of network end users (host computers) were selected with specially designed application programs. A specific program was written for the source host which performs all end user activities and records the nature and time of occurrences of each interface signal. A similar companion program performed the corresponding functions at the destination host. A number of user oriented performance parameters were determined to characterise the data communication service provided by ARPANET to its host computers. The performance parameters include network

access time, block transfer times, block error or block duplication probability, block loss probability, etc.

2.3.2 SATNET Measurement Tools

SATNET was a demand access broadcast satellite network which used the INTELSAT IV-A communication satellite. For the purpose of SATNET experimentation, a 64Kbit/sec channel was permanently leased. The physical configuration of SATNET consisted of four earth stations, associated with each station was a Satellite Interface Message Processor (SIMP). SATNET and ARPANET were interlinked through gateways. In fact, SATNET had no independent hosts of its own so these gateways were also used as SATNET hosts. Further details about the network can be found in [Jacobs 78].

SATNET used a powerful measurement tool called 'Timestamping' which was utilised in a number of ways to investigate the SATNET performance [Treadwell 80]. The timestamping measurement approach was later followed by a number of other operational and experimental networks [Becker 85].

2.3.2.1 Timestamping

Selected packets pass through the network from one station to another. These packets contain fixed storage areas into which statistical values can be placed as the packets traverse the network. The mechanism that inserts the values into the packets is called a timestamp station. These stations are situated at convenient points in the network. The timestamp packets whilst travelling through the network, pick up timestamps along their path. The time that they reach each station will be recorded, as well as any other information that the timestamp stations are programmed to place into the packets. Normally, the collected data will be the timing information and for this reason the technique is called the timestamping technique.

In SATNET, this technique was implemented by placing timestamp stations in the gateways and the SIMPs. The gateways were also programmed to generate traffic which acted as timestamp packets. The timestamp packets have two components : a timestamp 'trigger' causing the packet to be detected as a timestamp packet and a 'fixed area' of the packet into which the timestamp could be placed. The timestamp

stations put the timestamp data in the fixed area if a trigger was detected in the transit packets. The clocks in the timestamp stations were synchronised using a 'round-trip' synchronisation technique [Treadwell 80]. One unit of the timestamp data was equivalent to 10.24 milliseconds. When timestamp packets arrive at the destination (peer gateway), the packets had a complete history of their journey. The sum of all the timestamps would provide the end-to-end delay between two gateways and individual timestamps would indicate how much time was spent in each section of the network.

Although timestamping is a tremendously powerful tool for performance analysis, it has its drawbacks too. Firstly, it puts overheads on the network components when incorporating the timestamp stations. This affects the component's normal operation. Secondly, there is a clock drift problem in using the round-trip method of clock synchronisation. Experiments have shown that gateway clocks could drift apart by as much as 30 milliseconds over the course of 60 seconds. Such a synchronisation error may be unnoticeable in a network like SATNET, but could become significant if a network does not involve a satellite. Furthermore, if a network involves a number of nodes (SATNET has very few nodes), there would be a need for a number of timestamp stations and synchronisation of these nodes can become very difficult.

2.3.2.2 Node Emulation

In experimental networks such as SATNET, the number of host machines can be limited for initial experiments. This situation puts constraints on the experimenter in his attempt to understand the network behaviour in future and more realistic environments. The requirement therefore is that a single station should be able to emulate the traffic that could be generated from several sources. SATNET was faced with the problem of a limited number of host machines thus not allowing any meaningful experiments to be conducted. The experimental capabilities were extended by implementing in each host station 10 'fake' stations equipped with all the necessary protocols to allow their independent operation.

2.3.3 UNIVERSE Measurement Tools

Project UNIVERSE was a collaborative multi-site research project conducting research into the construction and use of a high bandwidth communication network. In this

Project, a number of Cambridge Ring local area networks were connected using a broadcast satellite channel, which provided 1 Mbits/sec of bandwidth to be shared between all the sites. The network was used as a vehicle for integrated services research, supporting distributed computing experiments, image and document transfer and packet voice links. The main objective of the UNIVERSE project was to develop a high speed wide area network which would have many of the properties generally associated with LANs.

A number of monitoring experiments were conducted and an architecture for network monitoring was produced. Due to the star-shaped architecture of the network, a distributed management and monitoring approach was adopted. Each UNIVERSE site was equipped with monitoring and diagnostic facilities to collect network data and sending them to a collection site for storage and analysis. The monitoring tools which were developed and used in the UNIVERSE network are described here.

2.3.3.1 Site Logging

A site log service provides an event logging facility to various network objects for reporting significant events. Access for writing to the monitoring data storage facility was provided by the bootserver [Wilber 84] at each site. The received data were timestamped with the date and time and filed in the current day's log; depending upon their nature and urgency, they might also be displayed on the system console screen. A timestamping facility was available on another machine and could be accessed through the network. A number of network components used the site log service : the table server reports the progress of nametable reloading, the nameserver reports various bridge error conditions, the bootserver reports on machines being bootstrapped.

The logging service had more applicabilities than just to collect the monitoring statistics. Most of the experimental network components used directly connected consoles to display regular operational and diagnostic information. It was observed that this led rapidly to the need for a large number of consoles and space to house them. In any operational network, greater use of the site log by network components would become mandatory.

2.3.3.2 Bridge Status and Traffic Statistics

In order to display status information of the satellite segment of the network, the satellite bridges used the monitor-broadcast channel provided within the bridges. A supervisory service monitored the status of the network and broadcast the information regularly, sending an item every 10 seconds. Satellite schedules and messages were also broadcast. A program was written for the BBC microcomputer which could be loaded when required to display each item of information pictorially as it was received. The attraction of this scheme for delivering operational information is that it is available to any display station situated anywhere on the network.

Most of the other UNIVERSE bridges, although provided with consoles to display operational information, were also used to record traffic statistics. The collected statistics included histograms of block counts for various block sizes, total number of blocks and bytes passing through the bridge, number of call setups, number of blocks dropped, etc. The mechanism for reporting these statistics was that, after start up, each bridge would send statistics to a collection point at a pre-defined rate. A monitoring controller could alter the rate.

2.3.3.3 Probes and Reflectors

A traditional tool for network verification and diagnosis is a set of strategically placed 'reflectors' which can be used to return the probe packets. Through these returned packets, the state of connectivity or, alternately, the points of failure of the network can be established. In UNIVERSE, no special purpose reflectors were built into the network, but all bridges and nameservers had the code-identification service which would respond to any request transaction by sending a reply transaction to the originator. The reply would contain the name and version of the code being executed. Using this as a reflector service, it was possible by probing successive bridges and nameservers, to establish the state of a specific path in the network. Information from the probes was processed periodically to produce reports useful for recording network status.

2.3.3.4 Traffic Monitor

For LANs, an ideal diagnostic tool would be one which can record statistics of packets passing between a specified pair of network nodes. Such a tool is also potentially useful for measuring the overall traffic on the LAN; it is often referred to as a 'promiscuous node'[Temple 84]. Although such a promiscuous node can observe all traffic on a token ring or ethernet LAN, the problem is more difficult in the Cambridge Ring LAN due to the fragmentation of packets (Blocks) into minipackets, each with their own flow control. In order to record fully the packets flowing from node A to node B, a promiscuous node must be placed after B and before A so that the response bits can be interpreted in order to reconstruct the basic block.

A monitor station was developed at Cambridge [Hopper 86] which can monitor Cambridge Ring traffic at the lowest level. It was implemented using dedicated hardware which was essential to monitor traffic at the minipacket level. The time between the arrival of consecutive minipackets is less than 4 microseconds and operations which would have to be performed on the minipacket would take much longer if a conventional computer using software was employed. The monitor was used in various configurations to record various types of traffic statistics. For instance, it was used to provide counts of minipacket transmissions between given pairs of nodes. Also, it was used to determine the utilisation of the ring, that is, the proportion of the available bandwidth in use over a given period of time.

2.3.4 Other Measurement Tools

In addition to the measurement tools described in the above sections, there have been attempts to develop measurement facilities for other local and wide area networks. This section briefly review these measurement tools.

The National Bureau of Standards (NBS) developed a LAN measurement centre for NBSNET [Amer 82], which is a distributed, broadcast, local area network. NBSNET is an operational as well as experimental network. Three measurement tools were developed for the measurement centre : an artificial traffic generator, a monitoring system, and data analysis software. The traffic generator is used to load the network to allow for controlled experimentation. The monitoring system records the traffic

information for both artificial and normal network traffic. Analysis software is used to prepare statistical reports from the collected measurement information. Due to the broadcast nature of NBSNET, a single monitor station acting in a 'promiscuous' mode is used to monitor all the packets on the network. Some desirable information cannot be monitored centrally using such a promiscuous station. For instance, transit delays for the data packets cannot be monitored since such information can only be recorded at the user interfaces. To overcome this problem, NBS used artificial traffic generators, and real time clocks were made available to the traffic generators to record timing information. One of the main disadvantages of adopting such a measurement methodology is the coordination of the measurement information recorded by the traffic generators and the traffic monitor. This is because NBSNET is also used as an operational network and the service traffic (not originating from the traffic generators) would affect the results recorded on the traffic monitor. Careful planning is required to minimise such a problem. The measurement centre has been used to prepare two types of measurement reports; traffic characterisation reports and performance characterisation reports. Traffic characterisation reports indicate the workload placed on the network. Such information is a primary source for functional testing of the network. Performance characterisation reports indicate the packet delays, utilisation, etc., which result from a given loading condition. These reports describe the dependent variables which are observed rather than controlled, and are useful for performance comparisons.

2.4 Selecting a Measurement Approach

The measurement method (or methods) chosen for any network will depend on a number of factors. The primary considerations include:

(1) Characteristics related to the network under study : An important consideration in selecting a measurement method is to study the network on which it is to be implemented. This would highlight the performance characteristics which are important to a particular network. Generally, networks can be categorised into two types : the experimental networks which are developed to test novel concepts, and the fully operational networks which are based on proven concepts and which provide a communication facility to a group of people. In the case of operational networks, the measurement methods are designed or selected to yield statistics related to network reliability and availability, network utilisation, user traffic characteristics for the

purpose of network accounting, etc. Such statistics are extremely helpful for the network operation. Also such statistics determine which part of the network is under-utilised, unreliable or unavailable most of the time, and which parts require more resources to overcome problems like congestion and overloading of network components. In experimental networks, the chosen measurement methods should address the performance issues such as protocol performance verification, performance of the packet switches in dynamic and mixed traffic environments, code testing and debugging of various network components, etc. With the help of measurements, investigation of such issues would enable improvements to be made to various network components. Furthermore, this would help to endorse newly developed concepts.

(2) Measurement Orientation : Although there is an overlap in terms of the benefits to be gained form the measurement methods, each is biased towards certain group of people. There are three main groups:

The Manager

The Designer

The User

Traditionally, there has been more emphasis on measurement methods which provide operational statistics for network managers. The majority of measurement tools found in existing and past networks indicate that their benefits are oriented more towards the network managers. For instance, cumulative statistics, snapshot statistics, site logging, etc., are mainly employed for the benefit of network managers. However, particularly in operational networks, the provision of such tools is essential for producing gross network performance for the network managers. Furthermore, the network picture produced from such measurements provide an operational guide for the network operators.

There are a few measurement tools which are beneficial to the network designers. For instance, trace statistics, node emulation, and message generator have been used to help the network designers in identifying practical performance problems. The results gathered from these tools can be used to verify the designer goals and can even be used to improve the performance of network components. The designer oriented

measurement tools are especially important in an experimental network; the designer has the liberty to change any performance related parameter if that could improve the overall network performance. Such changes are more difficult and expensive to make in an operational network than in an experimental network.

There are very few measurement tools which can be helpful in determining the network performance that a user would experience at the network user interface. Unfortunately, early wide area networks were not well equipped with user oriented measurement tools. In ARPANET, some experiments have been conducted for the benefit of the network users (as described in the earlier section). In integrated services networks, the provision of such measurement tools could play a very important role.

(3) Measurement Accuracy : Another factor which affects the selection of a measurement method is the measurement accuracy. Although all measurement methods share a common characteristic in that the measurement results can never be completely accurate, the relative accuracy is important in selecting a measurement method. For instance, in order to measure delay across the network, a number of measurement methods exist : the round-trip delay measurement method, timestamping, etc. Each method inherits its own inaccuracy problem.

2.5 Proposed Measurement Methodology

Different measurement techniques have been described in this chapter. None of these fulfill all the ideal requirements of a measurement strategy, each having its own strengths and weaknesses.

Unlike analytical or simulation techniques, performance measurements require that there be a physical network on which the measurement strategy can be implemented. Thus measurements are unavoidably tied to a particular real network. In developing the measurement methodology, the Unison network was considered to be the target network and initial considerations were given to the requirements of the Unison scheme. However, considerations were also given to extend the scope of the strategy developed to other similar networks.

The following requirements were considered important in developing the measurement strategy:

1. The measurement strategy should be able to determine the network performance at the network user interface for a variety of applications. In other words, the measurements as seen by the 'user' should be as accurate as possible over real network routes.

2. It should be able to measure the one-way packet transit delay accurately between remote users.

3. It should be able to characterise the operational performance of network bridging components under various loading conditions. In other words, the ability to load or stimulate the network bridges with simulated user traffic, and simultaneously monitor the perceived performance.

Consideration was also given to features which were not essential. These include:

1. The ability to monitor all network traffic simultaneously

2. The ability to monitor traffic at the lowest level of network abstraction.

These requirements allow immediate dismissal of certain approaches. These include simulation, as the results will only be as correct as the models adopted to represent network bridging components. Also rejected are passive measurements within the packet switches, as this would not yield user perceived network performance statistics but would generate an excessive amount of unwanted data. Furthermore, performance recording in the packet switches have some adverse impact on the normal switch operation and disturbance to the switch operation is proportional to the complexities of the embedded measurement procedures within the switches. In this thesis, the aim is to avoid the use of packet switches for statistics collection and to explore other means of obtaining measurements.

An approach based on independent traffic generators and receivers was adopted, in which the generators and receivers usually appear to the network as application 'clients'. By a combination of software and hardware control, these can generate, receive and analyse simulated user traffic either in isolation or in parallel with real application traffic. An interface to the Rugby MSF radio time clock was provided to enable accurate (1 millisecond) synchronisation at remote sites, providing the ability to perform one-way packet transit delay measurements. This is referred to as chronological timestamping [Siddiqui 89].

A traffic generator and receiver usually (but not always) operate as a pair, and can be placed anywhere where a user interface resides. By starting tests using the pair on the same client network, and steadily moving them apart by increasing amounts of network infrastructure, it has been possible to build up a picture of the traffic performance of different network components in isolation, as well as in sequence. The performance parameters such as maximum throughput, delay characteristics, and packet loss statistics can be determined by conducting tests over various network routes.

The performance of the packet switches can be better studied in a controlled traffic environment rather than a real traffic environment. In a controlled traffic environment, a known traffic load with a known traffic pattern can be exerted on the switch, thus enabling the exact performance characteristics to be investigated. Furthermore, the traffic generators can produce repeatable traffic patterns. By adjusting various parameters in the traffic switching algorithm of the switches, the repeated measurement experiments can help to investigate the comparative performance improvements. Such a measurement strategy can thus be used in a 'measure and tune' fashion to improve the overall network performance.

2.6 Discussion

An important consideration in developing a measurement strategy is to study the fundamental principles behind the network to be investigated. For instance, experimental networks are developed with practically non-proven concepts. The network would normally incorporate state-of-the-art technology and its designer would periodically require up-to-date and accurate performance information which can be used to validate the theoretical results. The performance information can even be used to optimise the network performance if required. Similarly, the application designer, not sure about the exact network behaviour, would like to know about the network capabilities before putting the application traffic on the network. The application designer normally relies on the network designer for such information and the network designer in turn normally relies on results obtained from simulation or analytical models, unless the network is equipped with effective measurement tools. The measurement methodology for any network should address the performance issues which are important in the context of the network under consideration.

Managerial measurement methods are of general use to the network users but each user may wish to know how well the network will handle his own specific application. Although results obtained from managerial measurement methods provide a measure of expected network throughput and delay for a good mix of traffic, details will vary depending upon the characteristics of the traffic, the frequency with which it is inserted into the network and other traffic on the network. Sadly, very few networks are equipped to measure even an individual user traffic stream. Installing measurement facilities in a network to monitor traffic streams is expensive.

In experimental networks, the normal user traffic is typically low and rarely stresses the network under full capacity. Incorporation of other traffic sources which can be used to study network behaviour in various loading conditions is extremely useful. In particular, this would help to pinpoint weak links within the internet.

Chapter 3

Measurement System Design

3.1 Introduction

This chapter describes the design of the measurement system and discusses various issues which may arise when implementing the system on the network.

The measurement system is based on a number of Traffic Generators and Receivers (TGRs) dispersed across the network. These are capable of generating and receiving traffic with various traffic patterns. The generated traffic can be injected and extracted at pre-defined network locations enabling performance characteristics of the network section under study to be investigated. By varying the traffic loading and traffic pattern, it is possible to determine experimentally the operating characteristics of the critical network components. A unique controller station connected to the network enables traffic parameters between the TGRs to be controlled. A data analysis machine is incorporated into the system. This machine collects measurement data for off-line analysis.

In order to measure the one-way packet delay, the problem associated with achieving such a measurement must be understood. This is that the processors of the traffic TGRs require their clocks to be synchronised with each other. This problem has been resolved by using a broadcast synchronisation technique in which a broadcast station

transmits a regular time signal to its clients. Each TGR is equipped with a time clock receiver which receives the time signal and continuously updates and re-synchronises the generator's clock, thus keeping them in synchronism with each other. The TGRs timestamp the generated traffic with chronologically ordered data thus enabling the measurement facility to determine the one-way packet delay. Since the packets are timestamped in chronological order, the technique has been referred to as Chronological Timestamping.

It is believed that this is the first time that such a measurement system based on TGRs along with the chronological timestamping feature has been extensively studied and used as a sole measurement tool for the performance analysis of a large experimental network.

3.2 Main Features of the Measurement System

There are five main features which describe the complete measurement system:

- TGR machine
- Traffic generating functions
- Chronological timestamping
- Network segmentation
- Control and statistics collection

It is the interplay of these five features which constitute a complete measurement system. The TGR machine is a piece of equipment which provides the necessary hardware for traffic generation. The traffic generating functions are a collection of software algorithms which reside in each TGR. These are responsible for producing various traffic patterns. The chronological timestamping facility enables the one-way packet delay across the network to be measured. Network segmentation allows a network to be divided into small chains such that any number of components can be selected in a chain for performance experiments. Control and statistics collection enables traffic parameters between TGRs to be controlled and collects measurement data for off-line analysis.

3.3 The TGR Machines

These machines can either be based on stand alone devices or be incorporated within the packet switching components whose performance characteristics are to be investigated. The decision would mainly depend on the functionality to be achieved from the measurement system. Each one has its merits and disadvantages.

3.3.1 Within the Switching Components

If the traffic generation facility is embedded within a switching device, it will introduce adverse characteristics, e.g., interference with the normal operation of the switch. This is because a portion of a component's resources would need to be allocated to perform the traffic generation operation. Another important consideration is the functionality. If a number of complex traffic generating functions are required, a large portion of the component's resources has to allocated to achieve this and as a result, the performance characteristics of the switching components will not reflect the true behavior unless extra resources are provided in the component. Also it would be difficult to determine the performance parameters between the user-ends unless a user protocol has been implemented at the switching components. Furthermore, such a mechanism will not be flexible in terms of control and data collection.

From an implementation point of view, it is easier and cheaper to incorporate such a facility within the switching components. Usually it is included within the component when a point-to-point link protocol between nodes is to be investigated.

This strategy has been adopted in the ARPANET and SATNET (as described in Chapter 2) , where the software of the packet switches was modified to generate traffic. These switches were used to generate simple traffic streams which were used for loading and protocol verification purposes.

3.3.2 Stand alone machines

A stand alone device has the advantage of being able to report on the performance characteristics of network components as seen by the network users. Complex traffic generating algorithms can be implemented to enable more of the network functionalities to be tested. However, such implementation would involve extra

hardware and network interfacing equipment. This problem can be resolved by slightly modifying existing user equipment to act as TGRs. This technique had been adopted in NBSNET [Amer 82] where a number of user boards were modified to generate artificial traffic on NBSNET for conducting measurement experiments. A similar approach has been adopted in designing the measurement system in which apart from basic measurement equipment, the application components are programmed to act as TGRs. This will reduce the extra cost involved with such a measurement methodology.

Other than using user boards for generating traffic, there are other network components which can be used as TGRs. Some machines on the network are used occasionally. For instance, in Project Universe, bootserver machines were employed at each site and were used intermittently for booting various network components [Wilber 84]. The functionality of such machines can be extended with very little additional hardware and software to act as TGR for each site. The multiple use of such machines would not upset the network's normal operation as would be the case if packet switches are used to serve the same purpose.

3.4 Traffic Generating Functions

The use of traffic generators has normally been limited to verifying network protocols since these can be better studied with controlled traffic rather than with real user traffic. The functionality of traffic generators can be extended by making them more versatile in terms of the traffic pattern they can generate and the application traffic they can emulate. This would enable more of the network characteristics to be observed from various perspectives. The traffic generating functions can be classified into three main classes:

- Network loading function
- Application emulation function
- Traffic accounting function

These functions can either reside in the TGR (depending upon the resource capacity of the TGR) or in the controller station.

3.4.1 Network Loading Function

There are three types of traffic generating functions which can be used for network loading : deterministic traffic generating function, exponential traffic generating function and burst traffic generating function.

In the deterministic generating function, the traffic is generated with a constant packet rate. The rate can be varied by varying the delay between successive packets. Other traffic parameters which can be varied are the packet size and the traffic type (priority/non-priority). The maximum rate should be high enough to overload any of the network bridging components. This rate would mainly depend on the speed of network interfaces.

The traffic generated by the exponential generating function follows an exponential distribution. The traffic related parameters are similar to the deterministic function except that the inter-packet delay is now exponentially distributed and the mean inter-packet delay can be selected to increase or decrease the traffic intensity.

The burst generating function periodically generates a burst of packets. The traffic related parameters include packet size, traffic type, burst size (number of packets in each burst), inter packet delay (delay between the emission of two consecutive packets in a burst), inter burst delay (delay between the emission of two consecutive bursts).

3.4.2 Application Emulation Functions

As the name suggests, these functions are capable of emulating a variety of user application traffic patterns. The applications include voice, video and computer data. Each function allows the various traffic parameters concerning the emulated function to be input. By injecting emulated traffic into the network, user-oriented performance parameters of various applications can be observed. For instance, in real time voice communication, the distribution of packet transit time, distribution of packet inter-arrival time and packet discard rate are important user-oriented parameters.

The voice traffic pattern is emulated either with silence supressed or without silence supressed. In the former case, a number of silence suppression algorithms exist which

can be employed [Gruber 82]. Non-compressed voice traffic can be easily generated by producing a constant packet rate such that the effective data rate is 64 kbits/sec.

Video traffic generation can either be constant block rate (which is easy to emulate) or variable block rate. The variable block rate source can be easily modeled by storing an abridged set of results from a real coder and scaling these according to the load required [Chin 89]. The computer traffic exhibits bursty behavior and can be easily emulated by the burst generating function as explained in the previous section.

3.4.3 Traffic Accounting Functions

These functions are responsible for the preparation of histograms of the requested performance parameters. These functions reside either in the TGR machine or statistics gathering machine attached to the TGR. The parameters required in the histogram preparation (such as low and high ends of histogram cell range, cell increment, etc.) can be passed from the controller station before the start of the test. These histogram based statistics are then used to prepare various statistical distributions which facilitate the interpretation of performance results.

3.5 Chronological Timestamping and Delay Measurement

The traffic generation measurement technique, although conceptually simple, is not straightforward to implement. For instance, in order to measure the one-way delay, the processor of each TGR requires clock synchronisation. Clock synchronisation is difficult to achieve and accurate one-way delay measurements in general cannot be carried out in a wide area network. There are other delay measurement techniques which do not require synchronisation. There are three general delay measurement methods:

- Round-trip delay measurement
- Pickup delay measurement
- Absolute delay measurement

3.5.1 Round-trip delay measurement

This is the simplest and the most commonly used technique employed in wide area networks [Burren 89a]. The delay measurement between two distant machines is made by sending a packet containing a local clock value from the sender. When the packet arrives at the receiver after traversing through the network, it is immediately returned to the sender. When this packet arrives back at the sender, the sender determines the round-trip delay by subtracting the clock value in the packet from the current clock value. This gives the combined outward and return time relative to the sender's clock. The round-trip measurement divided by two provides the one-way delay. However, this assumes that the network characteristics are symmetrical in both directions, i.e. outwards time equals return time. Often this is not the case. It is likely that the delay in one direction is low (because of no traffic loading), while the delay in the other direction is high (because of excessive loading). Although it is tempting to use such a simple technique, there is in fact a trade-off between accurate estimation of one-way delay and the complexities of the delay measurement technique.

3.5.2 Pickup delay measurement

This method involves network switching components in determining the end-to-end packet delay. The idea is to measure packet delay where it actually occurs. The packet switches are the source of packet delay because of processing and queuing within the switches.

In this method, certain packets, called pickup packets [Treadwell 80], contain 'storage areas'. These packets are used to pick up timing information injected by the switching components as they travel through the network. The packet switches use the storage areas for timestamps indicating the amount of time spent within the switch. The delay calculation can be made by each network switch, using only its local clock. For example, when a packet enters a packet switch, it is tagged with the current value of the local clock. The packet then experiences variable delay in processing and queuing. As the packet is about to leave the switch, the current clock value minus the previous tagged value is used to stamp the packet as the amount of time spent in the switch. Since only one clock is being used by the switch for delay calculation, frequency drift in the clock will not introduce any measurement error. Fig. 3.1 shows the delay

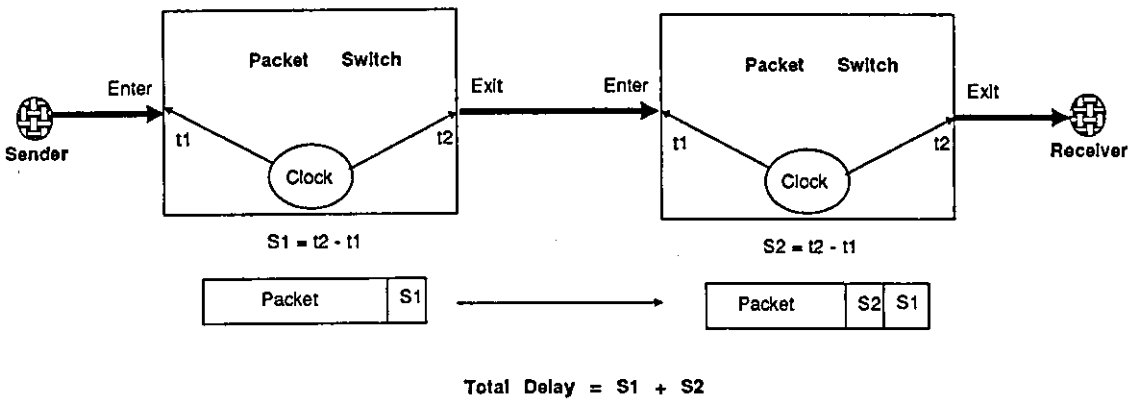


Fig. 3.1 Illustration of pickup delay measurement technique

measurement mechanism as a pickup packet traverses a network. The end-to-end delay can be calculated by adding the individual timestamps and the propagation delays on the transmission links.

Although the pickup delay measurement technique produces more accurate results than the round-trip delay measurement, it has its problems too. For instance, the technique involves packet switches for timestamping; this will not only put extra processing overhead on the switches but also another protocol layer may need to be defined to overcome packet checksum problems. Packet checksuming in packet switched networks is calculated between packet switches (hop-by-hop basis) and also between the user-ends (end-to-end basis), since timestamp data is inserted at the switches, some extra measures have to be taken to overcome the checksum updating problem. There are other limitations which have been discussed in detail in Ref. [Treadwell 80].

3.5.3 Absolute Delay Measurement

In this method, the clocks between distant machines are synchronised with respect to an absolute reference time. The transmitter timestamps the packets with the absolute timing value just before launching them onto the network. At the receiver, the remote timestamp is compared with the current local clock value. Since both the transmitter

and receiver clocks are synchronised to the same absolute time reference, the difference of the clocks gives the amount of time spent in the network. Here the measurement technique looks quite simple, the problem is one of maintaining synchronised clocks.

Maintaining synchronised clocks in a geographically distributed system is, in general, a complex problem. What is usually required is the distribution of a standard reference frequency to each local clock via a reliable channel (usually dedicated communication lines) to obtain frequency synchronisation. Time synchronisation is achieved by distributing the absolute time reading of a master clock to the local clocks over communication channels with known propagation delay. This process is used to maintain synchronisation in digital telecommunication networks. This sort of synchronisation strategy can prove very costly for the sole purpose of packet delay measurement in a packet switched network. There are other synchronisation techniques which are discussed in the next section.

3.5.4 Chronological timestamping

In carrying out delay measurement experiments using the TGR, the main aim is to adopt a synchronisation technique which is cheap and easy to implement, and at the same time provides enough accuracy in delay measurements across the network. Another consideration is the extendability of the synchronisation technique; a number of TGRs scattered over a number of network sites can be synchronised with each other such that delay measurements can be conducted between any two network sites.

Other issues include the format and the resolution of the timestamp data. The timestamp data can either be the global time clock or just the chronologically ordered data from a well-known reference. In the former case, a synchronised global time clock is used for each TGR such that all TGRs contain identical times. The global time can be encoded in a timestamp with sufficient resolution (not to be the whole global time) so that the receiver station can unambiguously determine when the packet was produced. The receiving station can infer un-sent timestamp data by examining the timestamp and its own global time clock.

When chronological ordering data is used for the timestamps, only a synchronised well-known reference is made available to each machine and the timestamp data

would be the chronological ordering data with respect to the well-known reference. The resolution of the chronological ordering data (timestamp data) would depend on the crystal clocks provided in the timestamping hardware.

3.5.4.1 Clock Synchronisation

If two clocks are started at the same time with the same starting values, the two clock values will no longer agree with each other after a certain period of time. The variation in clock values is usually termed clock drift. Crystal clocks found in today's processors run at rates that differ by as much as 1 usec in one second and thus can drift apart by 1 sec every 10 days [Schneider 86].

The accuracy of the performance statistics computed in terms of elapsed time between events depends on how closely the clocks in participating machines can be synchronised. There are several methods of synchronising the clocks in a distributed system.

A simplest method of clock synchronisation can be explained by considering two clocks, one acting as a master and the other as a slave, as illustrated in Fig. 3.2. The slave periodically adjust its clock value by reading the masters clock. If master and slave are residing on a distributed network, then message passing is the only way the slave can read the master's clock. The slave adjusts its clock by doing the following operations.

- The slave sends a packet to read the master's clock and at the same time saves its own clock value, say as S_1;

- When the packet reaches the master's interface, it collects two clock values; one on entry into the master's interface and other on exit from the master's interface, say M_1 and M_2 respectively;

- Finally, when the packet arrives back at the slave, the slave reads its own clock again, say S_2;

- The time spent in the master to turn the packet around will be $(M_2 - M_1)$. The time spent in the network, N_t

$$N_t = (S_2 - S_1) - (M_2 - M_1)$$

N_t divide by 2 gives the one-way network time with the assumption that the packet experiences same time in both directions, the time to get from the slave to the master is equal to the time to get from the master to the slave. If this is true then any adjustment to the slave's clock is calculated as

$$\text{Adjustment} = M_1 - (S_1 + (N_t / 2))$$

This synchronisation technique, usually referred to as the round-trip synchronisation technique, is easy to implement and gives good results with very small synchronisation error. The synchronisation error would depend on how frequently the clocks are synchronised and on the network loading at the time of synchronisation. However, this technique has its limitations too. The uncertainity in determining $N_t/2$ (hence the need to measure it) cannot be entirely eliminated even with the use of the most efficient protocol between the master and the slave. Furthermore, if a network

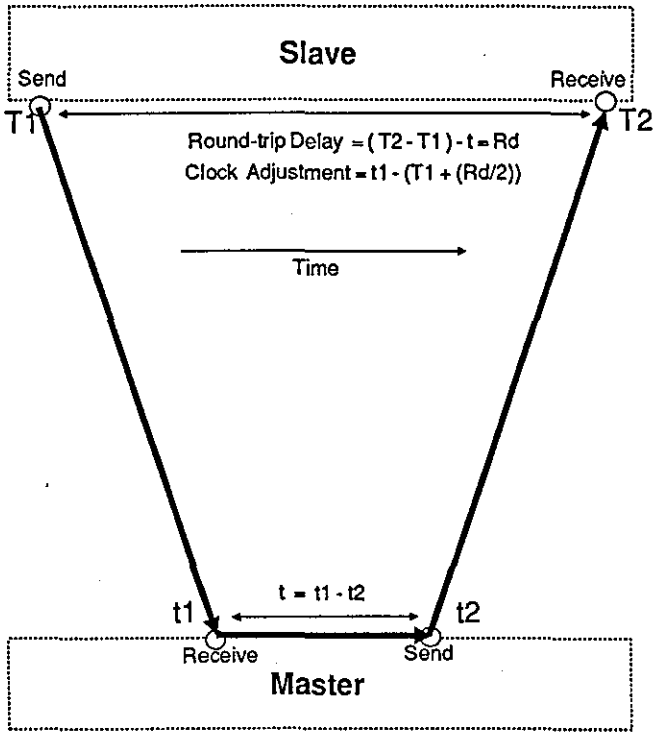


Fig. 3.2 Illustration of round-trip clock synchronisation technique

provides alternative routing between the master and the slave, the synchronisation error may increase as a result.

Synchronisation can also be achieved by virtue of the network architecture. If a network architecture involves broadcast satellite systems, then the clocks provided at the satellite ground stations can be used for synchronisation. Each satellite ground station site possesses an accurate representation of a global time so that it can make transmissions to the satellite that will not conflict with other users. Since these clocks are updated continuously, the problem of clock drift is resolved.

3.5.4.2 Broadcast Synchronisation Technique

This technique involves a reliable and accurate time source and a broadcast station which periodically broadcasts the correct time to its clients. Upon receipt of such broadcast, the clients adjust their clocks according to the correct time provided. The reliable time source serves two functions in the clock synchronisation. Firstly, it periodically generates an event that causes every client to resynchronise their clocks at about the same time. Secondly, it provides a time value that can be used in adjusting the client's local clock. If the broadcast is done frequently enough, the client's clock will not drift too far apart in the interval between the broadcast.

The broadcasting media can be either communication networks or radio/satellite communication. In the case of communication networks, the reliable time source has to make sure that the synchronisation message takes a known and constant time to reach all the clients. This can be easily achieved in a star network. In other types of networks, broadcast synchronisation would be difficult to achieve and other approaches have to be sought.

In the case of radio broadcasting, the synchronisation problem is much simplified since the propagation speed for the synchronisation signals is the speed of light, i.e. 3×10^8 meter/sec. The propagation time over a few tens of kilometers is almost negligible. In most countries, a radio frequency is allocated for the purpose of broadcast synchronisation. In the United States of America, the National Bureau of Standards (NBS) provides an NBS time dissemination service via the National Oceanic and Atmospheric Administration's GOES satellite [Seitz 83]. This service makes it possible to obtain a time signal accurate to within 1 msec anywhere in North America. Several vendors supply clock receiver units to receive the time signals.

In the United Kingdom, the National Physical Laboratory maintains a radio transmitter which provides a continuous source of accurate absolute time. The radio transmitter is located at Rugby, England. The transmitter generates continuous atomic time signals which come from the caesium beam standard at the National Physical Laboratory and are accurate to 1 part in 10^{-13} (300 microseconds in a century). A number of vendors in the UK supply Rugby radio time clock receivers. The receiver receives the time code during each one minute cycle at a rate of 1 bit/sec. The time code is decoded to produce absolute time. Although the whole time code is updated after every minute, a second pulse (which carries the code bit by bit) is used to keep the clock synchronised second by second.

3.5.4.3 Broadcast Synchronisation - Implementation

The broadcast synchronisation technique has been used to synchronise the processor clocks of the TGRs. Rugby time clock receivers have been incorporated in each generator/receiver.

As described in the earlier section, the Rugby clock receiver provides absolute clock values. The question is whether to use the absolute values for the timestamps or just a well-known reference whereby the timestamping data would be the chronologically ordered values from the well-known reference. A well-known reference is defined as a reference which is known to all participating machines that need to be synchronised with each other.

Since the aim of synchronising the clocks is merely to provide a delay measurement facility, the provision of absolute clocks in the measurement system is quite unnecessary. This would have involved extra cost and effort. However, such provision can be justified if the measurement system is also used for other purposes, such as to timestamp the measured statistics with the absolute time values or to provide a time-and-date service on the network. Timestamping of measurement results is extremely important in an operational network because this will help in the accurate analysis of the measurement data with respect to the time of the day and day of the month. This will eventually help in planning the network growth in terms of allocating more network resources to overcome peak load problems. Because of the experimental nature of the network for which the measurement system is to be designed, such a facility is not essential.

If a well-known reference is used for the synchronisation and chronological ordering values are used as timestamp data, then implementation becomes extremely easy. Most commercially available single board computers are equipped with hardware clocks or counters which can be used in conjunction with the well-known reference to make a complete timestamping system. Using such a scheme, a simple delay measurement technique has been developed which is explained below.

3.5.4.4 Timestamping Operation

The clocks mark instances of time with 'ticks' and computers use these ticks as a yardstick for time. The elapsed time between events is then measured by counting the number of ticks. As described earlier, the Rugby clock receiver receives second pulses once every second. These pulses, termed here as Rugby ticks, can be fed to the computer which uses them as a time yardstick.

Consider a simple case in which a pair of test machines : a Traffic Generator (TG) and a Traffic Receiver (TR) are to be synchronised with each other. Both machines are connected to the network and the only way that the machines can communicate with each other is over the network. The Rugby ticks are made available to the processors of both TG and TR. These ticks, although synchronised (happening at the same time), have no values attached to them. In order to attatch certain values, the TG is requested to initiate a 'synchronisation session'. A synchronisation session is a period of time (which can last until the machines are switched off) during which the machines remain synchronised with each other.

The requirement for the machines to be in a synchronisation session is that both processors (in TG and TR) should have identical tick counts. Each Rugby tick is used to increment a 'tick counter' and the contents of both tick counters are updated with identical values. To achieve this, the TG makes the content of its tick counter zero and just after the next Rugby tick, sends a control packet to the TR in order to make the content of the TR tick counter one. The contents of the counters act as a well-known reference between the two machines. Here, it is assumed that a control packet takes less than a second to reach TR. Each succeeding Rugby tick in the TG and TR will increment the contents of the tick counters by one with both counters having identical values.

Once the machines are in synchronisation, they can be used for one-way delay measurement. However, at this stage, the delay measurement error could be up to two seconds. This measurement error may be justified if the packet delay is in minutes or hours (rather than in milliseconds). In order to reduce the measurement error, a 'micro tick counter' is incorporated which provides a tick every microsecond between successive Rugby ticks. A crystal clock driven high frequency counter can be used as a micro tick counter. Each Rugby tick resets the micro tick counter which means that the micro tick counter in TG and TR would remain in synchronism as long as the Rugby ticks are ticking at the same time.

The TG and TR use the contents of both counters (Rugby tick counter and micro tick counter) as the timestamp data for delay measurement. The TG timestamps the packets with both counter readings just before launching them on the network. When the packets arrive at the TR interface, these are timestamped again with the TR counter readings. The difference between the two stamps determines the one-way

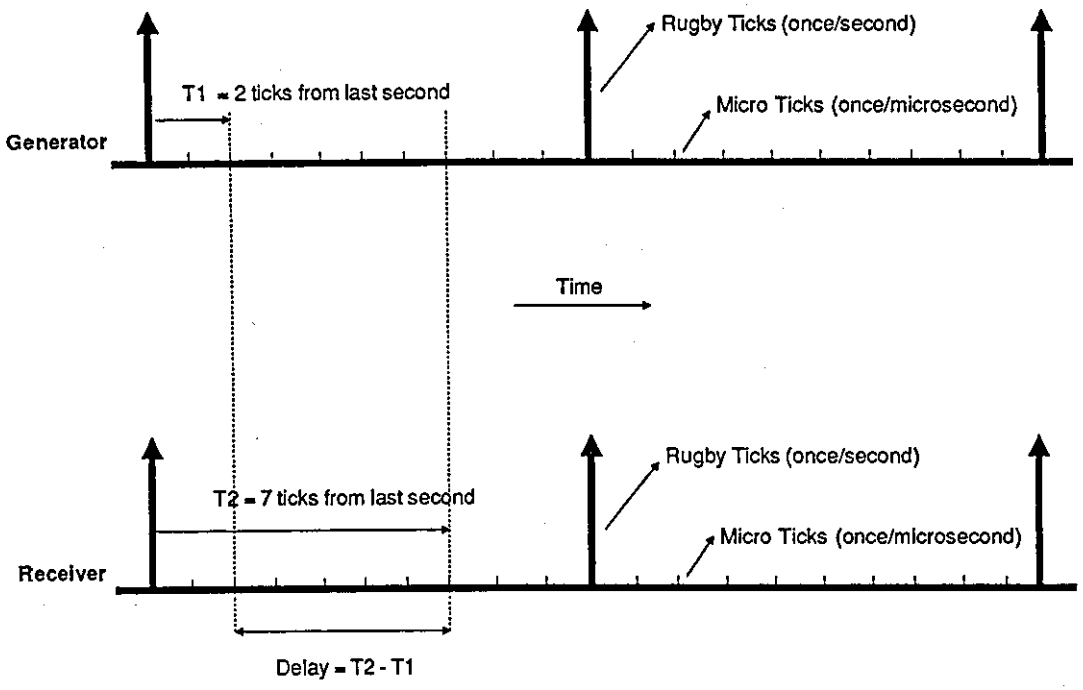


Fig. 3.3 Illustration of delay measurement using chronological timestamping

packet delay. Since the timestamp data in both machines is in chronological order, the

delay measurement technique is referred as Chronological Timestamping technique. Fig. 3.3 illustrates the one-way delay measurement between two stations using the Chronological Timestamping technique.

3.5.4.5 Synchronisation - more than two test machines

The synchronisation procedure is very similar to the one explained for two test machines. If more than two machines are to be synchronised, then one has to be selected as a master and the rest as slaves. In order to initiate a synchronisation session, the master communicates with the slave machines to pass on the well-known reference. The only limitation is that the master has to communicate with all machines within one second. This may limit the number of machines to be synchronised and the maximum network delay for a control packet to reach the farthest slave machine. If there are only a few machines and the network delay is of the order of milliseconds, then the limitation can be overcome by designing an efficient synchronisation protocol between the master and slaves. Once all machines have an identical well-known reference, any pair of machines anywhere on the network can be used for timestamping to measure packet delay across the network.

3.5.4.6 Main Features of Chronological Timestamping

The main features of the chronological timestamping technique are described below:

1. The most important feature of this technique is that the clock drift does not have a cumulative effect over the measurement results. Although the well-known reference between the synchronised machines remains fixed for a sync session, the Rugby ticks continuously make a sub-reference for the micro tick counter (initialising the counter to zero every second). This means that cumulative clock drift is restricted only to that accumulated over one second and is thus negligible.

2. The technique is easy and cheap to implement. The timestamping process only involves a very small software overhead.

3. Although this technique is oriented towards delay measurements, it can be easily used for other purposes, such as packet voice synchronisation [Montgomery 83].

3.6 Network Segmentation

An important requirement for conducting performance experiments in a wide area network is the provision of a network segmentation facility which can be used to divide the network into defined segments (from a traffic path point of view). This would enable one to select any number of network components in a segment in order to investigate the performance characteristics of the components involved in that segment. The bridging components in an integrated services network are complex machines and analysis of such machines is a non-trivial matter, especially when a number of these machines are involved in a communication chain. The use of a segmentation facility would help to investigate the relative performance degradation or improvement of adding or removing components into the network segment under study. This would in turn help to pinpoint weak links within the network .

The network segmentation facility plays an important part in the effective use of the measurement system based on TGRs. There are two possible ways through which the network segmentation can be achieved.

- Traffic Echoing
- Physical Placement of the TGR

3.6.1 Traffic Echoing

In this method, a number of traffic reflectors are placed at various places in the network. If a traffic stream is sent to such a traffic reflector, it will be echoed back to the client local network where it was originated. The use of the reflectors would enable the network to be segmented into a number of traffic loops, with each loop containing a pre-defined number of network components. Initially, for example, the loop may only involve the first switching component. This can be extended to involve another switching component, enabling the effect of adding more of the infrastructure to be observed.

The provision of traffic reflectors can easily be made possible using the traffic routing mechanism provided in the network. For instance, in the Unison network, the secretaries (at the exchange and client networks) which form part of the network

management infrastructure for addressing and routing, are also used to configure the network switching components in such a way that traffic can be reflected or returned to the originating local network after traversing a pre-determined proportion of the network.

3.6.2 Physical Placement of the TGR

This scheme of network segmentation would involve the physical connection of the TGRs at the end-points of the network segment under study. The traffic streams can be injected and extracted at the segment end-points. The physical connection of measurement components would require the support of different interface cards between different network sections and measurement components.

In the CFR based Unison network, the traffic generator can be connected to both types of network; the client network or the backbone network. Fortunately, the same interface card enables the TGR to be attached to either type of network. Both kinds of segmentation techniques have been employed on the Unison network for the effective performance analysis of each network component.

3.7 Control and Statistics Collection

In order to control the measurement facilities, and to collect and analyse the measurement data, an important requirement for the measurement system is to possess a flexible and user friendly control and statistics collection mechanism.

The controlling mechanism can be either based on a stand alone machine residing on the network or it can be incorporated within a measurement device. In the latter case, a dumb terminal is usually connected to one of the measurement device which acts as a measurement machine as well as providing the necessary processing power for controlling measurement activities. The dumb terminal just acts as a display and interactive input/output device. Such a mechanism may not be flexible in the sense that measurement activities cannot be controlled from a site which does not possess a measurement machine. If a stand alone device is used for control purposes, it can provide more flexibility. A stand alone device can easily control measurement machines which can be anywhere on the network. A site workstation which is commonly used to control loosely coupled application components is an ideal machine

in which measurement control mechanism can be employed. In either case, the control machine should provide a user friendly interface between an experimenter and the measurement devices.

The statistics collection machine provides measurement data collection and storage facility for off-line analysis. If a controlling machine is a stand alone device, it can also be used for collection and analysis of the measurement data.

3.8 Summary

The measurement system based on TGR provides a powerful tool for performance analysis of experimental networks. The TGR with its chronological timestamping feature can gather network performance information which would be difficult to obtain using other measurement methods. The performance information will be useful to the network designer and user alike.

The selection of machines for generating traffic is an interesting problem. If stand alone machines are employed, the advantage is to be able to achieve more functionality without any disturbance to the normal operation of the network components. In experimental networks, the user equipment can easily be used for the purpose of traffic generation with slight modifications. This reduces the cost and effort which would be required if everything is to be developed from scratch. On the other hand if packet switches are used for such a purpose, this will not only disturb the normal operation of the switch but may require an extra layer of protocol manipulation in order to achieve the required functionality. The traffic generating functions provides a wide choice for generating various traffic patterns. This enables the network performance to be studied from different perspectives.

Accurate packet delay measurement across wide area networks is difficult to obtain due to physical separation of transmitting and receiving stations. The problem is associated with the synchronisation of clocks between stations. This has been resolved by using the broadcast synchronisation technique. The Chronological timestamping technique is introduced which provides the simplest possible delay measurement implementation and at the same time is effective in providing accurate measurement results.

The network segmentation facility allows a large network to be divided into sections so that any number of components can be selected in a section and performance experiments conducted on the network components under study. An important requirement for the measurement system is to possess a flexible mechanism for controlling measurement activities and statistics collection. This can be achieved by having a stand alone machine at a network site for control and collection.

Chapter 4

Measurement System Implementation

4.1 Introduction

This chapter describes the implementation of a measurement system based on the Cambridge Ring (CR) local area network. The CRs were used as the client networks in the earlier phase of the Unison network. In the latter stage of the Unison network, the emphasis was shifted towards using Cambridge Fast Rings (CFRs) as the client networks. The measurement work carried out on the Cambridge Fast Ring based Unison network is the subject of the next two chapters.

Before describing the design and development of the measurement system, it is essential to understand the network on which it is being implemented. This understanding will highlight the characteristics that can be measured and the performance issues which can be investigated. This chapter starts with a detailed introduction to the Unison network. The network components, their functional details and related performance issues are discussed. Various considerations in designing the measurement system are then discussed, for example what is worth monitoring and how effectively it can be monitored. The measurement components are described in terms of hardware and software.

Using the measurement system, a number of performance measurement experiments have been conducted in various network configurations. These experiments have provided a quantitative and qualitative measure of the operational characteristics of the Unison network. Important lessons learned from the measurement results are included towards the end of the chapter.

4.2 Project Unison

Project Unison [Clark 86] a collaborative research project was established to conduct experiments in local and wide area networking techniques for multi-media office applications. The applications include voice, video, graphics and conventional computing services. The main objective of the Unison Project was to develop an experimental network test bed and application components in order to test the suitability of the network for integrated services.

The Unison network architecture has been based in part on the experience gained during the earlier Universe Project [Burren 89a]. This project used satellite technology to interlink geographically dispersed local area networks. The Universe network was based on the use of CR LANs at the various sites. Much of this Universe infrastructure was inherited by Project Unison, and initial efforts were directed towards the interconnection of the same CR LANs. Later on in the project, work migrated to the newer client CFR LANs.

4.2.1 Network Infrastructure

The basic network architecture is shown in Fig. 4.1. The network comprises of any number of client LANs interlinked by 2.048 Mbits/sec CCITT G732 Megastream circuits. Megastream links from each site are switched by a prototype ISDN switch in London. This pilot ISDN forms the Alvey High Speed Network [Keith 86] and provides each site with a Primary Rate interface as defined in the CCITT I-Series Recommendations. The interface between client LANs and the ISDN is by means of a Unison Exchange. A byte-parallel Cambridge Fast Ring [Hopper 86] forms the basis of this switch. A Portal provides access from a client LAN to the Exchange Fast Ring and a Ramp provides Primary Rate access from the Exchange to the ISDN.

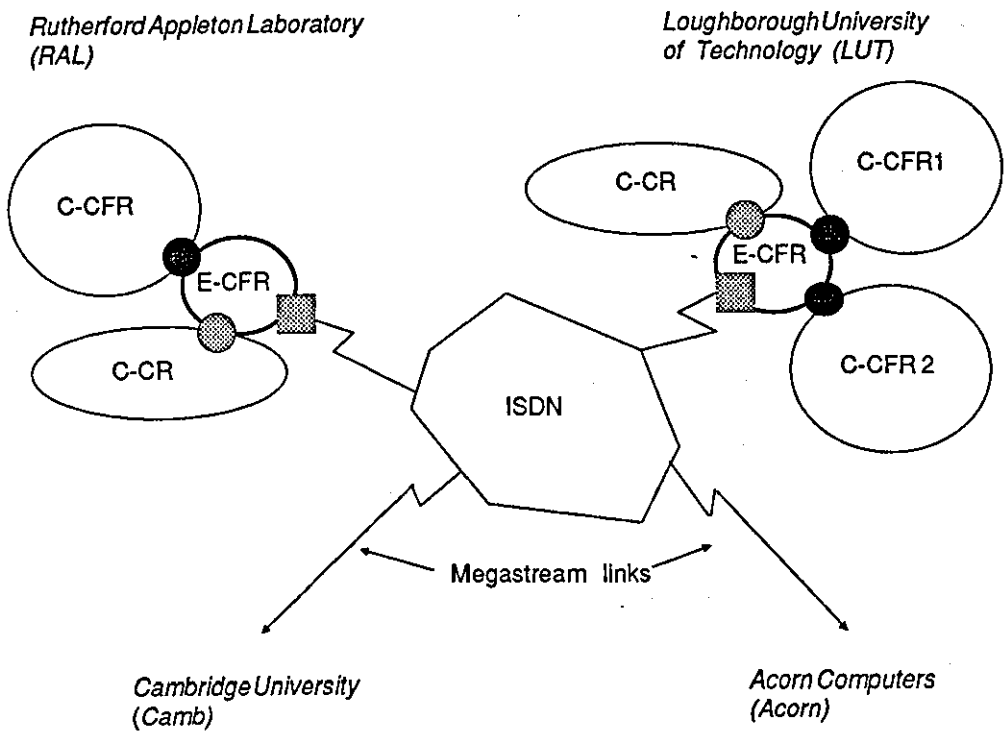


Fig. 4.1: The Unison Network Architecture

Each site may have a variety of client LANs, and each client LAN has its own portal. In the initial phase of the Unison Project, only CRs were used as the client LAN and all application components were developed and interfaced to this LAN. CR-CFR portals were developed to interface the CRs with the Exchange Ring [Tennenhouse 87]. At the latter stage of the Project, the availability of the CFR led to the use of the CFR as the client LAN. The application components were re-designed for the client CFR. This also resulted in the development of CFR-CFR Portals so that applications developed for the client CFR can be integrated into the Unison network.

4.2.1.1 Cambridge Ring

The Cambridge Ring is a slotted ring local area network and it is based on the empty slot principle. A slot, or minipacket, is a collection of 38 or 40 bits with the format shown in Fig. 4.2. There is a fixed number of slots continuously circulating around the ring. A station wishing to transmit data waits for an empty slot to arrive, fills in the data and address fields and marks it full. The slot travels around the ring to the destination station which marks the appropriate response bits to indicate whether it has been accepted or not. The slot then returns to the transmitting station where the response bits are read and the slot marked empty.

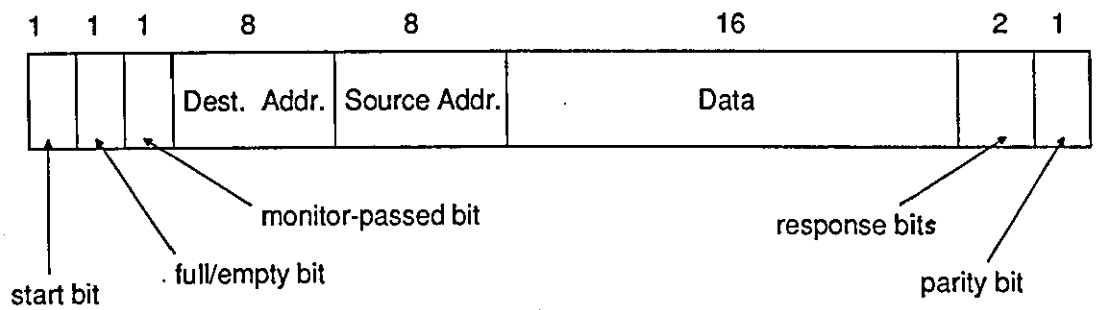


Fig. 4.2 Minipacket Format of the Cambridge Ring

There is an important CR protocol restriction that a station can only use one slot at a time. Furthermore, in order to prevent hogging, a transmitting station cannot immediately re-use a slot which it has just emptied. Also, due to technical reasons, it

is prevented from using the next slot [Logica 81a] . Thus the maximum point-to-point bandwidth is limited to s / (n + 2), where s is the system bandwidth and n is the number of slots on the ring. This is true when there is only one pair of stations communicating with each other as fast as the ring allows. If more than three pairs are communicating at the maximum allowed speed, then the system bandwidth would be equally shared amongst all pairs and as a result, the maximum point-to-point bandwidth of each pair would be reduced.

It has been argued that the performance characteristics of the slotted rings are suited to carry mixed traffic [Needham 87]. In an integrated services environment, one important characteristic of the CR is its fair bandwidth sharing mechanism. For instance, if S users request a small amount of bandwidth and M users request the maximum bandwidth, then S users will achieve their requested share almost immediately (with minimum queuing delay) and the M users receive equal share of the remaining bandwidth. This assumes that (S+M) times the bandwidth required by S users is less than the total bandwidth available. This property is highly attractive in the sense that low bandwidth real time applications can co-exist with high bandwidth application in a single network.

4.2.1.2 Network Protocol Layers

The following layers of communication protocol exist in the CR based Unison architecture. The protocol layers have been described in detail in [Adams 84].

a. The Minipacket Protocol

This is the lowest layer of the CR communication protocol and describes the operation of the CR access mechanism as discussed in the previous section. The protocol defines how the two byte data in a single minipacket are transferred from one ring station to another. For communication on a single ring, the minipacket protocol provides flow control and error control features. None of the Unison applications use this protocol for communication because the 16 bits of a minipacket are inadequate for most transfers.

b. The Basic Block Protocol

Immediately above the minipacket protocol is the basic block protocol. A basic block is made up of a sequence of minipackets and begins with a header minipacket.

is prevented from using the next slot [Logica 81a] . Thus the maximum point-to-point bandwidth is limited to s / (n + 2), where s is the system bandwidth and n is the number of slots on the ring. This is true when there is only one pair of stations communicating with each other as fast as the ring allows. If more than three pairs are communicating at the maximum allowed speed, then the system bandwidth would be equally shared amongst all pairs and as a result, the maximum point-to-point bandwidth of each pair would be reduced.

It has been argued that the performance characteristics of the slotted rings are suited to carry mixed traffic [Needham 87]. In an integrated services environment, one important characteristic of the CR is its fair bandwidth sharing mechanism. For instance, if S users request a small amount of bandwidth and M users request the maximum bandwidth, then S users will achieve their requested share almost immediately (with minimum queuing delay) and the M users receive equal share of the remaining bandwidth. This assumes that (S+M) times the bandwidth required by S users is less than the total bandwidth available. This property is highly attractive in the sense that low bandwidth real time applications can co-exist with high bandwidth application in a single network.

4.2.1.2 Network Protocol Layers

The following layers of communication protocol exist in the CR based Unison architecture. The protocol layers have been described in detail in [Adams 84].

a. The Minipacket Protocol

This is the lowest layer of the CR communication protocol and describes the operation of the CR access mechanism as discussed in the previous section. The protocol defines how the two byte data in a single minipacket are transferred from one ring station to another. For communication on a single ring, the minipacket protocol provides flow control and error control features. None of the Unison applications use this protocol for communication because the 16 bits of a minipacket are inadequate for most transfers.

b. The Basic Block Protocol

Immediately above the minipacket protocol is the basic block protocol. A basic block is made up of a sequence of minipackets and begins with a header minipacket.

Following this is a route minipacket and this is followed by between 1 and 1024 data minipackets. Finally, there is a checksum minipacket which is for error checking (see Fig. 4.3). The header minipacket contains a 4-bit pattern to identify it as the header, a 2-bit type field and 10-bit size field which indicates how many data minipackets there are in the block. The route minipacket contains a 12-bit port number which is used to provide sub-addressing within a host. The type field in the header identifies the block as being one of four types. The most common type consists of a block with data and an end-around-carry checksum packet. A second type has a checksum of zero and a third type consists of just the header minipacket with 10 bits of data in what would normally be the length field. The fourth type is a slightly different form of the first mentioned type, in which the second minipacket contains the block length and in the third minipacket is the routing information.

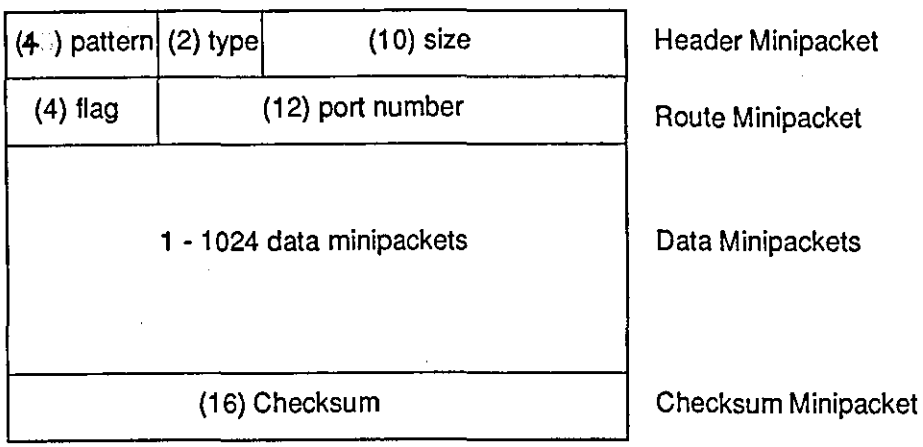


Fig. 4.3 Format of a Basic Block (the number in brackets represents the number of bits)

The basic block protocol provides a simple link level service in which the order of the blocks is preserved during transmission to a given receiver sub-address. In line with the assumption that bit error rates are low, no block acknowledgement scheme is provided. Bridging components are designed to discard blocks in the event of congestion; blocks lost caused in this way can only be detected by using a higher level protocol.

c. The Single Shot Protocol

The Single Shot Protocol (SSP) is based on the basic block protocol and provides a simple single transaction service. It is an exchange protocol in which an SSP request block (SSPREQ) is sent from a client to a server and is answered with an SSP reply block (SSPRPLY). The SSP is commonly used for control purposes and for distributed computing applications. A typical example of this protocol is a request for the time of the day. The requester sends a block asking for the time and receives a block containing the time information in reply.

d. The OPEN/OPENACK Protocol

This is a connection-oriented virtual circuit protocol and like the SSP protocol, it is also based on the basic block protocol. A connection is set up by exchanging blocks, the OPEN and OPENACK blocks. The only difference from the SSP exchange is in the use of the reply port and the addition of a connection port, contained in the OPENACK block. In the case of the SSP exchange, the reply port is only active until the reply is received. In the case of the OPEN/OPENACK exchange, the reply and connection ports remain active until the connection is closed. The circuit is closed by timeout rather than by explicit closing.

Once the connection has been established, the basic blocks can be sent over the established connection in both directions. The protocol is extremely lightweight in the sense that flow control and error control are left entirely to the hosts and are not performed by the network. This implementation of such controls will depend upon the nature of the services supported by the host. One important characteristic of the protocol is that the blocks travelling down a virtual circuit will arrive at the destination in the same order as they were sent.

4.2.1.3 The CR-CFR Portal

The CR-CFR Portal is a bridging device between the 10 Mbits/sec client CR and the 50 Mbits/sec Exchange CFR. It contains 68000 and 68020 microprocessors with CR and CFR interfaces. It essentially collects basic blocks from the CR and retransmits them as CFR minipackets and performs the reverse operation. The Portal communicates with the clients using the physical layer, link layer and management protocol of the client network. Communication with the peer Portals is accomplished

through the exchange of CFR minipackets. The Portal interactions with the exchange management are in accordance with the relevant Unison protocols and are out of band with respect to peer Portal traffic. Exchange management interactions are used to establish and maintain associations between peer Portals.

The Portal supports light weight virtual circuits which implies that flow control and error control between peer Portals are not provided and are left to the end-users. The Portal handles variable block sizes when communicating with the clients on the CR. It is interesting to see how the Portal behaves in a mixed traffic environment where applications can opt for any block size depending upon the nature of the application. Also the Portal implementation does not incorporate any priority traffic handling mechanism, instead a simple 'first in first out' strategy is adopted. Since the Portal provides bridging between two dissimilar speed LANs, it is equipped with a large number of buffers which are especially important when traffic is transferred from the Exchange ring to the client ring.

4.2.1.4 The Ramp

The Ramp provides connection between the 50 Mbits/sec Exchange CFR and the 2.048 Mbits/sec ISDN link. It employs a network of transputers as a high speed processing engine [Burren 89b]. The function of the Ramp is to map CFR minipackets onto ISDN time slots and vice versa. It listens on the CFR and extracts those minipackets that are destined for the selected remote sites. The minipackets are then transmitted in the appropriate timeslots on the ISDN link to a peer Ramp which injects them into the exchange CFR at the destination site. The bandwidth allocation between peer Ramps is done by the bandwidth management entities which reside at each Exchange CFR. The bandwidth is allocated in the form of U-channels which are some (between 1 and 30) multiple of 64 Kbits/sec. The individual intersite calls are aggregated together to construct higher bandwidth bearer circuits, the U-channels. Thus the bandwidth of specific inter-site channels can be dynamically increased or decreased according to the traffic loading.

The Ramp switches traffic in fixed-sized small packets and provides priority switching to real time low bandwidth application traffic. A large number of packet buffers are provided in the Ramp in order to cache sudden traffic surges. The Ramp behaves as a

remote bridge so the throughput-delay characteristics will depend on the amount of ISDN bandwidth available between peer Ramps.

4.2.1.5 Network Addressing and Routing Mechanism

The Unison network is really a number of independent local exchanges (Unison exchanges) connected by an ISDN. The clients reside on local networks which are connected to the local exchanges via the Portals. Communication can be established within a client network, between different client networks at the same site, or between remote networks at different sites.

The naming and routing mechanism is handled by the network management. For management convenience, the CR name space has been divided into a number of naming domains. A unique domain name is allocated to each site , allowing local management of service and machine names. The mapping between a name and the ring service address (station, port, function) required by the initial connection protocol is performed by the nameserver located on every ring. A nameserver maintains a list of all the services available within a single domain. Peer nameservers interact to resolve names from foreign domains.

The nameserver is also a fundamental component in the routing mechanism. Nameservers interact with bridging components to set up paths across the network on behalf of the clients. This operation is transparent to the client since a path is merely a ring service address on a bridge. The path setup would involve allocation of ports at the switching components (the Portals) so that they can map the user traffic from one side to the other. This path will remain fixed and can be used by the client and server for the duration of the call. Further details about Unison routing mechanism can be found in [Tennenhouse 87].

4.3 Performance Issues in the Unison Network

Two fundamental issues arise when considering the performance of the Unison network.

1. The Unison network is supporting multi-media applications, and in so doing, the network should meet the differing performance criteria imposed by each traffic type. The performance criteria of one application may entirely conflict with that of

another. For instance, for real time voice, corruption of a few packets is not as important as changes in packet delay. The reverse is true for the transfer of a text file between a terminal and a host computer. This can be explained further by studying the performance criteria imposed by each application.

- Performance criteria for synchronous voice traffic [Adams 85]:

  - Guaranteed upper bound for packet delay
  - Guaranteed network throughput between user ends once a call has established
  - Minimum statistical jitter in packet delay
  - Some packet loss permitted

- Performance criteria for synchronous video traffic [Lazar 87] :

  - Guaranteed upper bound for packet delay
  - High to very high throughput
  - No loss is permitted if frame update is very low

- Performance criteria for computer data traffic [Illyas 85]

  - No hard upper bound for packet delay
  - Low to high throughput
  - low/no loss permitted (sensitive to loss)

It would be extremely difficult to fulfill all these requirements in a real network. However, the extent to which the differing performance criteria are met by the network is a primary issue in any integrated services network.

2. In extending the functionality of LANs to a wider area, the performance of the bridging components is extremely important. With the advances in VLSI technology, these components provide faster data switching as compared to their predecessors. However, current applications demand real time response as well as high throughput for communication, the provision of which makes these components complex and unpredictable in their behavior. In the Unison network, the Portal and

the Ramp are considered the most critical components. The performance of these components would determine the network performance as a whole.

4.4 Measurement System implementation

4.4.1 Design Considerations

Before implementing a network measurement program it is important to consider what one hopes to achieve from such work and what depths the measurement should go into. This is because the network can yield an almost limitless supply of information, which would be impossible to analyse. Such considerations will also indicate what measurements are worth taking. This in turn will help when designing the measurement equipment and procedures.

4.4.1.1 Level of Detail

As described in the earlier sections, the Unison network supports three layers of protocol for communication between user ends, e.g., minipacket protocol, basic block protocol and application-to-application protocol (SSP or OPEN/OPENACK). Almost all the applications use basic block as the standard unit for information exchange. Since the main objectives of developing the measurement system were to assess the network performance as seen by the applications and to investigate the user observed performance of the Portal and the Ramp. It was decided to perform all measurements at the basic block level rather than of the minipacket level. However, as will be described in the next chapter, the equipment designed for the CFR based Unison network performs measurements on the minipacket level since most applications on the CFR use minipacket protocol for information exchange.

4.4.2 The Basic Measurement System

The basic measurement system consists of two TGRs (Traffic Generator and Receiver), a controller station and a data analysis machine. The TGR machines and the controller station are interfaced to the Cambridge Ring. The data analysis machine is interfaced to one of the TGR, the traffic receiver.

Both TGRs are equipped with similar hardware and software but functionally they act differently. This implies that either of the TGR can act as the traffic transmitter or traffic receiver. Traffic generation and reception activities between the TGR machines are controlled by the controller station. The controller station is used to initiate tests, control traffic parameters, collect results from the TGR machines, etc. The TGR machines have limited on-line data analysis capability. The data analysis machine connected to one of the TGR machines provides a data storage function for off-line analysis.

4.4.2.1 TGR Machines

The basic components of a TGR machine are shown in Fig. 4.4. Each TGR machine is based on a multibus architecture and comprises four main components.

1. A programmable Single Board Computer (SBC)
2. A dedicated Basic Block Interface (BBI) board between the SBC and the Cambridge Ring (CR) node
3. A unique CR Polynet node
4. A Rugby Clock Receiver (RCR) with an interface board between the RCR and the SBC

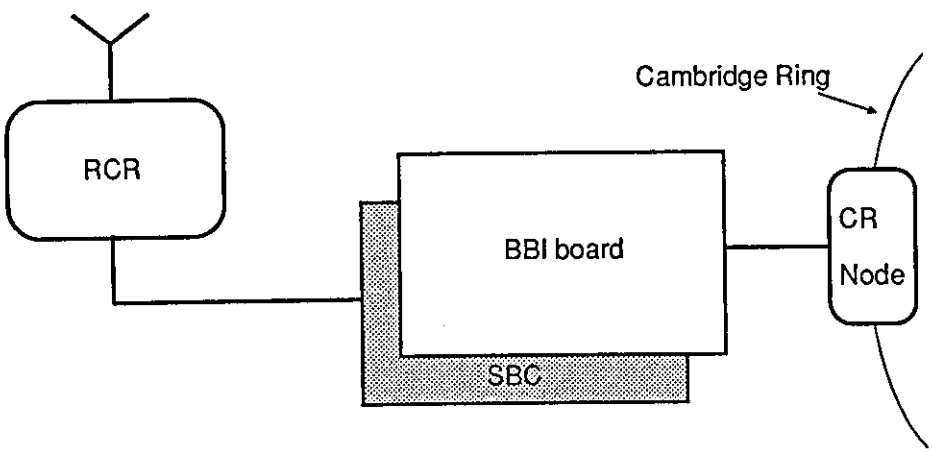


Fig. 4.4 Basic Components of the TGR machine

Hardware Details

The Single Board Computer (SBC) is a 86/30 system designed around the 16 bit 8086-1 microprocessor with a clock frequency of 8 MHz. The system has 256 Kbytes of dual port memory of which 64 Kbytes are used by the BBI as shared memory, for transferring data to and from the ring. Also it has 32 Kbyte of EPROM space for storing the software for the ring driver and traffic generating functions. The 86/30 system also has on-board programmable devices, i.e., interval timer, interrupt controller and I/O devices. The interval timer and interrupt controller are used in delay measurement and will be explained in the latter sections.

The Basic Block Interface (BBI), better known as VMI-1 [Logica 81b], is a high performance interface between the SBC and the Polynet node. It contains the firmware for supporting the basic block protocol. The interface uses direct memory access to the SBC memory to transfer data to and from the SBC. The VMI-1 signals the completion of the request by interrupting the SBC. It also passes a return code to the system software in the SBC. This code indicates the status of the transmitted or received block.

A Rugby Clock Receiver (RCR) is used for synchronisation purposes. Further details about its implementation follow in the succeeding sections.

Software Details

The structure of the TGR software is very simple (see Fig. 4.5). The ring driver code implements the Basic Block Protocol and communicates with the VMI-1 in structuring the basic blocks. The ring handler handles the incoming control and data traffic. It implements the higher layer protocols such as single shot protocol or open/openack protocol. These protocols are responsible for setting up connections between remote machines and to have short inquiry transactions. At the top layer, there is code for the traffic generating functions. All the data traffic filtered through the ring handler code is passed to the respective generating function.

The traffic generator software was developed on an Intel Ethernet based microprocessor development system and blown into EPROM. The software was developed in Pascal and 8086 assembly language. The bulk of the code (ring driver,

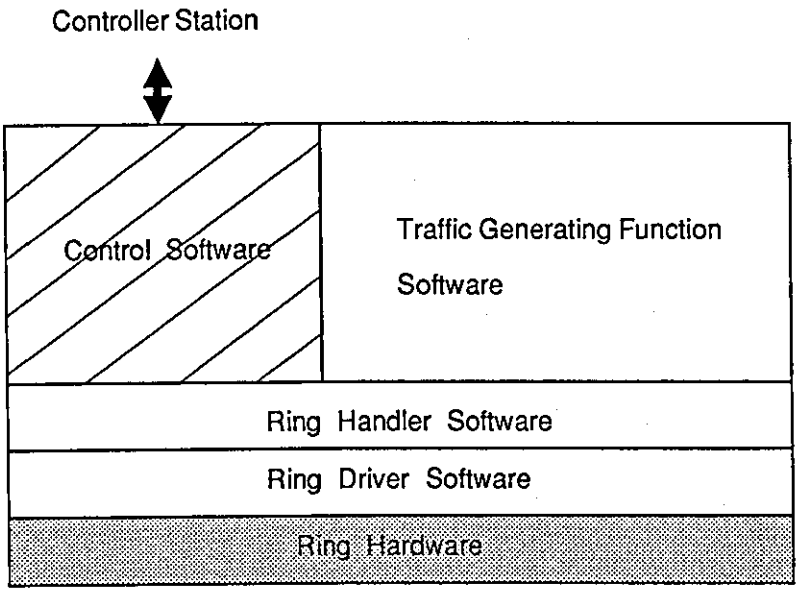


Fig. 4.5 Illustration of the TGR software sturcture

ring handler, and some generating functions) was written in Pascal while the time critical code such as the code for timestamping was written in 8086 assembler.

4.4.2.2 Controller Station

The controller station provides a menu driven user interface for controlling the traffic activities between the TGR machines. A BBC microcomputer (with ARM as the 2nd Processor) [Acorn 87a] is used as the controller station (see Fig. 4.6). It is interfaced to the client CR via the SEEL/ORBIS program interrupt interface which provides the basic block level protocol in firmware. Communication between the controller station and the TGR machines is via the network. The controller station is capable of controlling the TGR machines from any where on the network.

The controller station is normally used to control test traffic parameters, e.g., test duration, basic block size, generation rate, generation pattern, etc. The controller

station also displays the results which are sent by the TGR machines at the end of the test.

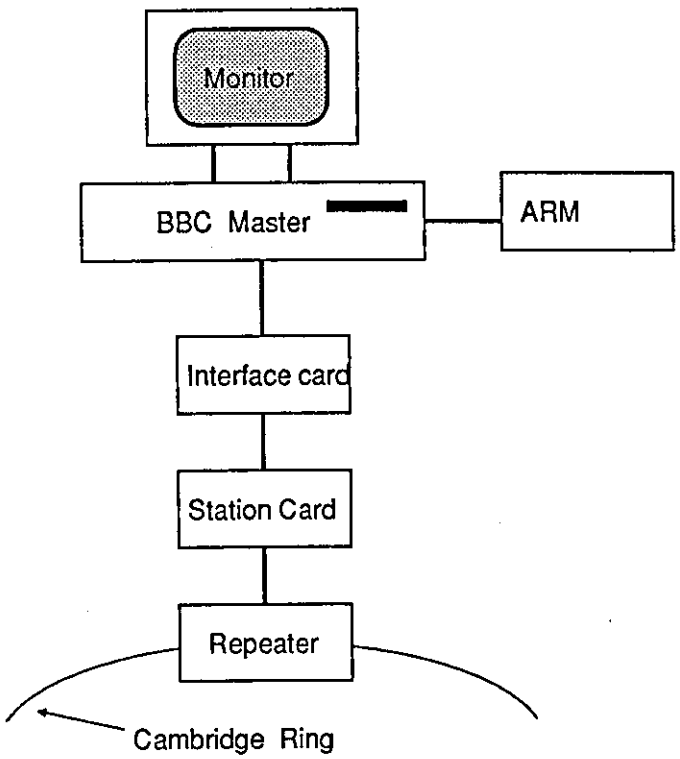


Fig. 4.6 The TGR Controller Station (main Components)

The software for the controller machine includes the ring interface code and the main controller program. The ring interface is based on a version of the Project Universe 6502 Ring Driver modified to run in an Acorn Master microcomputer. The ring driver implements the Basic Block Protocol. The software for the main controller program was written in BBC basic.

4.4.2.3 Data Analysis Machine

The data analysis machine is interfaced to the TGR machines and is responsible for storing results for off-line analysis. Although the TGR machines have some on-line data analysis capability, at high traffic rates the analysis overhead affects the normal operation of the TGR machines.

A BBC Master microcomputer has been interfaced via a 1 MBus communication port [Acorn 87a]. A parallel port is used on the SBC side which communicates with the 1 MBus in order to transfer data from the TGR machines to the BBC Master. An assembly routine running in the Master accumulates and stores data on a hard disc attached to the master. A detailed statistical analysis of the data collected is made available after the test experiment. The schematic diagram in Fig 4.7 shows the connection of the data analysis machine with the TGR machine.

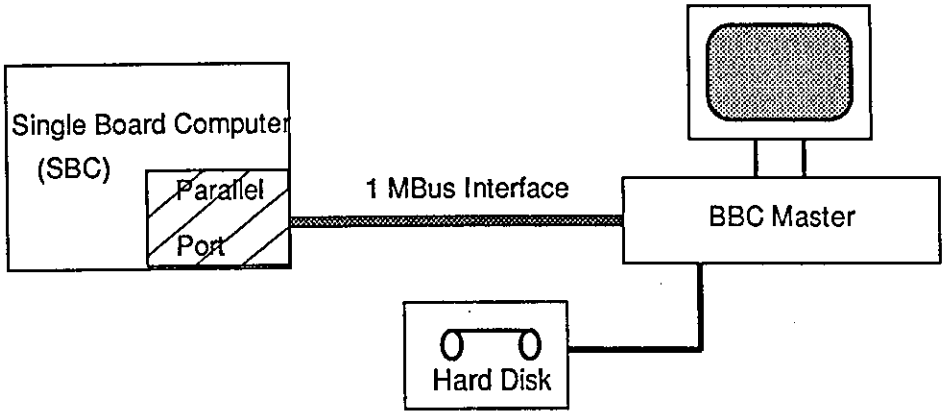


Fig. 4.7 Data Analysis Machine Connected to the SBC of the TGR

4.4.2.4 Equipment Placement and control

The Unison network is a multi-site network, so flexibility in equipment placement and its control is very important. The minimum requirement in conducting a measurement experiment is to have two TGR machines (a traffic generator and a traffic receiver) and a controller station. A test experiment can be carried out at the same site by putting TGR machines at the same site, or between sites by placing TGR machines at different sites.

Similarly, there should be flexibility in controlling the traffic activities between the TGR machines. A single controller station at a site should be able to control two machines at the same site or between any two sites.

4.4.3 Rugby Clock Receiver

The Rugby Clock Receiver receives the time signal transmitted at 60 KHz from the MSF, station Rugby, England. In most countries, a radio frequency is allocated for transmitting accurate timing information. In the U.K., the National Physical Laboratory (NPL) maintains a radio transmitter which is used to provide a continuous source of world time [NPL].

The clock receiver receives time code during each one minute cycle at a rate of 1 bit/sec.

4.4.3.1 Hardware Operation

The clock receiver generates two signals only : CLK and DATA. The former is the second pulse whose leading edge marks the beginning of every second, while DATA is the actual decoded time signal. Only the CLK signal is taken from the clock receiver and this signal is passed through a filter circuit to remove any noise superimposed on the CLK signal. After converting the signal into a TTL level signal, it is ready to be interfaced with the SBC. The signal going into the SBC is referred to as the Rugby Tick. Further details about the Rugby clock interface are described in Appendix 2.

The SBC has on-board programmable interval counters and an interrupt controller. A 4MHz crystal oscillator provides clock frequencies to the interval counters. The Rugby Tick is fed to two interval counters. It acts as a 1Hz clock for one counter (the Second Counter) whilst in the second (the offset Counter) it generates a reset rate of 1 Hz.

4.4.3.2 Software Operation

When a TGR machine is switched on, the offset counter is initialised to the mono-shot mode and is loaded with a large integer number (16-bit number) assuring the mono-shot pulse duration is more than one second. As mentioned earlier that the offset counter is reset at a frequency of 1 Hz, so that each successive count of the offset counter multiplied by the clock period, represents the elapsed time with respect to the previous Rugby Tick. The clocking frequency of the offset counter determines

the timestamping error. This error is around 16 usec which is almost negligible since the packet delay is in the order of tens of milliseconds.

The second counter, as the name suggests, is used to count the number of seconds passed with reference to the last synchronisation. At the time of synchronisation, this counter is reset by software and since it is clocked by the Rugby Ticks, so each successive count thus represents the number of seconds elapsed since the last synchronisation.

4.4.3.3 Basic Operation : Synchronisation and Chronological Timestamping

The basic operation of synchronisation and chronological timestamping using two test stations is described. The operation for multiple stations is discussed in a later section.

Synchronisation

The synchronisation procedural protocol between two TGR stations is simple. Let us assume that the controller station has configured one TGR station as a Transmitter (TX) and the other TGR station as a receiver (RX). The TX is told to synchronise with the RX. The TX, being the initiator, performs the following operations.

1. After receiving a message from the controller station, the test station unmasks the Rugby Tick interrupt and waits for the first interrupt.

2. Immediately after receiving the interrupt, it sends a message to the RX saying 'reset Second Counter' and acknowledges the reply from the RX.

3. Resets its own Second Counter.

4. Makes sure that step 2 and step 3 are completed before the next Rugby Tick interrupt occurs and finally informs the controller station about the success of the whole operation.

The operation in the RX is simple. In response to the message from the TX, it resets its Second Counter and sends an acknowledgment to the TX.

After the above operation, the Second Counter of both the TX and RX should have the same values for every Rugby Tick. The Offset Counters of both the TX and the RX

are always in synchronisation with each other (by virtue of hardware) because the Rugby Ticks are resetting the reset counters at the start of every second tick.

Chronological Timestamping

Once the synchronisation is achieved, the counts from both counters can be read (on the fly) at any time by the processor without disturbing the operation of the counters. On the TX side, packets are stamped with the second number and second offset (offset from the previous second number) just before launching onto the network. Similarly, on the RX side, the packets received are stamped with their own second number and second offset. Since these stamps contain chronological ordering values, the difference of the stamps (RX stamp - TX stamp) determines the one-way packet delay.

4.5 Setting up an Experiment

This section describes how an experiment is being setup using the measurement components.

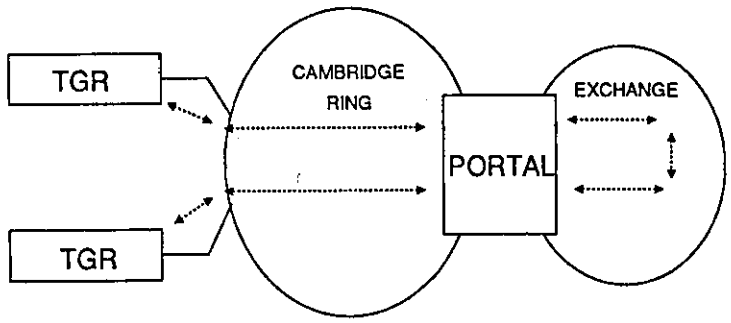
4.5.1 Defining a Network Segment

The first thing will obviously be to define a network section on which a performance experiment is to be conducted. Depending upon the network section, the traffic route and associated TGRs would be determined.

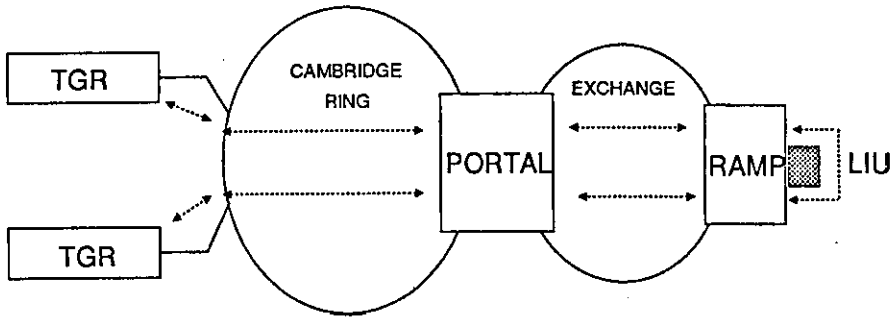
In the Unison network, a traffic loop back facility is provided in order to select the number of components in a communication chain. The use of this facility enables the data traffic to be returned to the originating client network after traversing the pre-defined communication chain. A number of communication chains are possible.

a. Short Loop back

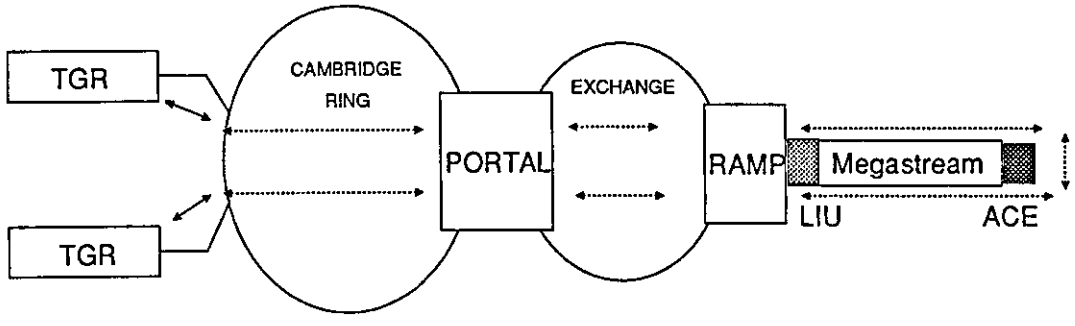
The network components selected in the short loop back communication chain includes a Cambridge Ring (CR), a Portal and an exchange ring (see Fig. 4.8). The data traffic originated from the client CR makes its journey to the Portal, onto the exchange ring, is switched back to the same Portal, and finally passed back to the originating client CR. The critical components in this chain would be the Portal, and extensive performance analysis of this chain will lead to an in-depth understanding the Portal's performance.



a) Short loop back test configuration



b) Long loop back test configuration



c) Mega - long loopback test configuration

Fig. 4.8 Typical Test Configurations

b. Long Loop back

This communication chain involves the client CR, the portal, the exchange ring, a Ramp and a Line Interface Unit (LIU) as shown in Fig. 4.8. The traffic path is similar to the one in the short loop back except for the additional two-way path through the Ramp via LIU. The communication chain has two bridging components in the traffic path and analysis of this chain will enable the effect of adding another component to be determined.

c. Mega-long Loop back

This communication chain is very similar to the one in long loop back except the addition of a Megastream path (ISDN link connecting the LIU with an Automatic Cross Exchange). The traffic is now reflected back from the Automatic Cross Exchange rather than from the LIU as is the case in long loop back (see Fig. 4.8). The performance characteristics of this communication chain would depend upon the amount of bandwidth available on the ISDN link. By varying this bandwidth, the performance of the Ramp can be observed.

Other Configurations

Other typical network configurations include inter-site network configuration (see Fig. 4.1) and inter client ring configuration at the same site. In the latter case, two client CRs are connected to the exchange ring via two Portals. This configuration is possible when two departmental rings are to be inter-connected or one large ring is broken into two for the sake of performance and reliability.

In the inter-site configuration, each Unison site has similar network components, i.e. client CRs connected to the exchange ring via Portals and the exchange ring connected to the ISDN links via Ramps. The Automatic Cross Exchange interconnects Unison sites via ISDN links.

4.5.2 The Test Controller Station

The Test Controller Station (TCS) provides an intelligent command interface between the TGRs and the experimenter. The TCS is a unique station on the client network which can be anywhere on the Unison network.

When the controller software is run, a menu will pop up displaying the various test options. In each test option, the traffic pattern which is to be generated and the traffic parameters associated with each pattern, the direction of traffic flow, the number of TGRs involved in the test, are selected. For instance, if the burst traffic pattern is selected, then the associated traffic parameters such as the block size, burst size, inter-burst time, and inter block time within a single burst need to be specified. Since each TGR machine can behave as a traffic generator or traffic receiver, one has to specify which machine is the generator and which is the receiver along with the direction of the flow of the traffic. Once all these parameters are specified, the controller station is ready to initiate a test experiment.

Each TGR has a name on the network, such as Indus and Ravi (famous rivers in Pakistan). Depending upon the network segment selected and the direction of the data traffic flow, the controller station will communicate with the TGR machines which have been selected as the traffic generators. In order to access the TGR machines over the network, the controller station makes an SSP transaction to the nameserver by providing the machine location followed by the machine name, i.e. Ral*Indus. In return, the nameserver provides the station address along with the port number and sets up the path between the controller station and the TGR machine. Once the path is established, the controller station sends an OPEN to the machine concerned along with the traffic parameters which have been selected by the experimenter.

The traffic generator makes an association with the traffic receiver and the traffic generation test will be initiated for the duration of the given test time. At the end of the test, the traffic receiver sends the result log to the traffic generator. The traffic generator, in turn, passes this result log along with its own result log to the controller station. The controller station displays the results. If more than one pair of TGR machines are involved in the traffic generation test, then the controller station displays the results generated from each pair of TGR machines.

4.5.3 Traffic Generation

Once the network segment on which the traffic is to be generated is made known to the traffic generator, the traffic generator makes a SSP transaction to the nameserver, providing the machine name of the traffic receiver along with the necessary routing information in accordance with the pre-defined network segment. The nameserver

uses this information to establish the path on behalf of its clients, the traffic generator. The traffic generator, then initiate the Open/Openack transaction over the established path to set up a virtual connection with the traffic receiver.

According to the traffic pattern and the traffic parameters provided by the test controller station, the traffic generator calls the appropriate traffic generating function and updates the previously set parameters (or default parameters) with the new parameters. A number of test experiments can be repeated by slightly modifying the traffic parameters. The repeatability facilitates comparative investigations. While generating traffic, the traffic generator makes histograms or counts of the requested parameters. Similarly, the traffic receiver collects statistics on the receive side. The statistics are passed on to the controller station at the end of the test.

A number of traffic generating functions are supported by the traffic generators, i.e. a deterministic traffic generating function, a burst traffic generating function, a speech traffic emulation function, a video traffic emulation function. In the deterministic generating function, the traffic is generated with a constant block rate. The maximum generation rate can approach the maximum point-to-point bandwidth of the Cambridge ring. The traffic parameters which can be selected for this function include block size (in bytes) , inter block delay, traffic generation time, chronological timestamping enabled or disabled. In the case of the burst generating function, the generator periodically generates a burst of blocks. This traffic emulates computer traffic which usually exhibits bursty traffic characteristics.

In the speech emulation function, conventional speech traffic can be emulated by generating a constant block rate which corresponds to an effective bit rate of 64 Kbits/sec. Silence supressed speech traffic can be emulated by generating traffic only during talkspurts. In the first case, the traffic generated is very similar to the one generated by the deterministic function, however, the performance statistics would reflect the end-to-end performance for a speech connection. The silence supressed speech traffic is generated with a geometric distribution function which has been shown to emulate real silence compressed speech traffic [Gruber 82]. It was not our intention to accurately simulate such traffic because a number of human factors are involved which can affect the simulated traffic, i.e. some people talk too fast with small silence intervals while some talk as if they are not talking at all. What is

important however is how accurately and precisely one can measure the user-oriented performance parameters by injecting such traffic into the network.

The video emulating function also supports two types of traffic emulation, i.e. constant block rate generation and variable block rate generation. The constant block rate traffic generation is the same as in the deterministic traffic generation. The variable block rate traffic generation is modelled by storing an abridged set of results from a real VBR coder, and scaling these according to the load required [Chin 88].

These traffic generating functions are described in detail in Appendix 1.

4.5.4 Delay Measurements

The single-way block delay measurements are achieved by synchronising the clocks of the TGR machines. The accuracy in the delay measurements will depend on how accurately the clocks of the test machines can be synchronised with each other.

When the TGRs are switched on, their clocks are in the non-synchronised state. The clocks between a pair of machines are only synchronised when a performance experiment is to be conducted. Both synchronisation techniques, broadcast synchronisation and turn-around synchronisation as discussed in Chapter 3, have been implemented in the TGR machines. The reason behind implementing both synchronisation techniques was to compare and confirm the measurement results. When the test controller station enables the timestamping parameter, the generator and receiver communicate with each other in order to synchronise their clocks. This is done just after the association establishment. How it is done has already been explained in the previous sections.

The clock values are accessible through software and each generated block is timestamped with the chronologically ordered values at the generator and the receiver. The difference between the timestamps is used to measure the single-way block delay. For inter-site tests (between LUT and RAL), the Rugby clocks were calibrated such that the 1 second pulses were generated with a relative accuracy of 1 msec [Parish 88]. The synchronisation error is about 2 % for a typical 50 msec block delay. This error is negligible for large block delays but it would be significant for small block delays.

4.5.5 Measurement Data Collection

The statistics collection functions in the TGR prepare histograms of the requested performance parameters. The controller station provides the parameter list for which the statistics are to be collected and also provides bounds for each histogram, i.e. lower and upper bounds of the histogram block range, the range of each block. The performance statistics include the block launching time statistics, single-way block delay statistics, inter-arrival time statistics, block lost statistics, sequence error counts, retransmission counts, etc.

At low generation rates, the processing overhead due to the statistics collection functions is negligible. But at higher rates, the traffic receiver is unable to cope with the extra processing burden required for statistics collection. In order to overcome this problem, a data analysis machine has been interfaced to the traffic receiver. This enables the fetching of on-line measurement data from the traffic receiver, thus providing extra processing power by sharing the statistics collection overhead. Also it provides a large storage facility for off-line analysis of the collected measurement data. The statistical results can either be dumped on the statistics gathering machine or they can be passed to the controller station for display/hard copy print outs. The histogram based statistics are then used to prepare various statistical distributions which facilitate further analysis of the network performance.

4.6 Performance Measurement Experiments

Before conducting a measurement experiment, one should first determine what technical questions need to be answered by the measurements and what performance reports need to be generated. The questions can be wide ranging. These questions can originate from various groups of individuals depending upon how someone is associated with the network under consideration. The designer group would be more interested in the operational performance of the various network components in a mixed traffic environment. A user might ask how well his individual traffic stream is being handled by the network when a number of traffic streams are passing through the network and is he getting a fair share of the network resources. The managerial group would care more about the reliability, availability and utility of the network

resources. The measurement experiment are then designed to gather network performance information which can satisfy these questions.

In the Unison network, the main aim in performing measurement experiments was to provide an on-line source of performance information for the network designers and the network users. Due to the experimental nature of the network, no attention was given to performance information which satisfies the network managers.

The experimental process for performing a measurement test can be broadly divided into four main steps. Firstly, the object of the experiment needs to be clearly defined, i.e. the specific aspect of the network which is to be investigated, for instance the performance of the bridging components. Secondly, the selection of the performance parameters which can best characterise and quantify the object under study. The average block delay and throughput are some of the conventional performance parameters which are considered important when investigating the performance of the bridging components. Thirdly, the design of an experiment that can fulfil the requirements of step 1 and step 2 in the best possible way. This is perhaps the most difficult and delicate part of the whole experimental process. This step involves a number of considerations: what should be the network configuration for performing the experiment, what should be the test environment in which all the necessary network effects can be captured effectively, what should be the test duration which can produce adequate performance statistics, what are the factors which can effect the experiment results, etc. Finally, the proper interpretation of the measurement results which are the end product of the experimental process. Graphical representation of the performance statistics helps in understanding the object of the experiment.

A number of measurement experiments have been carried out over the Unison network. A summary of the important experiments is presented here.

4.6.1 The Performance of the TGR on the CR

It is always advisable to start with a simple experiment. The experiment was conducted to determine the maximum traffic generation rate that can be achieved using a pair of TGR on a CR. There are a number of factors which influence the maximum generation rate or the maximum point-to-point bandwidth on the CR. The maximum point-to-point bandwidth is calculated as [Logica 81a]

Useful Bandwidth (BW_{uf}) = ((N_s * U_d) * C_f) / ((N_s * M_b) + G_b)

Where

C_f Clocking frequency of the CR (raw bit rate)

U_d Useful data bits in a minipacket

M_b Minipacket size in bits

G_b Gap size in bits

N_s Number of slots on the ring

assuming the gap size to be very small, then the point-to-point Bandwidth (BW_{pp}) is given by

$BW_{pp} = BW_{uf} / (N_s + 2)$

The above equation assumes that there is only one pair of stations communicating with each other. If S_t stations are simultaneously transmitting at the maximum allowed speed, then

$BW_{pp} = BWuf / (N_s + S_t)$

The main factors which effect the BW_{pp} are the ring size, the number of stations using the ring and their transmission rates. When the gap size is large, the BW_{pp} is also affected as a result. For a typical Polynet CR, where C_f is 10 Mbits/sec, U_d 16 bits, M_b 40 bits, and assuming N_s is 1 with zero gap bits (G_b), then BW_{uf} is 4 Mbits/sec. The maximum BW_{pp} available for a pair of TGR will be 1.3 Mbits/sec (for a one slot ring with no gap bits). This would vary if the bit latency of the ring is varied. Experiments were conducted at some typical ring latencies. Fig. 4.9 shows the maximum traffic generation rates achieved for the various ring latencies. The ring latency has a proportional effect on the block launching times on the CR, i.e. the longer the ring size the more time the minipacket would take to get around the ring. This is shown in Fig. 4.10 which shows the block launching time at some typical ring sizes. As can be seen a one bit increase or decrease in the ring latency (at certain points) can considerably effect the traffic generation performance of the traffic generator. This result is equally valid for any high bandwidth user machine using the CR as a client CR. Also, when designing the bridging components in order to

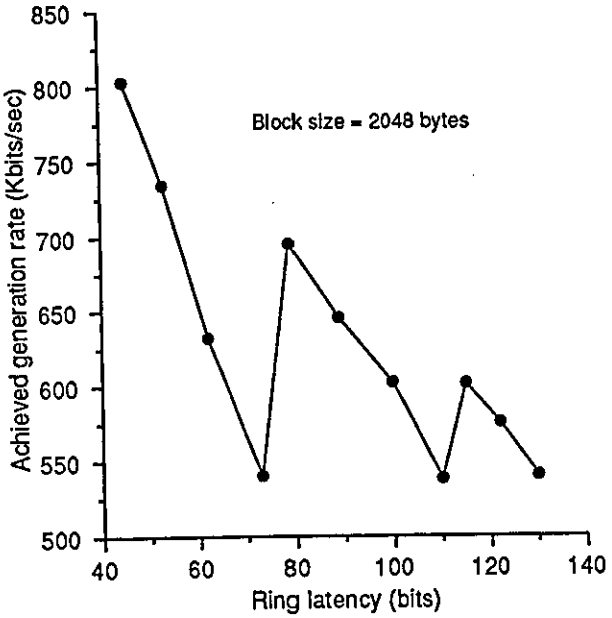


Fig. 4.9 Achieved Generation Rate at Various Ring Latencies

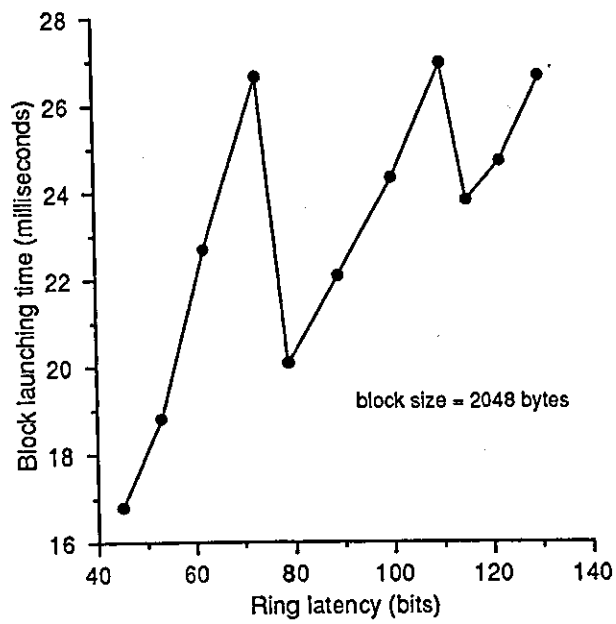


Fig. 4.10 Block Launching Time at various Ring Latencies

interconnect two dissimilar sized CRs, such performance considerations have to be taken into account, for instance the amount of buffering required in the bridge, flow control strategy, etc. Although, the ring size usually remains static for most of the time, insertion/removal of stations would result in an increase or decrease in the ring size.

4.6.2 The Portal Performance

In order to determine the traffic throughput (blocks/sec or kbits/sec) and block transfer delay of the portal, a number of performance experiments were carried out using the short loop back configuration. For all these experiments, the ring latency was kept fixed at around 2 slots and 20 gap bits. For most of the experiments, the deterministic traffic generating function was used to generate traffic in determining the performance of the portal. The throughput performance of the portal at various block sizes is shown in Fig. 4.11. As can be seen, throughput for the maximum block size is quite low (23 blocks/sec) but it rises upto a maximum of 175 blocks/sec for the smallest block size. The same figure also shows that the actual useful bit rate reduces

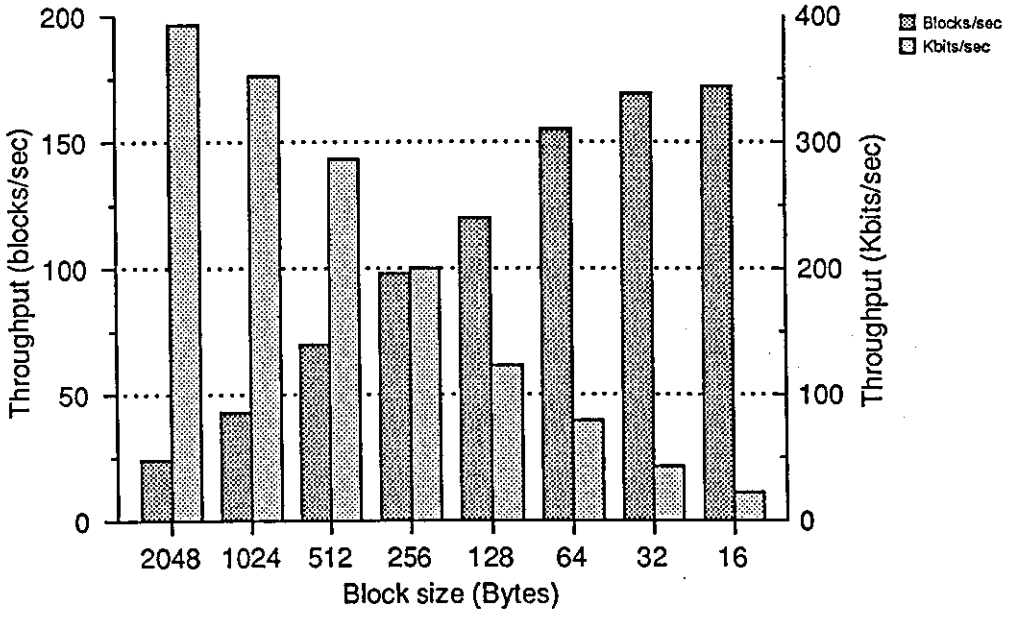


Fig. 4.11 Maximum Throughputs at Various Block Sizes in Short Loop Back Configuration

quite considerably as the throughput in blocks/sec rises. This figure clearly demonstrates two fundamental limits in CR portal performance : firstly the maximum number of blocks which the device can transfer and secondly the maximum number of bits/sec transferred. This presents a serious problem for some real time applications. For instance, the voice applications use smaller block sizes in order to reduce packetisation/de-packetisation delay. If for example, a 32 byte block size is used for voice communication, one 64 kbits/sec voice connection (without silence supression) would generate traffic of the order of 250 blocks/sec. The implications for speech communications are significant; large block sizes with large end-to-end delay would have to be used.

Fig. 4.12 shows the minimum block tranfer delays experienced in the portal in short loop back configuration. The traffic loading is kept reasonably low in order to ensure that the blocks should experience minimum delay. The total loop delay includes the block launching and formation delay on the CR and delay in the loop components (the portal and the exchange ring). It can be seen that block launching and formation contributes a high proportion of the total loop delay.

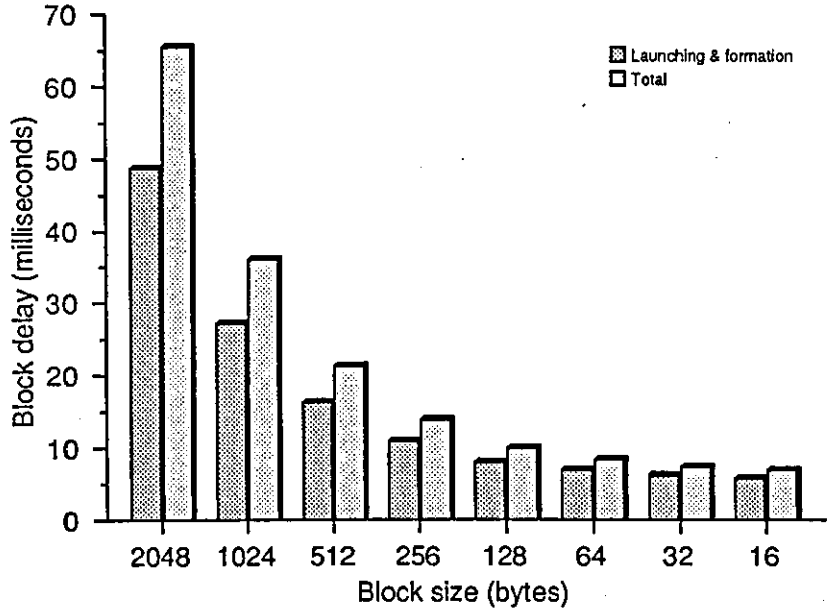


Fig. 4.12 Block Delay across Portal at Various Block Sizes in Short Loop back Configuration

The theoretical throughput performance of the portal is also restricted by the maximum point-to-point bandwidth of the CR. This limitation is due to the low level CR protocol as explained in the earlier sections. Furthermore, the point-to-point bandwidth depends on the ring latency. The point-to-point bandwidth decreases as the ring latency increases.

4.6.3 The Ramp Performance

The ramp behaves as a remote bridge (sometime referred to as half bridge): two ramps interconnect Unison sites via the ISDN links. The performance of such bridging components would primarily depend upon the amount of ISDN bandwidth available between peer ramps. The ramp performance was evaluated in the long loop back configuration. The console terminal connected to the ramp is used to manually change the ISDN bandwidth. The ISDN bandwidth is varied in units of slots where one slot is equivalent to 64 kbits/sec . Fig 4.13 shows the block transfer delays at typical bandwidths of the ISDN loop. As expected the block latency in the ramp is a function of the ISDN bandwidth. The graph indicates that if the ISDN bandwidth allocation is restricted, the delay in the ramp increases considerably especifically if large block sizes are used for communication.

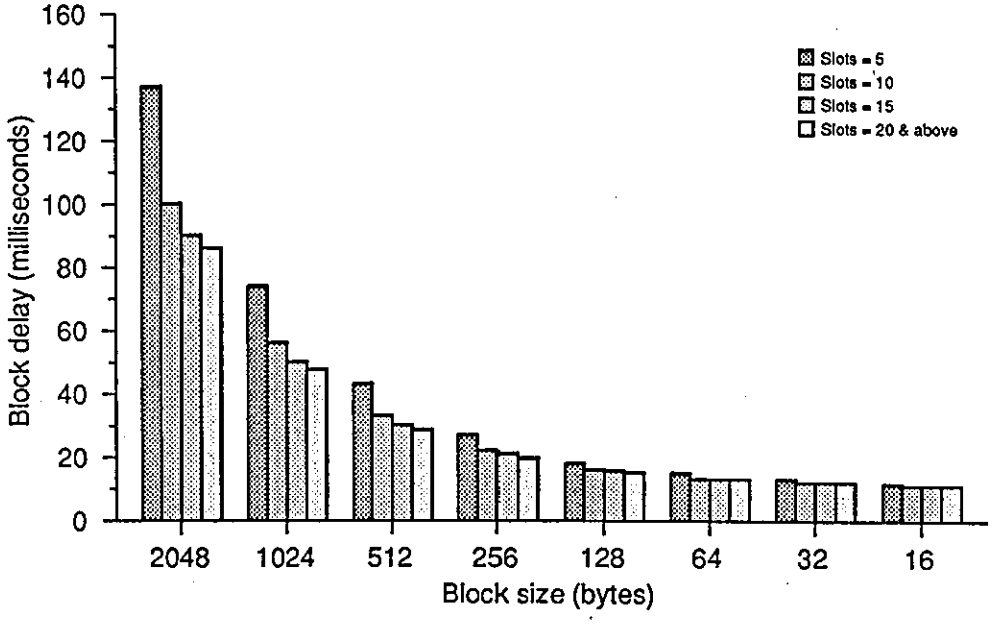


Fig. 4.13 Block Delay Vs Block Size at typical ISDN Bandwidths in Long Loop Back Config.

The throughput performance during long loop back tests remained very similar to that of the short loop back cases. Ideally, the throughput-delay characteristics can be best determined if the Ramp is loaded from the Exchange ring rather than from the client ring. This is because the portal acts as a first bottlenech component in the long loop back configuration. The Ramp therefore cannot be overloaded if traffic generators are connected to the client ring. However, CFR based traffic generators were built latter which can be connected to the Exchange ring. The extensive performance experimentation with the Ramp is discussed in the next chapter.

4.6.4 Inter-site Experiments

In order to conduct inter-site performance experiments over the Unison network, the RAL site was selected (LUT was the main site for data collection and analysis) as the second site mainly because of the availability of manpower for occusional help (in terms of setting up the system components). The TGR was placed on the client CR at RAL. The Rugby time clock receivers between RAL and LUT were calibrated such that clocks at both ends were synchronised with a relative accuracy of one millisecond. The ISDN bandwidth between RAL and LUT was increased from the usual 10 slots to 20 slots. This eliminated the chance of ISDN bandwidth (between peer ramps) being the performance restricting factor for inter-site throughput and delay test experiements.

The experiments have shown that inter-site throughput performance very much depends on the throughput of the portals. The inter-site throughputs for various block sizes remained similar to throughputs achieved in the short loop back tests. The block delay transfer times for various block sizes is shown in Fig. 4.14. The main aim of the inter-site experimentation was to determine the end-to-end performance characteristics for a typical voice connection in a mixed traffic environment. For this, the network was loaded with other application traffic on top of the traffic genearted from traffic generators. The application emulation function was used to generated constant block rate voice traffic. The block size for the traffic was selected as 256 bytes. This block size had been used in the real voice applications. The slow scan video traffic was used to load the inter-site communication chain. The video traffic used 2048 bytes block size.

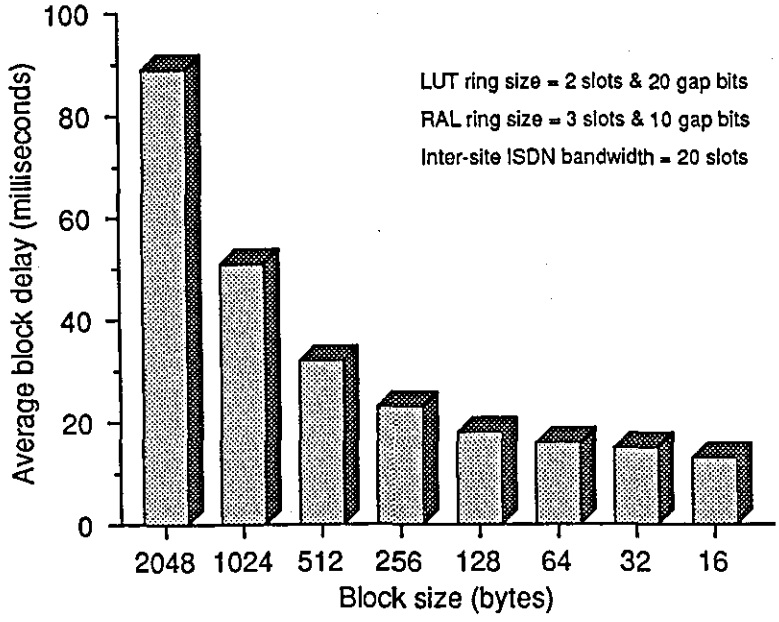
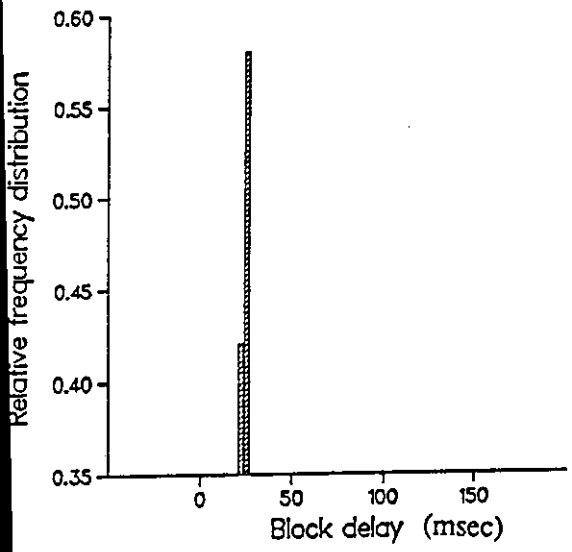
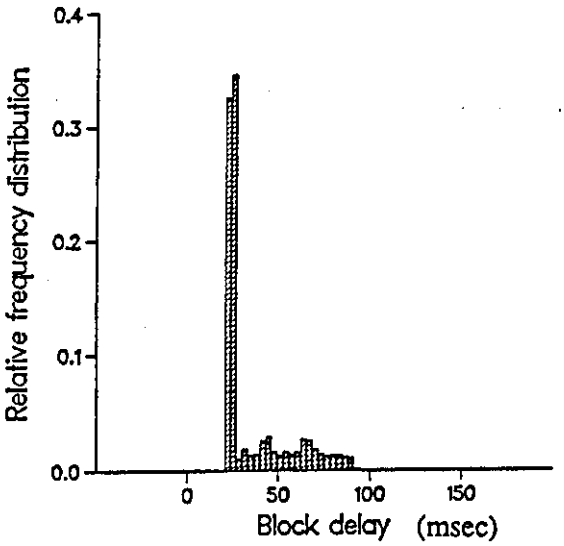


Fig. 4.14 Intersite Block Delay at Various Block Sizes

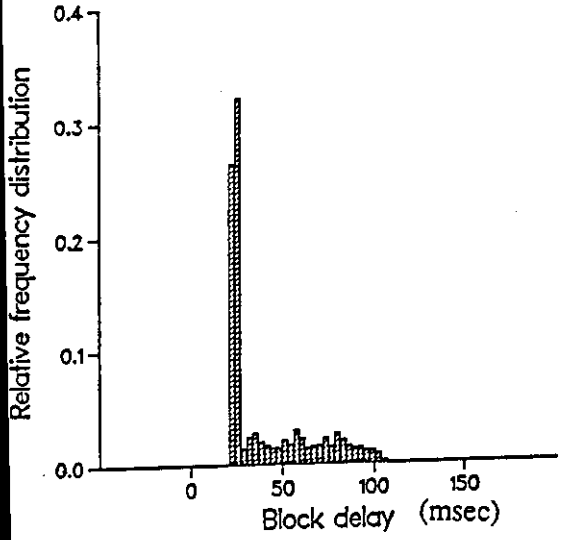
Fig. 4.15 shows the relative frequency distribution of speech block delays for the test generator traffic in parallel with the video traffic. The figure shows an increase in spread of the block delays as the video application traffic is increased. With a traffic loading of two video sources, the spread in speech block delays is as much as 5 times the minimum block delay. The main reason for such behaviour is that the CR portal switches user traffic in basic blocks. Since the block size is variable, the applications can choose any block size in accordance with the basic block protocol. Since the portal cannot multiplex user blocks in transmission or reception, this introduces contention in the middle of the network. There are two consequences. Firstly, the basic block protocol puts additional processing overhead on the portal. For smaller block size, this overhead impairs the effective data throughput of the portal. Secondly, the portal receives or transmits one block at a time, therefore the applications which use smaller block sizes for communication are effected severely in terms of fairness in accessing the portal. For instance, when a number of traffic streams each with different block sizes are passing through the portal, the streams with smaller block sizes will experience a varying queueing delay in accessing the portal. The spread in delay will depend on the size of the block used by other streams, the block receiving



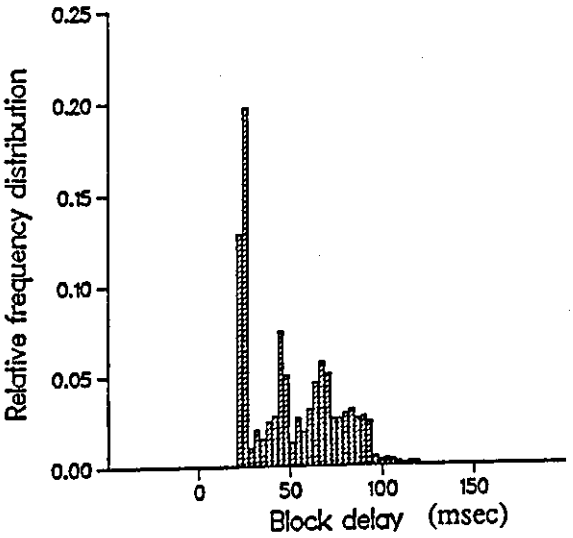
(a) TGR Traffic only



(b) TGR traffic with one video connection



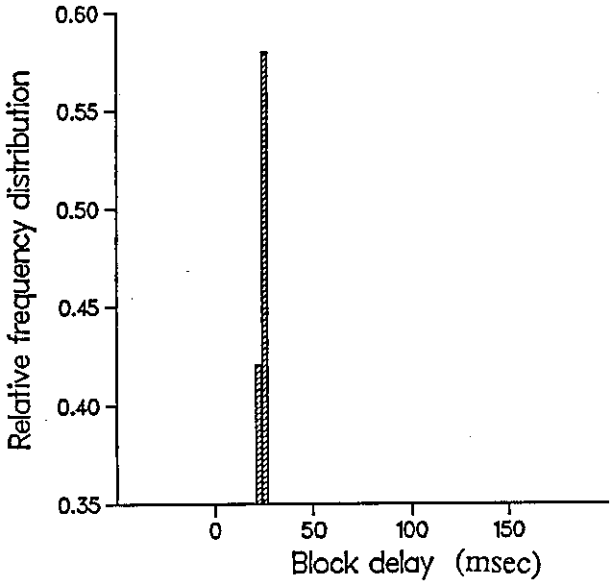
(c) TGR traffic with two video connections in opposite direction



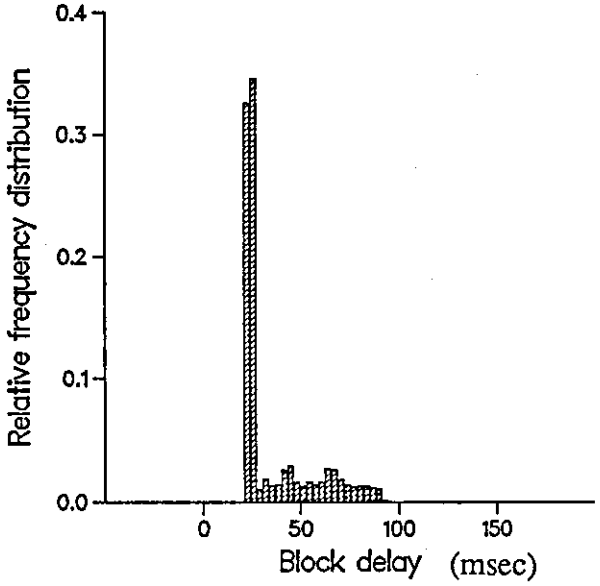
(d) TGR traffic with two video connections in the same direction

One video connection load = 5 blocks/sec
Video block size = 2048 bytes

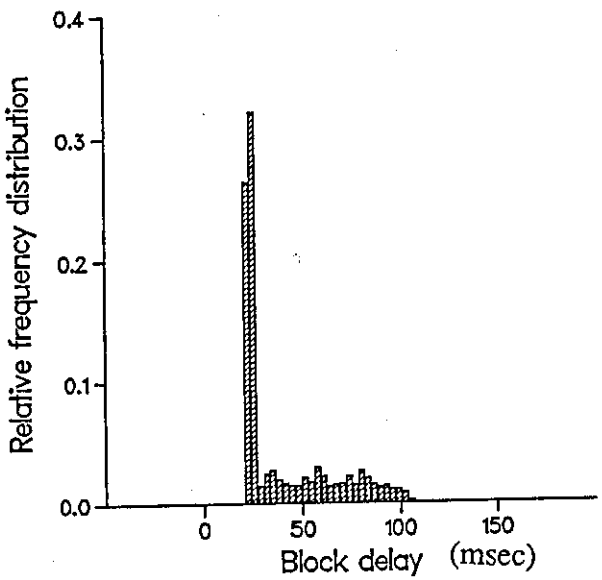
Fig. 4.15 Relative Frequency Distribution of Speech Blocks delay (milliseconds) at Various Loading Conditions



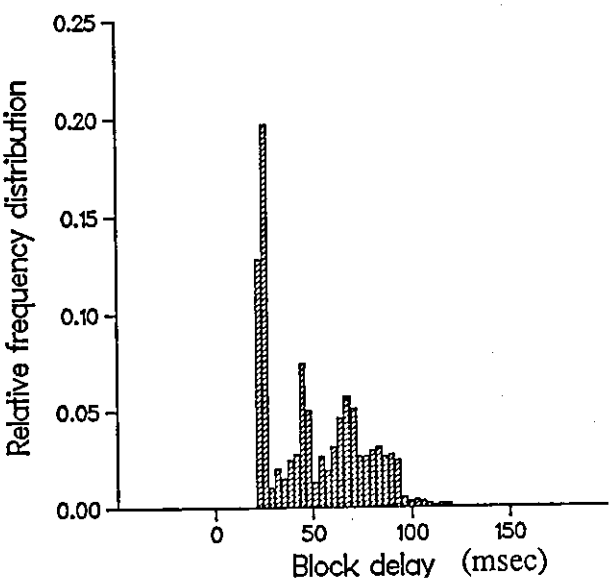
(a) TGR Traffic only



(b) TGR traffic with one video connection



(c) TGR traffic with two video connections in opposite direction



(d) TGR traffic with two video connections in the same direction

One video connection load = 5 blocks/sec
Video block size = 2048 bytes

Fig. 4.15 Relative Frequency Distribution of Speech Blocks delay (milliseconds) at Various Loading Conditions

or transmitting speed of the end user machines and statistical loading of the portal itself. Furthermore, the portal switches traffic on a first-in-first-out basis. Again, in an integrated services environment, the application using smaller block sizes can experience a large spread in block delay epecifically in heavy loading conditions. This delay will be due to the buffers provided in the portal in order to cache sudden traffic surdges.

Some experiments were also carried out to determine the performance of an emulated video traffic stream in the inter-site network configuration. In the CR based Unison network, the real video traffic sources generate traffic of the Constant Bit Rate (CBR) type, without any image compression [Murphy 89]. The CBR video traffic can be easily emulated by the TGRs. In general, the area of a video frame to be transmitted comprises p pixels (in the x direction) and l lines (in the y direction). Each pixel is coded as 16 bits (2 bytes) in the Unison system. The quantity of data to be transmitted will be $p * l * 2$ bytes where p and l determine the size of the video frame. For a video frame size of 512 pixels by 512 lines, the data for one frame will be 524288 bytes. In the Unison video frame stores, each line of a video frame requires two bytes of addressing information specifying the line number relative to the area origin. In the TGRs, the addressing overhead is ignored since it would be negligible for large video frames. The number of basic blocks required to transmit a video frame of 524288 bytes is around 256 blocks if the block size is 2048 bytes. This block size has been used for transferring video data between peer video traffic sources. The intersite video experiments have shown that the maximum update rate achieved for a frame size of 512 pixels by 512 lines is around 1/8 frames per second. In the TGR, the inter-block delay is used to control the requested frame rate - a parameter which can be selected at the start of the test. At the end of the test, the TGR shows the achieved frame rate which can be different from the requested frame rate. Block sequencing is used to detect lost blocks; each block in a video frame is given a sequence number which is checked at the TR.

Experiments were also carried out to determine delay variation for the emulated video traffic stream in the intersite network configuration. Real video application traffic sources were used to load the intersite communication chain. For a frame size of 128 pixels by 128 lines and a frame rate of one frame per second, the average intersite block delay is typically around 85 milliseconds without any other application traffic load. Table 4.1 shows average block delay for the emulated video blocks under

Intersite Traffic Loading	Average Delay (msec)
TGR traffic and one real video connection (in the same direction)	87
TGR traffic and two real video connections (in opposite directions)	88
TGR traffic and two real video connections (in the same direction)	103

Table 4.1 : Typical intersite block delays for a video traffic stream under different loading conditions

different loading conditions. The traffic load generated from a real video connection was roughly 8 blocks/sec.

4.7 Summary

In this Chapter, the main building blocks of the measurement system and their functional details are described. The measurement system demonstrates a practical approach to investigate performance issues related to experimental networks. Some of the issues related to the performance of the CR based Unison network have been addressed. As expected, the performance analysis has revealed idiosyncrasies in the behaviour of some of the network components such as the CR-CFR portal. The results over different parts of the network clearly show the portal to be the critical component in most situations. The throughput-delay characteristics of the CR-CFR portal are seen to be unacceptable under some loading conditions. This has been found to be due to the software operation of the portal and the low level access protocol of the CR.

Chapter 5

Transition to CFR Based Unison Network

5.1 Introduction

The introduction of the Cambridge Fast Ring (CFR) as the client LAN in the Unison network opened up a new area of research into network architecture and network application components in order to offer improved network performance for the applications. This also provided the opportunity to extend the measurement work to the CFR based network in exploring the practical performance issues in the new environment.

This chapter describes the design and implementation of a measurement system used for performance analysis on the CFR based Unison network. The methodology in developing the new measurement system is the same as that described in Chapter 3. A number of high performance traffic generators (TGRs) were built using transputer technology. The use of the measurement system has enabled a number of performance experiments over the new network to be carried out. In general, the experiments investigated the performance characteristics of the network switching components and studied the network performance effects at the network to

applications interfaces. However, some experiments were also conducted to test and study the performance characteristics of the Unison Multicast Server. These experiments are discussed at the end of this chapter.

5.2 CFR based Unison Network

Topologically, the CFR based Unison network is the same as the CR based Unison network. However, with the introduction of the CFR as the client LAN, a number of new developments have been made which make the Unison network architecture more suitable for multi-media communications. The main features of the CFR based network are summarised below.

- The network transports user information in fixed-size small packets, i.e minipackets. In the CR based network, the user information was carried in basic blocks from the application interfaces to the portals and vice versa. However, the communication between the peer portals (via the exchange rings and ramps) occurred in the form of minipackets. The CFR based network carries user data in the form of minipackets right from the application ends and provides a unified way of transporting user traffic. The new architecture has opened up new research dimensions towards the realisation of broadband ISDN based on fixed-size packet switching. The question, however is whether such a novel architecture could fulfill all the performance requirements imposed by the multi-media components.

- The CFR based network incorporates a novel minipacket switching device which enables the CFR client networks to be connected to the Unison network. This is referred to here as a CFR-CFR portal. The performance of the CFR-CFR portal can be extremely important in a mixed traffic environment.

- The network incorporates a new protocol architecture for connection setup and data transfer. These protocols have been exclusively developed for the CFR based Unison network and no effort has been made towards the compatibility between the CR and CFR based network protocols, i.e., the CR based applications cannot talk to the CFR based applications and vice versa. The connection set up between peer applications is achieved via the

secretary which replaces the function of the nameserver of the CR world. A Remote Procedure Call (RPC) mechanism has been employed in setting up associations between various services[Hamilton 84].

For data transfer, the applications are required to provide the application data in the form of minipackets at the network/application interface. Most of the real time applications, however, would prefer to use the minipacket layer for communication since the processing overhead in forming big blocks if higher layer protocols are incorporated would be avoided. However, a Unison Data Link (UDL) protocol has been defined which resides over the minipacket protocol. The UDL protocol is very similar to the basic block protocol used in the CR based Unison network. This protocol is quite lightweight in the sense that it does not provide any loss or error recovery, etc.

- A multicasting facility is provided in the CFR based network. This facility enables the application data to be replicated to a number of different sites. This would be useful when a conference involves more than two sites. It would seem sensible that data should be transmitted to some form of agent, which then takes the responsibility for replicating the data stream to each of the destination sites.

5.2.1 CFR as a client Network

The CFR is a high speed local area network, developed at the University of Cambridge Computer Laboratory as a successor to the CR (the medium speed local area network). As its name suggests, it is faster than its predecessor. The clock speed for the CFR is currently around 45 Mbits/sec. The CFR is made up of two types of nodes, stations which transfer data between devices attached to the ring, and a monitor which is required on each physical ring to set up and maintain the slot structure. The CFR access protocol is essentially the same as the CR. However, there are some additional features which make it more attractive than the CR.

Apart from the difference in speed, the CFR differs from the CR in a number of important aspects. The slot size has been increased from 40 bits to 304 bits. The format of the CFR slot is shown in Fig. 5.1. The data field has been increased from 2

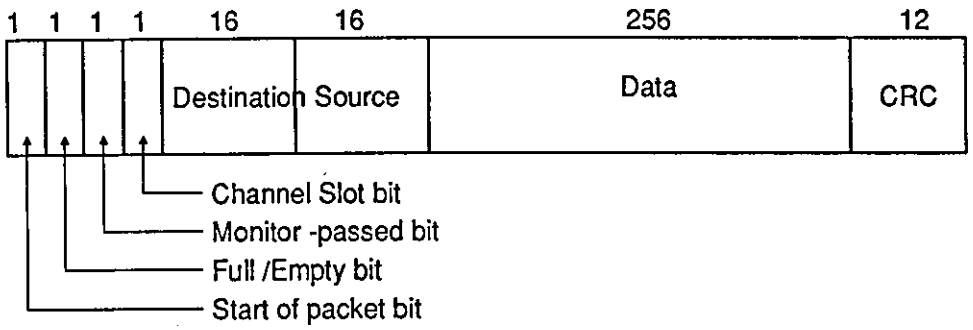


Fig. 5.1 Minipacket Format of the Cambridge Fast Ring

bytes to 32 bytes. This has resulted in a considerable reduction in the control overhead in transferring the user data across the ring. As a result of this change, the point-to-point bandwidth between a pair of station has increased along with an overall increase in the system bandwidth. The CFR access mechanism also differs slightly. In the CFR, there can be two types of slots on the ring, i.e. a normal slot and a channel slot. These slot types are distinguished from each other by a special bit in each ring slot. Only a slot with this bit set can be used in channel mode and the number of such slots are controlled centrally. The difference in the use of these slots is that a channel slot may be re-used by the source station while the normal slot must be passed on to the downstream station. Channel slots are especially suited for large bandwidth or bursty sources. However, a fair access to the medium is guaranteed by using the normal slots. Further details about the CFR can be found in Ref [Hopper 86].

5.2.2 Connection Set up

There is a hierarchy of management services which allows a client A to connect with a service B on the network. The most commonly used service is the secretary service which interacts with the clients and servers for connection establishments. This section briefly describes how a connection is set up between a client and a server.

Each client ring on the Unison network has a secretary. All the hosts (clients or servers) registering themselves on the ring at 'start of day' establish S-associations

with the local secretary indicating that the hosts are up and running. It is on this S-association that the clients later communicate with the secretary to offer any service on the network or to ask the secretary to create an association with another service on the network. In order to establish an association, the client initiates the action by sending a RPC call to the secretary through the existing S-association. In this RPC call, the client provides the service name to which it wants to connect and also provides a private port to which the server can reply to the client. It is now the secretary's responsibility to find out where that service resides on the network. If the service is on the same client LAN, the secretary sends a RPC call to the server (through the server's S-association) together with the client address and private port. The server either accepts the RPC call by giving a private port to which the client can communicate or rejects the call if it does not want to talk to the client by sending a negative reply to the secretary. The secretary then replies to the client either by providing the server's address and server's private port for the association set up or by providing a negative answer indicating the server's refusal for association set up. The client and server now know the station addresses and private ports which they can use in transferring the user data. In the case when the server is on another client LAN at the same site or distant site, the secretary communicates with the peer secretary requesting it to locate the service. The peer secretary will locate the desired service and perform the same sequence which the local secretary would have done if the server was on the same LAN. The peer secretary replies to the local secretary and this reply will be passed on to the client. The secretary serves two functions: first it locates the service and second it provides a direct link (making a path through bridging components) used later by the clients and servers for actual data transfers.

The association establishment process, although quite complicated, is out of band when user data is being transferred. The intention was to keep the control and actual data transfer separate. Once the association has been set up, there are no restrictions on the upper layer protocols that can be used over the association.

5.2.3 Priority Traffic handling

In the CFR based Unison architecture, it was decided to provide a two-level priority scheme in the network switching components. Priority packets get preferential treatment in the switching components in terms of delay and congestion control. Only low bandwidth applications, such as voice, were considered as the potential candidates

for priority treatment. The traffic priority is fixed on an association-specific basis at association establishment.

5.3 CFR Based Measurement System

5.3.1 The Design

The first consideration in designing the measurement system was to decide which machine should be used for traffic generation. In the CFR environment, there were two development approaches to interface user equipment with the high speed LAN. Cambridge University and Olivetti Research Laboratory were working together on 68000 based systems for developing the CFR ring driver code and the Unity RPC package to be used for workstation and workstation based applications. The workstation based applications include a remote filing system, document scanning system, and black and white video transmission system. RAL and LUT were working on Transputer based systems aimed at achieving similar functionality to be used for real-time application components and switching devices. The application components were telephone systems, wideband speech systems, and colour video transmission systems and the switching devices were the CFR-CFR portal, the multicast data server and the ramp. The important difference in these developments was that the 68000 based system was using UDL and RPC protocol layers for transporting application data whilst the transputer based system was also offering the user an access to the minipacket protocol layer on top of UDL and RPC protocols.

The transputer based system was selected due to the following reasons.

1. The network transports user data in minipackets which means that the bridging components were behaving as minipacket switches, so it would be wise to measure network performance in minipackets. Furthermore, the Unison multicasting machine used the minipacket layer to replicate user streams which means that the measurement system can also be used to test the muticasting machine.

2. Most of the application components were using the minipacket protocol layer for data transfer. The use of the minipacket protocol suits real time applications since the processing overhead involved in forming the bigger blocks can be avoided. However, for non-real time services, such a light weight protocol may not be

suitable and heavy weight protocols have to be used to provide error free data transfer. Since the main goal was to determine the network behaviour at the network/user interface for a variety of applications, the transputer based system provided the right access to carry out measurement at the lowest layer, i.e. the minipacket layer.

3. The transputer based system provided the processing power and a simple mechanism for extending the processing power (by virtue of the parallel architecture of the transputer) that would be required if complex traffic generating functions were to be supported. Furthermore, processing at the minipacket level itself demands more processing power in the first place.

5.3.2 The Control and Statistics Collection

The other consideration in designing the measurement system was to decide which machine should be used for control and statistics collection. The workstation was one of the obvious candidates for providing the control and statistics collection facilities [Murphy 89]. However, the use of the workstation would involve the development of code on both sides: the TGR side has to provide an RPC level interface to interact with the workstation and similar code on the workstation side would need to be provided. This also means that some understanding of the Tripos operating system would be essential to develop user friendly interface software on the workstation side. Fortunately, for control purposes, general purpose interface software was available on the workstation side that enabled the TGRs anywhere on the network to be controlled. In order to interact with the workstation, RPC level handshaking software was written on the TGR side.

An easier option would be to use a microcomputer which can be attached to one of the TGRs for control and statistics collection. This option was much easier to implement and was adopted as a result. The microcomputer provided control and collection for frequent experimentation while the use of workstation was restricted only to specific performance experiments.

5.3.3 Transputer-based TGR

5.3.3.1 Hardware

The TGR uses a VME-like architecture and contains a transputer based Single Board Computer (SBC), a CFR station card, a Rugby time clock receiver and an interface board between the clock receiver and the SBC (see Fig. 5.2). The SBC contains a T414 transputer chip, 1 Mbytes of RAM, 128 Kbytes of Eprom memory space and the CFR MAC-layer interface circuitry which maps the CFR station registers onto the SBC's memory space.

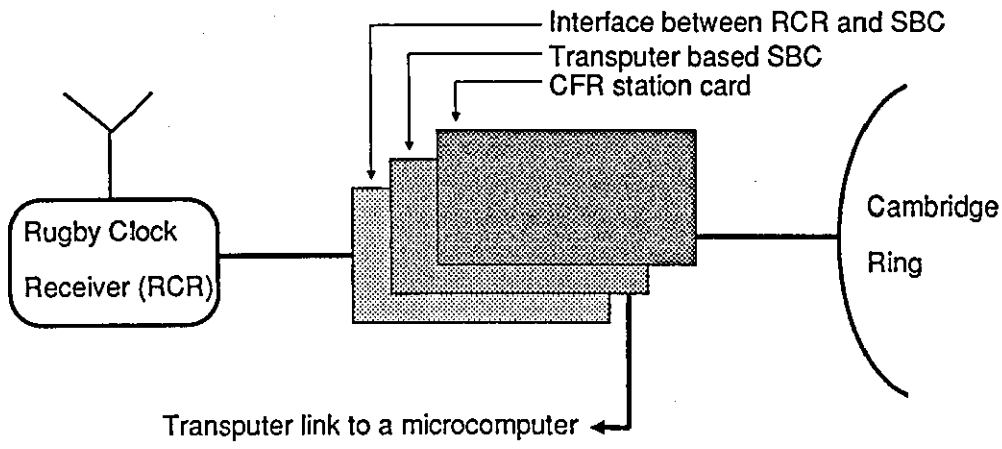


Fig. 5.2 Basic Components of a transputer based TGR

The SBC and CFR station card were commercially available (from Caman Systems Ltd.). The Rugby clock receivers were the same as used in the CR based Unison network. An interface board between the Rugby clock receiver and the SBC was developed enabling a chronological timestamping facility to be incorporated within the CFR based measurement system. The circuit diagram for the interface board is shown in appendix 2.

A microcomputer was interfaced to the traffic TGR via a transputer communication link. A general purpose interface card (Sension card) was commercially available which provides a byte-wide serial communication facility between the microcomputer and the transputer link. The microcomputer is used for control and statistics collection purposes.

5.3.3.2 Software Development Environment

The software for the TGRs was developed in Occam - a concurrent programming language [Inmos 88a]. The Transputer Development System (TDS), which was based on a stand alone microcomputer with a BOO4 single board computer and the development software, was used to develop the software for traffic generation [Inmos 88b]. A developed program can now be easily loaded into the SBC of the TGR via the transputer link. If the TGR is connected to the CFR, a simple display/keyboard control process running in the TDS enables the functionality of the developed program to be tested interactively over the network. Once the program is fully debugged and the desired functionality is achieved, the program can be blown into EPROMS to be used for frequent experimentation.

Unfortunately, there was no facility for booting the TGRs via the bootserver. The Unison bootserver facility was used to boot various machines across the network. The use of such a facility enables the old version of the software to be updated without blowing the EPROMS for every update. The bootserver facility was available to the ARM and 68000 series machines but not for transputer based systems, as the necessary boot protocol was never implemented.

5.3.3.3 Software Structure

The software comprised of a number of processes. The communication between processes occurs via the software and hardware transputer channels. The schematic diagram showing the main processes is shown in Fig. 5.3.

Timestamping Process

This process provides the chronological timestamping facility for one-way packet delay measurement across the network. The timestamping data includes second counts and microsecond counts. On the transmit side, the minipackets are timestamped with

these counts when they are launched on to the CFR CMOS chip for transmission. Similarly, on the receive side, the minipackets are timestamped when these are received from the CFR chip. Once the minipackets are timestamped, they are passed to the rx.handler process for further analysis. This process also communicates with the Rugby clock interface via a hardware channel to receive the time signals for synchronisation.

CFR.Interface Process

This process handles the low level handshaking signal from the CFR station chip to transmit and receive minipackets to and from the network. The receive and transmit processes run in parallel : the transmit process is interrupt based while the receive process is based on polling. These processes also record the low level statistics such as retransmissions, etc.

RPC process

The code for the RPC process was mainly developed at RAL. This code is responsible for setting up connections with the secretary and peer TGR across the network. Although the RPC process is quite big and complicated, it does not effect the traffic generation performance of the TGRs. Several low level processes form the RPC process, i.e, RPC Stub process, S-handler process, Sec.in process, Sec.out process, Rx.split process, Init process, etc. Their main function is to construct UDL blocks and RPCs to be used for association set up, to establish S-associations with the secretary using RPC and to refresh the S-association periodically with the secretary.

Tx.Handler process

This process contains the software for traffic generating functions. These generating functions are described in details in Appendix 1. The timer process is used to put delays between successive minipackets and the distribution of the inter-minipacket delay determines the patterns for traffic generation. The timer process uses the transputer on-board hardware clock for reading clock tick counts. The minipackets are properly sequenced by stamping them with sequence numbers. This process communicates with the CFR.Interface process for minipackets to be launched onto the network.

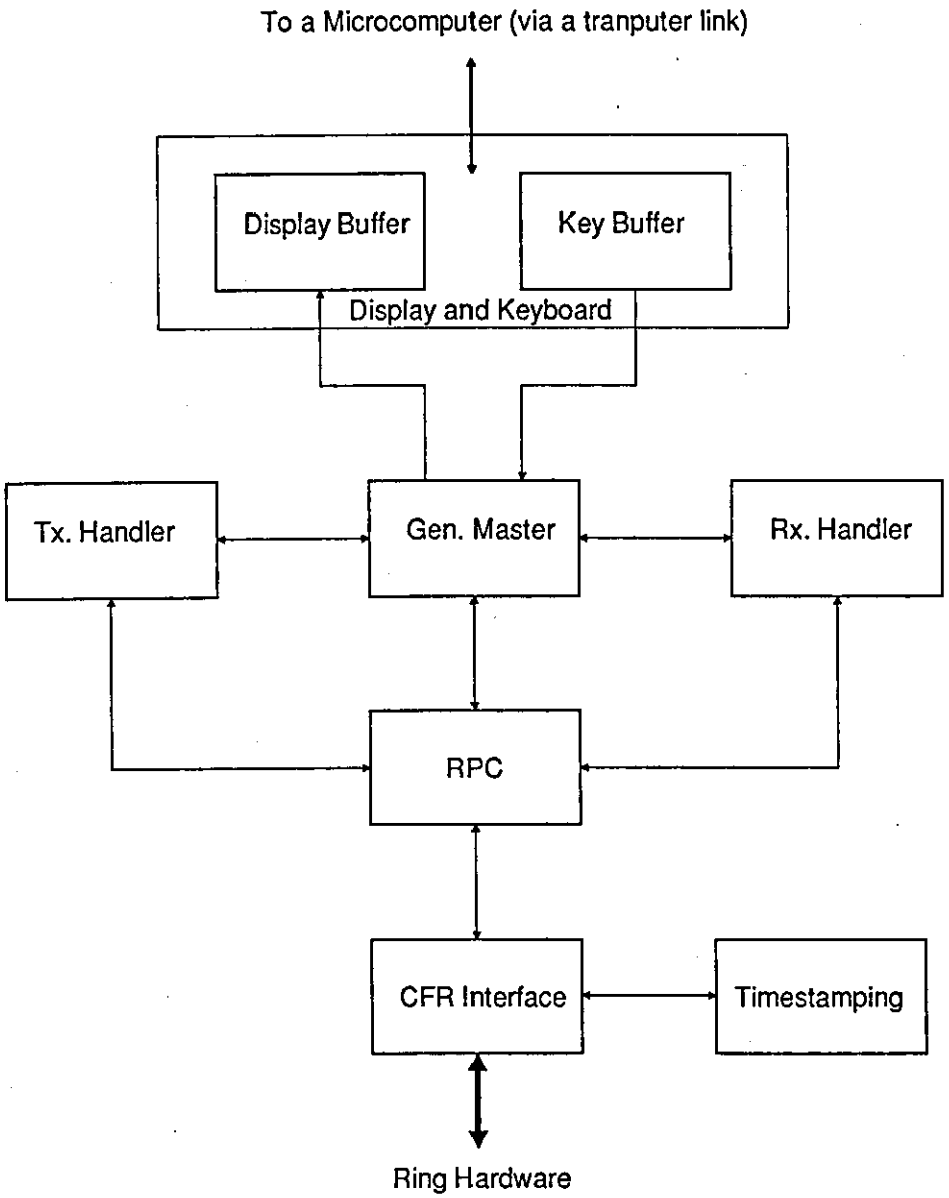


Fig. 5.3 Software Structure of the TGR (arrows between processes show the direction of software channels for inter-process communications

Rx.Handler

This process receives minipackets from the CFR.Interface process and accounts for the traffic statistics for a number of traffic parameters. It prepares statistics based on histograms for one-way minipacket delay, minipacket inter-arrival time, minipacket loss, etc. Also it records any sequence errors which occur due to minipacket loss or minipacket duplications. This process also interacts with the Gen.Master process to pass on any control information.

Gen.Master Process

This process provides the command interface between the controller station and the TGR. The commands are decoded and actions are initiated by relaying the control information to the appropriate target processes. For instance, if the controller station asks for the minipacket delays to be recorded for a sample of 2000 consecutive minipackets starting from the 10th second, the Gen.Master process will direct the RX.handler process to record the desired statistics at the desired time. This process interacts with the RPC process in order to make an association with the peer TGR. When the workstation acts as the controller station, this process provides the necessary mechanism for having multiple associations, i.e. with the workstation and the peer TGR. The Gen.Master process also implements the clock synchronisation protocol as defined in Chapter 3.

Display and Keyboard Processes

When a microcomputer attached to the TGR is acting as a controller station, the display and keyboard processes provide the communications facility between the microcomputer and the Gen.Master process. This communication occurs via a transputer hardware channel. The keyboard process receives the command in the form of keyboard codes when keys are pressed on the microcomputer. The keyboard process buffers these commands and relays them to the Gen.Master when Gen.Master is free (the keyboard process is the lowest priority process). Similarly, the display process serves to pass any information from the Gen.Master to the microcomputer for display or storage purposes.

5.3.4 Basic Operation

The operation for conducting a performance experiment is very similar to the one described in the previous chapter. The minimum requirement to perform an experiment is to have two TGRs connected to the LANs with a microcomputer attached to one. A workstation is required for certain experiments to act as a controller station. The traffic loop back facility provided in the network architecture allows traffic to be returned to the originating LAN. This enables any number of components to be selected in a communication chain to study the performance behaviour of the desired components. The operation of a simple performance experiment is described here to study the performance of the CFR-CFR portal.

The components involved in the experiment are shown diagrammatically in Fig. 5.4. The workstation, residing on the client LAN, is used as the controller station. A pair of TGRs is employed for generating traffic. A microcomputer attached to one of the TGR is used to collect the performance statistics. The application traffic sources (voice, video, computer file transfers) are used to achieve the experiment results in a mixed traffic environment. The dotted line shows the traffic path; the traffic originating on the client LAN is returned to the same LAN after traversing through the portal and the exchange ring.

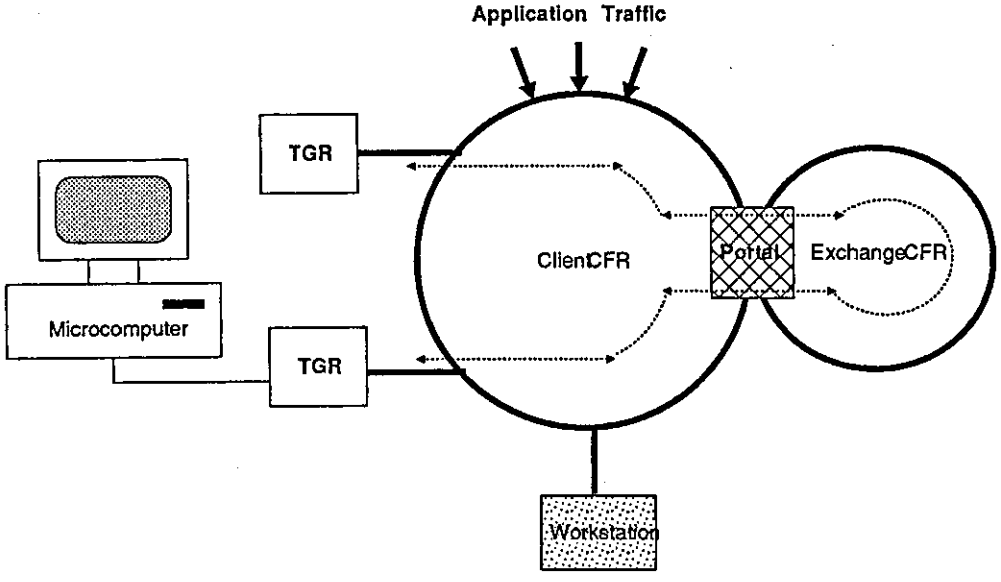


Fig. 5.4 Measurement System Components in a typical test configuration

When the TGRs are switched on, each one registers its services with the secretary that can be accessed on the network. The services include 'indus-c' which is used for control purposes and 'indus-t' which is used for traffic generation. The workstation, being the controller, initiates the action by selecting one TGR as the Traffic Generator (TG) and the other as the Traffic Receiver (TR). The workstation sends an RPC to the TG via the service 'indus-c', directing it to establish an association with the TR. The test parameters are also sent within the data part of the RPC. On receiving the RPC, the TG does a RPC with the secretary providing the name of the TR and the service name 'indus-t'. The secretary will locate and inform the TR about the TG's intentions. With the interaction of the TG, TR, and secretary, a direct path is set up which will be used for carrying data minipackets. The application components use a similar strategy for setting up the path on which the application traffic can flow.

Once the path has been set up, the TG generates traffic with the pre-defined traffic pattern for the specified duration. The same communication path is also shared by the other application traffic streams enabling the performance characteristics to be evaluated in a mixed traffic environment. At the end of the test, the TR and TG pass the performance results to the microcomputer for storage or display purposes. Some knowledge about the application traffic loading would be necessary in correlating the performance results achieved from the TR and TG.

5.4 Performance Measurement Experiments

The first obvious experiment was to determine the maximum traffic generation capability of the TGR on the CFR. Initial experiments regarding the traffic throughput indicated that the maximum generation rate was limited to only 3000 minipackets/sec. It was found later that the low level ring driver code was the limiting factor and it was re-written to achieve traffic throughput up to 10,000 minipackets/sec.

A number of experiments were conducted on the CFR portal. The measurement results have enabled the performance of the portal to be significantly upgraded. The next chapter discusses how portal performance has been improved and presents the final measured characteristics of the portal. The following sections describe the experimentation with other network components.

5.4.1 Experimentation with the Ramp

The CFR based TGRs can be connected to both types of CFRs, to the client CFRs or to the exchange CFRs. This enabled the performance characteristics of the ramp to be determined by placing TGRs on the exchange CFR. Four traffic TGRs were used to load the ramp with two traffic types (priority and non-priority). The traffic mix between priority and non-priority was kept at around 30:70. The ramp was configured in the loop back configuration which means that the same traffic stream would pass through the ramp twice.

The ramp throughput is a function of the ISDN bandwidth. The results have shown that the ramp can support around 198 minipackets/sec per ISDN slot (one slot = 64 Kbits/sec).

The minipacket delay across the ramp will depend on the following parameters:

Total Minipacket Delay (T_d) = P_d + ((F_t * M_s) / N_s)+ Q_d + Δ_j

Where

P_d is processing delay (microseconds)

F_t is one frame time of (125 microseconds)

M_s is minipacket size (minipacket size when launched over ISDN is 40 bytes)

N_s is number of available ISDN slots

Q_d is queueing delay (microseconds)

Δ_j is delay due to jitter (microseconds)

As can be seen, the delay across the ramp is function of the ISDN bandwidth. Δ_j is of the order of 500 microseconds (four frame times) since the ramp ISDN implementation operates in 4 byte wide quad frames [Burren 89b]. P_d depends on the processing speed of the ramp and is found to be 4200 microseconds (two-way). When the ISDN bandwidth is one slot (N_s=1), the minimum two-way delay (M_d) across ramp would be

M_d (microseconds) = 4200 + (125 * 40)

The minimum delay across ramp at various ISDN bandwidths is shown in Fig. 5.5. The graph shows that the minipacket delay remains almost constant as long as the ISDN bandwidth is 10 slots or above. The delay increases by 0.5 millisecond when the ISDN bandwidth is between 9 slots and 5 slots. Then it starts gradually rising and finally peaking at around 9 milliseconds when ISDN bandwidth is 1 slot. The graph also indicates that if the ISDN bandwidth is greater than 10 slots, the increase in total minipacket delay due to the ISDN bandwidth limitation is comparatively negligible (total delay is approximately equal to P_d).

Fig. 5.6 shows the throughput-delay characteristics at typical ISDN bandwidths. The

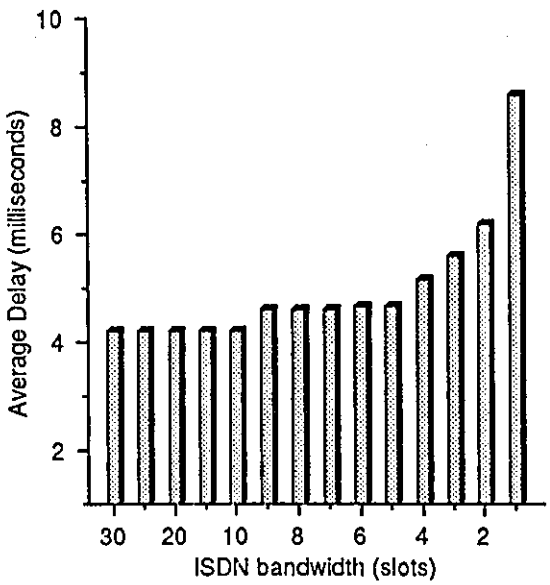
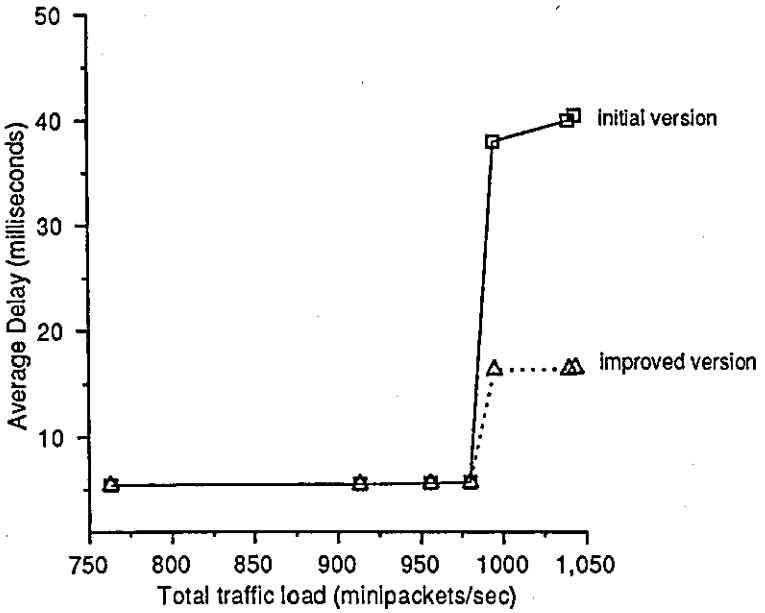
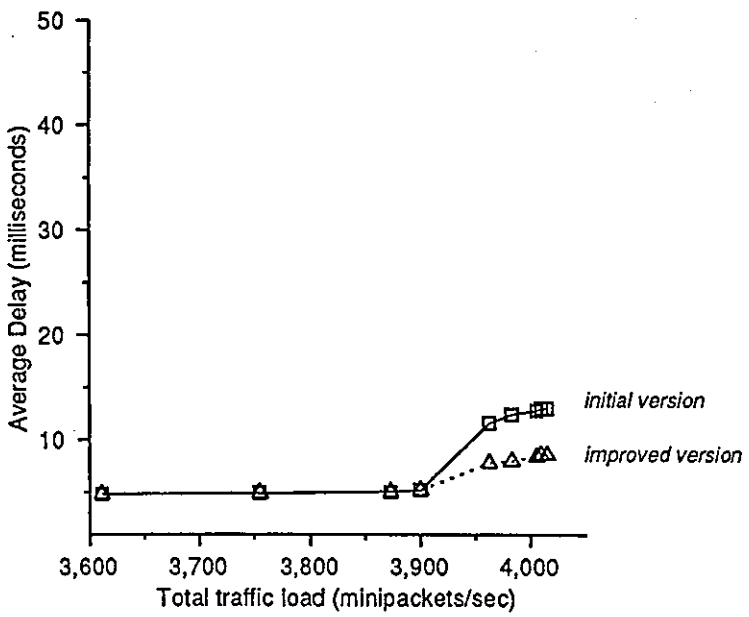


Fig. 5.5 Minipacket Delay across Ramp for various ISDN bandwidths (at low loading conditions)

graph shows that there is no increase in the average delay until a certain critical load above which it rises rapidly. The critical load is equal to 198 * n, where n is the number of ISDN slots. For instance, the critical load for 5 slots would be about 990 minipackets/sec. After passing the critical load, the delay does not increase with further increase in traffic loading. As can be seen in Fig. 5.6 (a) and (b), the rise is a function of the ISDN bandwidth. The smaller the ISDN bandwidth, the larger the ratio of rise. The reason for such behaviour is due to a fixed-size small buffer pool on



(a) ISDN bandwidth = 5 slots



(b) ISDN bandwidth = 20 slots

Fig. 5.6 Throughput-delay Characteristics of the Ramp at typical ISDN bandwidths

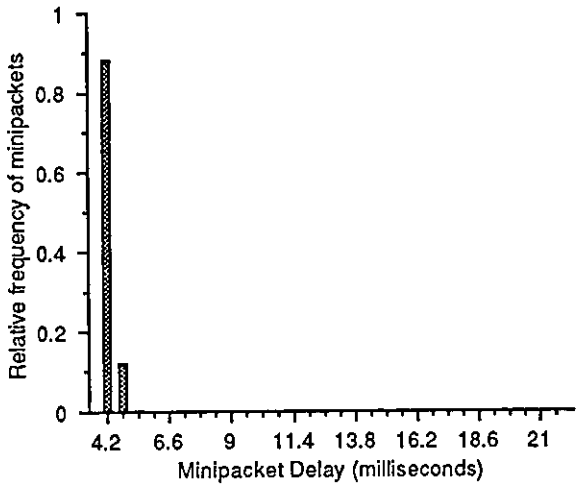
the ISDN side of the ramp which is shared by the priority and non-priority traffic streams. In heavy loading conditions, the priority traffic would experience larger delays when the ISDN bandwidth is smaller, and vice versa. The hard lines in the graphs show such behaviour from an initial version of the ramp software. Later, the ratio of rise was made almost independent of the ISDN bandwidth and was adjusted to a reasonable value by making the size of the buffer pool a function of the ISDN bandwidth. The dotted lines in the graph shows the delay characteristics from an improved version of the ramp software code.

Fig. 5.7 shows the relative frequency distribution of the minipacket delays at typical loading conditions. The ISDN bandwidth for the experiment was 20 slots. The graph shows that below the critical loading, there is virtually no spread in minipacket delay. However, at and above critical loads, the spread increases considerably.

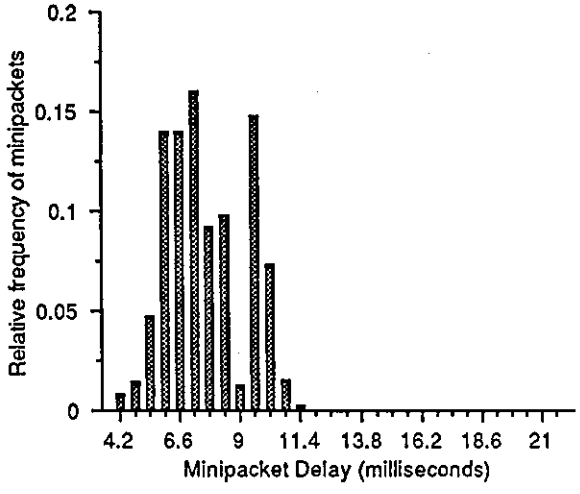
5.4.2 Inter-site Experiments

Inter-site experiments were conducted between the RAL and LUT sites, and for this, the inter-site ISDN bandwidth was increased to 25 slots. It was not considered appropriate to increase the bandwidth further since there was at least one slot needed in the mega-long loop back configuration for each site. Using this, each site can test their ramps independently for the purpose of local functionality testing.

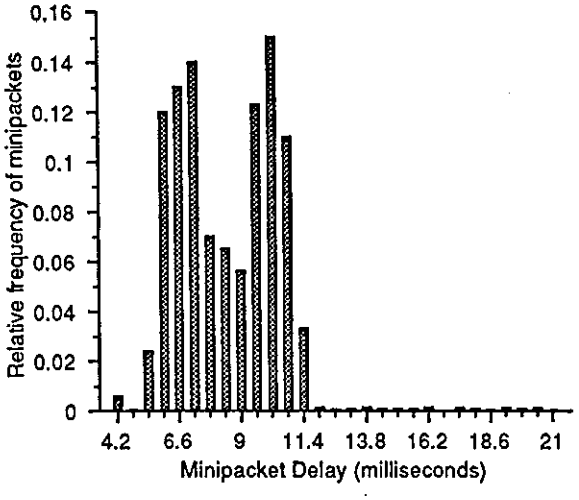
The purpose of conducting these experiments was to determine the performance characteristics of the inter-site communication chain under various loading conditions. In particular, the emphasis was to determine delay characteristics for a typical voice connection in various loading conditions. Unfortunately, there were no high bandwidth application traffic sources available which could be used for the purpose of extra loading and which would give some correlation of the test traffic with the application traffic. However, to overcome this problem, two TGR pairs (one pair at each site) were used; one pair was used to conduct the actual tests while the other pair was merely used for extra loading. In terms of inter-site traffic throughput, the experiments have shown that the inter-site communication chain can support 4950 minipackets/sec (two-way) of traffic without any traffic loss. Further loading caused traffic loss to occur. The limiting factor was the ISDN bandwidth, although the portal and ramp can support higher traffic throughputs (when tested locally). The inter-site minipacket delay at low loading condition was around 7 milliseconds.



(a) Below saturation (load = 3800 minipackets/s)



(b) Near Saturation (load = 3962 minipackets/s)



(c) Above saturation (load = 4025 minipackets/s)

Fig. 5.7 Relative Frequency Distribution of the Minipacket Delays across Ramp at Typical Loading Conditions

Fig. 5.8 shows the delay characteristics of a simulated voice traffic stream under typical loading conditions. Under light loading conditions, there is virtually no spread in minipacket delay and a high percentage of minipackets experience similar delays. When at high loading conditions, the spread had increased and the majority of the packets experience a fixed high delay. It is interesting to note that there are two peaks in the delay distribution. These peaks indicate that there are two operating points; most of the time the delay remains high (high peak), however periodically it goes low for a small duration (low peak) This problem was associated with the queueing mechanism

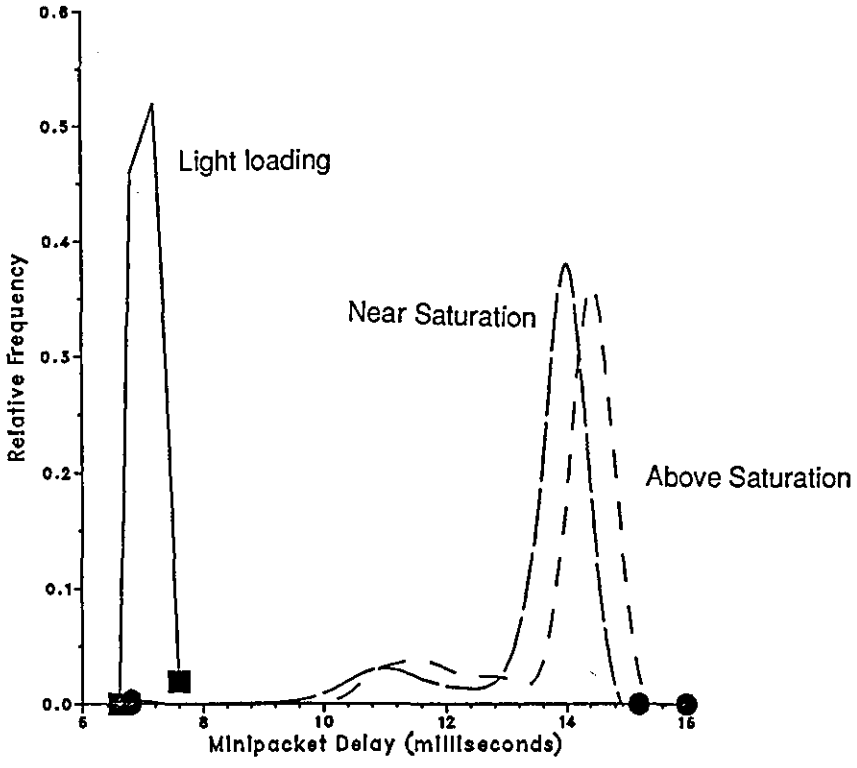
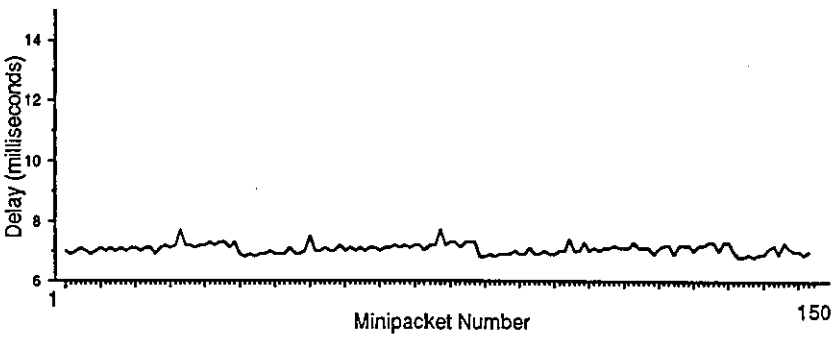


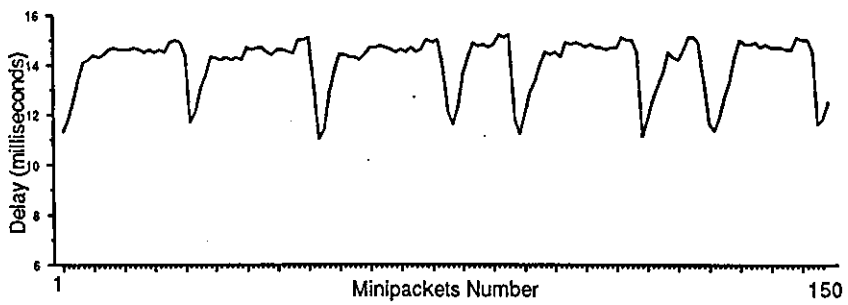
Fig. 5.8 Delay Characteristics of a Simulated Voice Traffic Stream at Typical Loading Conditions

in the ramp. It is worth mentioning here that the current portal code does not provide any buffering and relies on the backoff mechanism onto the CFR chip in the case of heavy loading (this is discussed further in the next chapter). In the ramp, as mentioned earlier, there is a small buffer pool on the ISDN side shared by the two traffic types. The size of this buffer pool is around 40 minipackets. Other than this pool, there are separate buffer pools for the priority and non-priority traffic (the traffic

received from the CFR is buffered into these pools). The common buffer pool takes priority traffic in preference to the non-priority traffic. In order to avoid overflow of the common buffer pool, the pool is served by two level pointers. For instance, if the common pool has become full, the buffer serving process would stop taking more traffic until the pool is 60% vacant. In heavy loading conditions, such a buffer serving mechanism would result in two above-saturation operating points. Fig. 5.9 shows delays for 150 successive minipackets of speech data packets in typical loading conditions. This graph also shows a similar phenomenon as explained above. This graph clearly shows that for most of the time, the minipacket delay remains high (14



(a) No loading

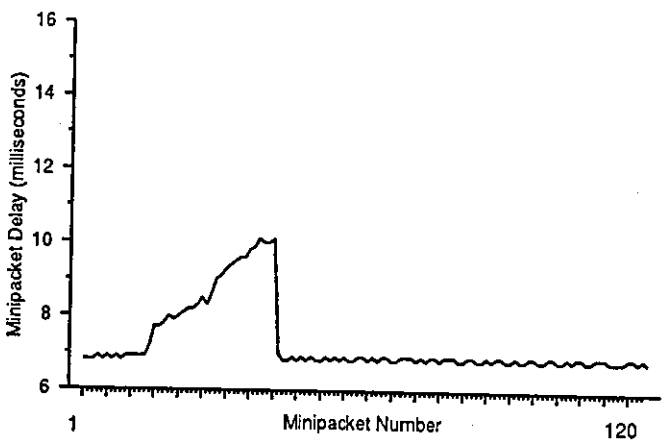


(b) Under heavy loading (4682 minipackets/sec)

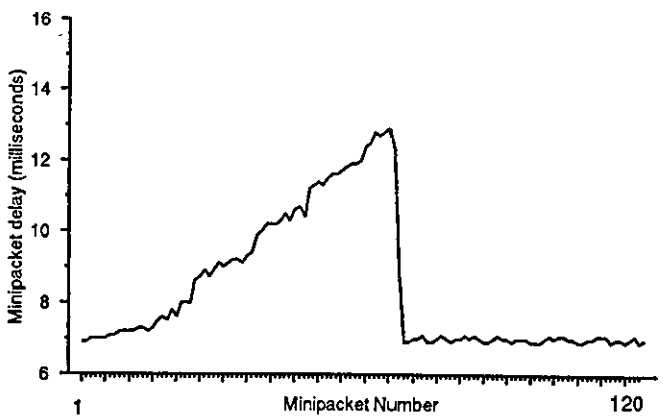
Fig. 5.9 Successive Minipacket Delay for 150 Minipackets of Voice Data at Typical Loading Conditions

milliseconds), however, periodically it goes low for a short duration. At low loading conditions, the graph shows an almost constant delay

Fig. 5.10 shows the delay characteristics of a simulated voice connection when the network is loaded with a burst loading function. The delays for successive 120 minipackets are show. Two typical burst sizes were used and burst intensity was selected to emulate a high loading condition during a burst. The graph shows that when the burst size is 500 minipackets, the delay rises to around 10 milliseconds during the burst. Once the burst is over, the delay pattern returns to normal. When the burst size is increased to 1000 minipackets, the delay rises to 14 milliseconds during the burst.



(a) Burst Size = 500 minipackets



(b) Burst Size = 1000 minipackets

Fig. 5.10 Delay Characteristics of a Simulated Voice Traffic Stream Under the Burst Loading Conditions

Fig. 5.11 shows the inter-arrival time distribution of voice packets under no loading and high loading conditions. The graph indicates that more than 90% of the minipackets observe no deviation from the mean inter-arrival time when there is no loading. When the communication chain is loaded 100%, the percentage of minipackets with no deviation reduces and one can expect a deviation of 3 milliseconds from the mean in either direction (positive or negative).

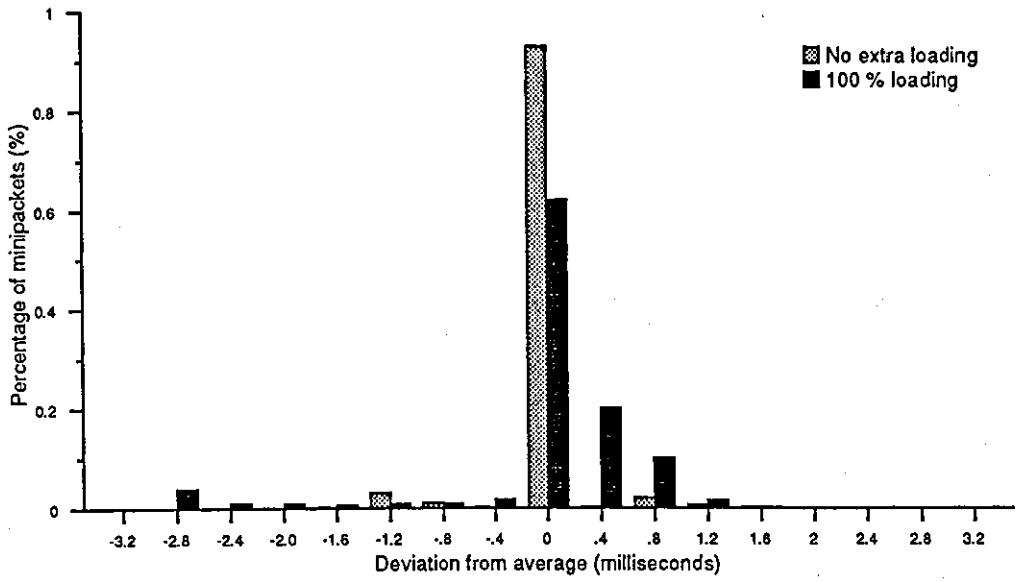


Fig. 5.11 Inter-arrival time distribution of Voice Packets under Various Loading Conditions

Table 5.1 shows the connection establishment times in various network configurations. This is the time taken by the network management (secretary, etc.) to set up an

Network Configuration	Connection Set up time (milliseconds)
Between Client CFR and Exchange CFR	1.84
Between two Client CFRs at the same site	2.6
Inter-site Without Secretary-to-Secretary Association	7.1
Inter-site With Secretary-to-Secretary Association	14.7

Table 5.1 Connection Establishment Times in Various Network Configurations

association between two peer TGRs. This gives a user an idea of how much time the user has to wait for a connection to be set up before actually transferring any useful information.

5.5 Experiments on the Unison Multicast Server

5.5.1 Introduction to Multicast

A multicasting service on a network offers a replication facility which is beneficial for certain applications such as multi-destination file delivery, distributed data base updating and teleconferencing. Such a service can be useful for two main reasons. Firstly, the multicast service will remove the need for source replication of data for multiple sinks. Instead, the replication will be done by the multicast service on behalf of the clients. Secondly, the bandwidth requirements can be reduced through critical sections of the network such as packet switches and inter-site communication links. For example, a traffic source at one site wishing to send data traffic to two sinks at a distant site need only send a single traffic stream to the multicast server at the distant site; the multicast server will then handle the local replication of the traffic stream. The following section describes some of the performance experiments conducted to test the performance of the Unison multicasting facility.

5.5.2 Testing Environment

The prototype Unison multicast system comprises a collection of two entities; the Multicast Data Server (MDS) which provides a many way data split facility and the Multicast Control Server (MCS) which provides the control aspects for setting up associations, etc. The multicast server provides replication only at the minipacket level and no support is provided for higher layer protocols such as RPC. Full details of the Unison multicast facility can be found in Ref. [Shrimpton 87].

The test configuration is shown in Fig. 5.12. All the components are connected to the client LAN. The MDS and MCS reside in a single machine, the multicast machine. To start with, three TGRs were used in the experiment; two of them were used as Traffic Receivers (TR) while the third acted as a Traffic Generator (TG). The software for the TG and TR was slightly modified to overcome problems regarding association set up with the MCS. A workstation was specifically required to act as a

controller station for the TG and TRs. To set up a multicast, the TG and TRs contacts the MCS requesting the creation of a multicast channel. The MCS contacts the MDS on behalf of the TG and TRs to set up the required data path. The TG acts as 'transmitter only' while the TRs are acting as 'receiver only'. Once the data path has been established, the traffic stream generated from the TG is replicated by the MDS to form two streams, one for each TR.

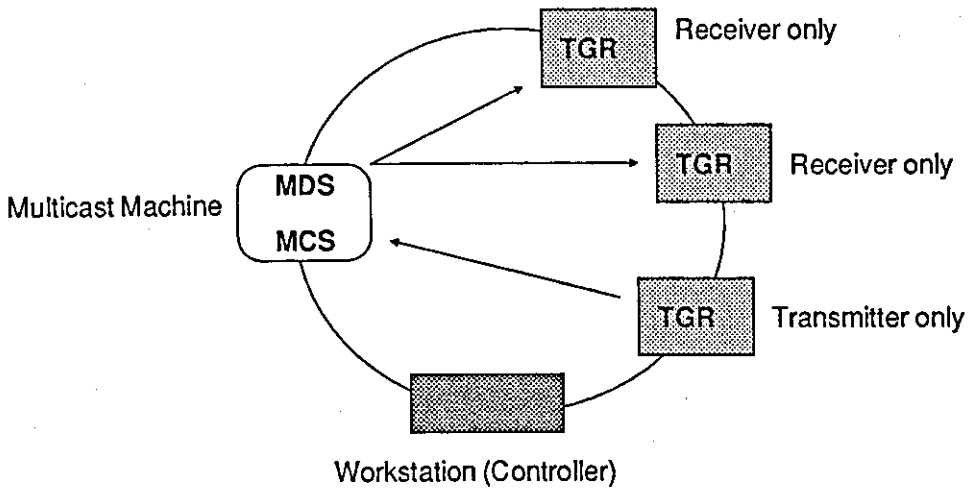


Fig. 5.12 Multicast Test Experiment Over the Client CFR

A number of performance tests were conducted to assess the performance of the multicast machine. By gradually increasing the load from the TR, the maximum one to two data split capability of the multicast machine was determined along with the evaluation of other associated performance parameters. Complex experiments were conducted later involving more TGRs to produce one to three and one to many data splits. The use of TGRs proved extremely useful in debugging the MDS software and in refining the performance of the multicast machine. These experiments provided the basis for the multicast machine to be used for real application components over the network.

Table 5.2 shows the throughput performance of the multicast data server over a client CFR.

Multicating	Throughput (Kbits/sec)
One to One	1640
One to two	1125
One to three	715

Table 5.2 Multicast Data Server Performance on the Client CFR

5.6 Summary

A measurement system has been designed, implemented, and used on the CFR based Unison network. The system has enabled performance experiments to be carried out with various network entities. The results achieved from extensive experimentation demonstrate the effectiveness and usefulness of the measurement methodology. The measurement exercise was specifically useful to the designers of the ramp and the multicast server. The measurement results have helped in debugging and improving the performance of these components.

Chapter 6

Improving the Portal Performance

6.1 Introduction

A naive designer may suggest that all traffic streams which pass through network switching components should be tested at various network points to see whether the traffic switching algorithms incorporated in the switching components have the desired effect on the traffic streams, and whether each stream is getting a fair share of the network resources. If performance of the network components is not as requested, investigation may be made in order to achieve the desired performance. Although it seems impossible to follow such an approach but that is exactly what had been done in investigating the performance of a packet switching device called CFR-CFR portal. The measurement system based on traffic generators allows such an idealistic approach to be realised to some extent. The traffic streams generated from the traffic generators can be thoroughly analysed to extract the necessary performance information which would be useful to the designer in adjusting various parameters in the portal if necessary.

The CFR-CFR portal was designed and developed at RAL [Siddiqui 89]. The development and investigation of the traffic switching algorithm was carried out by the author in a close collaboration with RAL. A number of experiments were

conducted to study the performance of the traffic switching algorithm. The experiment results were used in a 'Measure and Tune' fashion in order to achieve optimum performance from the portal.

This chapter starts with a brief introduction to the CFR-CFR portal which is necessary to understand the significance of the work presented. The traffic switching algorithm is described in depth, explaining the parameters which can be adjusted in order to achieve optimum performance from the portal. The chapter ends with a discussion of the throughput-delay characteristics of the improved portal.

6.2 The CFR-CFR Portal

6.2.1 Background

The measurement results shown in Chapter 4 indicate that the CR-CFR portal is the weakest link in the CR based Unison network. The main reason is that the CR portal does not possess any mechanism for performance controllability. As a result, the quality of service for certain applications is severely affected.

The CR portal provides switching at the block level and block size is variable. Due to some implementation characteristics (some of the characteristics have been explained in [Leslie 83]), the portal does not provide multiplexing for the transmission and reception of basic blocks. This has two consequences on the switch performance and user performance. Firstly, the useful throughput of the portal is greatly reduced especially when switching small blocks. Secondly, delay sensitive applications, such as voice which use small block sizes in order to reduce packetisation delay, are affected most when a number of parallel traffic streams with larger block sizes are passing through the portal. This is because the portal is tied up for much of the time dealing with large blocks. Very large blocks monopolise the use of the portal thus increasing access delay for other users.

The problem with applications which use small block sizes is magnified due to the fact that the portal switches traffic on a first-in-first-out basis with no class distinction. These applications can have an unacceptable spread in block delay resulting from queueing whenever there is an overload.

Generally, the switching devices which provide variable-size-blocks switching involve more processing overhead (which becomes quite significant for small blocks) in comparison with the devices which provide only fixed-size-block switching.

6.2.2 Design

The CFR-CFR Portal behaves as a minipacket bridge between the client CFR and exchange CFR. The portal was designed with three criteria in mind.

1. It should relay CFR minipackets between two CFR LANs as quickly as possible. Lightweight virtual circuits would be set up through the portals which would carry this traffic.

2. It should provide two classes of services.

    - A high priority service which implies that this traffic should experience minimum delay in the portal. This is taken to imply only minimal buffering and retry facilities.

    - A low priority service which requires that the portal should provide some buffering and retry facilities.

3. The network based on portal interconnections should scale. A number of portals with, perhaps, a number of simultaneous connections should be possible in the network.

The portal is built from two T414 transputer systems, each driving the CFR on its side through a CFR general purpose interface. The minipacket traffic between the transputer systems is carried through transputer links and the traffic streams for both directions pass through the portal on separate transputer links. The present state-of-the-art sequential processors are limited by the maximum clock rate from providing faster processing in time critical components. The transputer systems offer flexibility and simplicity in adding extra processing power to reduce the portal's processing time.

Packets on lightweight virtual circuits are directly addressed to the portal, each packet containing a destination address and a port number. The portal provides the mapping between the two-tuple [portal-address, portal-in-port] and [destination address,

destination-in-port]. The portal itself selects the portal-in-port and the network management sets up the mappings using the Remote Procedure Call (RPC) mechanism. The RPC mechanism, although fairly complicated, is out of band and does not seriously affect the performance of the established lightweight virtual circuits.

The portal switches user data in fixed-sized minipackets, thus providing an application independent service to various applications. The applications can use any packet size depending upon the nature of the applications, but at the network user interface, all applications are required to provide the application data in the form of minipackets. The packet formation overhead is at the application end rather than at the network bridging components. This has resulted in a much simpler traffic switching mechanism in the portal.

6.3 The 'Measure and Tune' Technique

Measurements serve a special purpose when these are carried out on an experimental network. The network designer can use the measured statistics to adjust various parameters (related to the traffic switching algorithm) which may enhance the overall network performance. This strategy was used to improve the performance of the portal.

The traffic generators are capable of producing repeatable traffic patterns. By varying the parameters in the traffic handling mechanism of the portal, the repeated measurement experiments help to investigate the comparative performance improvements. Also, the use of this technique enables the designer to observe a components behaviour in an overload condition which is difficult to predict when analytical or simulation techniques are used for performance analysis.

The following sections describe how the 'measure and tune' technique has been employed in a systematic manner to study the traffic handling mechanism of the portal.

6.3.1 Traffic Handling Mechanism

Ideally, the switching components should provide infinite throughput and zero switching delay in transferring the user data across the network. Not surprisingly, no switching device is close to this goal. But the degree to which a particular device falls

short of this idealistic goal is of great interest, and it is by comparing these shortcomings that the relative merits of these devices can be assessed. The traffic handling mechanism incorporated in the bridging devices plays an important role in determining the performance of a device.

An important consideration in designing a traffic handling mechanism is the buffering and retransmission strategy for different classes of application traffic. Buffers are required in the bridging devices to overcome the effects of occasional traffic surges. Retransmissions are useful when the end machines are slow to receive data traffic or when the data is corrupted on the way to the end machine.

6.3.1.1 Buffering

In packet-switched networks, the incorporation of buffers in the switches is an essential measure to overcome traffic overload conditions in the switches. An important consideration for the provision of the buffers is the selection of the size of the buffers and their management.

In the CFR based Unison network, user data is transported in small fixed-size packets, it is therefore natural to organise the buffers as a pool of identical size buffers, with one packet per buffer. However, if the packet sizes were different, as was the case in the CR based network, a pool of fixed-size buffers would present problems. If the buffer size is chosen to be equal to the largest possible message, space will be wasted when a small packet arrives. If the buffer size is chosen to be less than the minimum packet size, multiple buffers would be needed for long packets, with the attendant complexity. Effective buffer management involves the efficient use of buffers while keeping the processing overhead in the switch minimal. If multiple priority levels are supported by the switch, then the number of buffer queues would normally equal the number of priority levels; one buffer queue for each traffic type. In the CFR based Unison network, two priority levels i.e. low priority and high priority, are supported. This would result in two buffer queues. However, if it is assumed that priority traffic load would normally be low then it may not be necessary to provide a buffer queue for the priority traffic. The priority traffic is then always served first and is passed straight across. In the Unison network, since only low bandwidth delay-sensitive applications are considered for priority treatment, the above assumption can be justified. However, there must be a limit on the maximum number of priority

connections that can be supported by a switch at any one time in order to ensure that the non-priority traffic is not overrun by the priority traffic. The above mechanism would result in a simpler queue serving process since only one buffer queue would need to be processed. If there are two buffer queues, complex but flexible queue serving mechanisms can be incorporated at the expense of extra processing overhead. These mechanisms can provide better performance controllability. Details on these mechanisms can be found in [Tanenbaum 88].

6.3.1.2 Retransmission

The automatic retransmission facility provided in the CFR chip is useful in two ways. Firstly, it is useful to the portal in the case when the portal is transmitting packets to a slow receiver or relaying traffic to another portal which is heavily loaded. A slow receiver can mark the 'try again' response bit for those minipackets which arrive while the receiver is still busy dealing with the previous packet. A few retransmissions are worthwhile in overcoming such problems. It is worth mentioning that these retransmissions are initiated automatically by the CFR chip without any interference to the portal, thus without incurring any processing overhead in the portal. Secondly, the CFR station chip provides the minipacket CRC calculation in hardware. This means that in the case of a CRC error, the receiver can use the 'try again' response bit to ask for retransmission from the transmitter. This is one of the advantages of slotted rings which release slots at the source because this provides an opportunity for the source to retransmit the erroneous minipackets immediately when a 'try again' response is received.

Retransmissions also have a bad effect on the portal's performance. Retransmissions block the passage of other packets waiting in the queue to be served, thus reducing the effective throughput of the portal and increasing the packet forwarding delay. The retransmission algorithm must compromise between not occupying the portal too long and yet being sufficiently prolonged to allow slow receivers to receive effectively.

6.3.2 Performance tests with the Portal

The configuration used for conducting performance experiments is shown in Fig. 6.1. Four traffic generators, two connected to the client LAN and two connected to the backbone LAN, were used to generate traffic streams in order to determine the

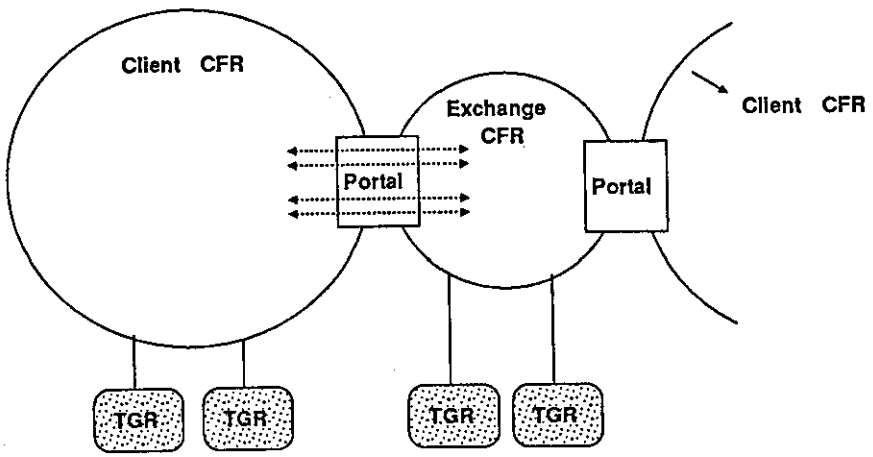


Fig. 6.1 Configuration for the Portal Performance Tests

performance characteristics of the traffic switching algorithms. The traffic mix between priority and non-priority traffic was kept at 30:70. This mix was purely arbitrary and experiments with other traffic mixes were also conducted.

The traffic switching code of the portal was developed in three stages. The results achieved from the measurement experiments enabled the performance characteristics of the portal to improve.

6.3.2.1 First Design Implementation

The first design of the traffic switching algorithm was implemented at RAL. In this design, only one buffer queue (First In First Out type) was provided for all traffic types. The rationale behind such a design was the simplicity in implementation and low processing overhead involved in serving only one queue. Since priority, non-priority, and control traffic were buffered in the same queue, the required quality of service for each traffic type cannot be maintained in heavy loading conditions. RPC based traffic requiring associations to be set up between clients is recognised by the portal as control traffic.

The retransmission strategy was also kept simple. There are two types of retransmissions used in the portal, software retries and hardware retries. The CFR

chip has an automatic retransmission mechanism and retransmissions carried out by the chip are called hardware retries. The number of these retransmissions can be adjusted as required by writing to the repeat control register. A software retry can be attempted when a minipacket has been Thrown On Ground (TOGed); after the maximum number of hardware retries has occurred, the CFR chip gives up and passes the TOG response to the Portal. In the first design, the number of retransmissions was kept the same for all traffic types.

The maximum two-way throughput (traffic passing in both directions) of the portal was found to be around 1200 minipackets/sec which amounts to a useful data rate of 260 Kbits/sec. The one-way throughput was found to be around 2000 minipackets/sec which was 40% higher than the two-way throughput. This immediately suggested that there was "hogging" between the transmit and receive processes. Similarly, the delay characteristics showed that under low loading (below-saturation), the minipacket delay was higher when the portal was loaded in both directions than when it is loaded in one direction only. In heavy loading conditions (above saturation), the minipacket delay was proportional to the number of buffers provided in the portal and the traffic loading itself. This was understandable because all traffic types were sharing the same buffer queue.

6.3.2.2 Second Design Implementation

In the second design, it was decided to use separate buffer queues for non-priority and control traffic, and no buffering was provided for the priority traffic. The buffer for the non-priority traffic is of an overflow type meaning that traffic will be thrown away if the buffer becomes full.

The buffer queues are serviced sequentially in the sending process; the priority traffic is served first and is sent straight across while the non-priority queue and association traffic queue are serviced afterwards in order. The queue serving process is kept simple so that there would be minimum processing overhead for switching the data traffic. One of the main advantages of splitting the traffic into different queues is that the software and hardware retries for each traffic type can be different. The retry value for the priority traffic can be selected to be a minimum to ensure that excessive priority traffic does not affect other traffic types.

A few modifications were also made to the low level transmit and receive processes. These modifications were very trivial. For instance, if the code of the transmit and receive processes is placed in the transputer on-chip memory, then run-time code performance is improved. This resulted in a slight increase in the throughput. Another modification was made to the transmit process in which a packet ready for transmission was launched immediately on the CFR chip rather than putting it first into a look aside buffer. Such modifications resulted into an overall increase of 50% in the portal throughput. This exercise revealed that even a minor change to such a time-critical code could significantly affect the overall component performance. Although these changes increased the portal throughput, the hogging phenomena found in the first design implementation still persisted.

6.3.2.3 Third Design Implementation

The main problem identified with the above two implementations was the 'hogging' between the transmit and receive processes since both processes were based on interrupts. Another problem was associated with the CFR chip; there were a few bugs in the CFR chip (these are described in the next section). In fixing these bugs, the low level transmit and receive code became very large and as a result, the portal throughput was affected.

A radical re-design which split the CFR handling into processes with the receive handled by polling and the transmit handled by interrupt, allowed the throughput to increase dramatically. This was achieved by removing the buffer code and relying instead on hardware backoff to the CFR ring chips. This resulted in the clients retrying data minipackets until the minipacket could be handled by the portal. This redesign of the portal code allowed the two-way throughput to exceed 5400 minipackets/sec (over 1.2 Mbits/sec of useful data rate). The one-way throughput was found to be around 6000 minipackets/sec which was 10% higher with respect to the two-way throughput. This indicated that the hogging between transmit and receive processes had significantly reduced.

Fig. 6.2 shows the throughput-delay characteristics of the portal (with one-way loading). The average minipacket delay for the priority traffic is plotted against the total traffic load. The figure shows that when two portals are involved in the communication chain (two client LANs connected to the exchange LAN via the

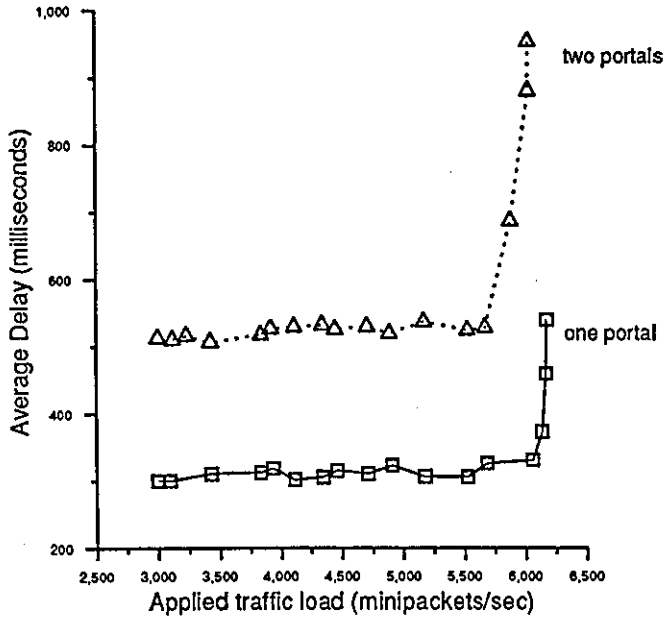


Fig. 6.2 Throughput-Delay Characteristics of the Portal (one-way loading)

portals), the minipacket delay is almost double that of the minipacket delay when only one portal is involved in the communication chain (see Fig. 6.1). The increase in minipacket delay at higher loads is merely due the retransmissions to the portal. Since no buffering is provided in the portal, there will not be any queueing delay.

Fig. 6.3 shows the throughput-delay characteristics when the portal is loaded in both directions. At relatively low loading (below saturation), a steady increase in the average delay can be seen. Again, this increase is due to the backoff effect from the portal; the generators retransmit those minipackets which have a 'try again' response from the portal. The number of automatic retransmissions can be adjusted as required. It was adjusted to 11 retries, with each retry being attempted after two ring gaps (the gap between tail and end of the CFR slot train) had passed the node. At higher loads, the minor fluctuations in the minipacket delay are thought to be due to a synchronisation effect; delay for a given traffic stream depends on the instantaneous blocking effect of parallel synchronous streams passing through the portal. This can result in a distinctive performance behaviour for certain traffic streams.

Fig. 6.4 shows the percentage loss of the priority traffic under heavy loading conditions. A similar fluctuating performance behaviour can be seen here. The graph

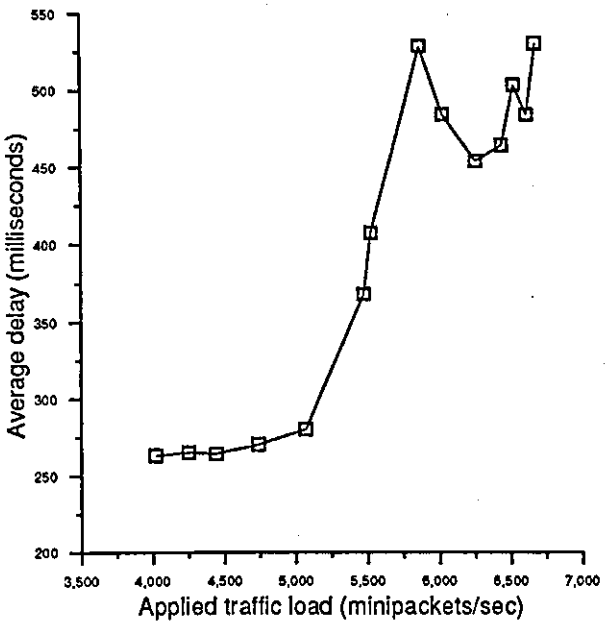


Fig. 6.3 Throughput-delay Characteristics of the Portal in Heavy Loading Conditions

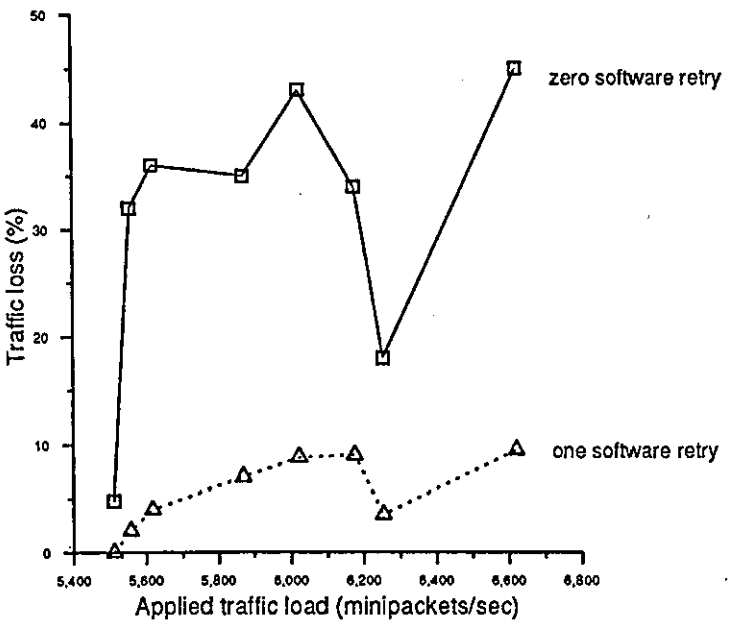


Fig. 6.4 Packet Loss Characteristics of the Portal in Heavy Loading Conditions

also shows that if one software retry is allowed (on top of chip hardware retries), there is a significant reduction in minipacket loss. Of course this reduction is at the expense of an increase in minipacket delay and a slight reduction in the throughput. This can be seen in Fig. 6.5 in which the relative frequency of the minipacket delay is plotted at two typical traffic loads. Five retransmissions were allowed in order to eliminate any traffic loss. At 5000 minipackets/sec traffic load, there is virtually no spread in

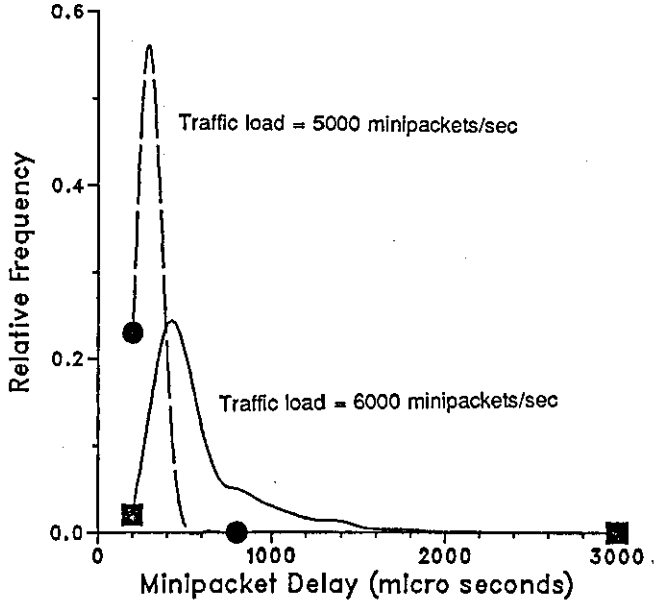
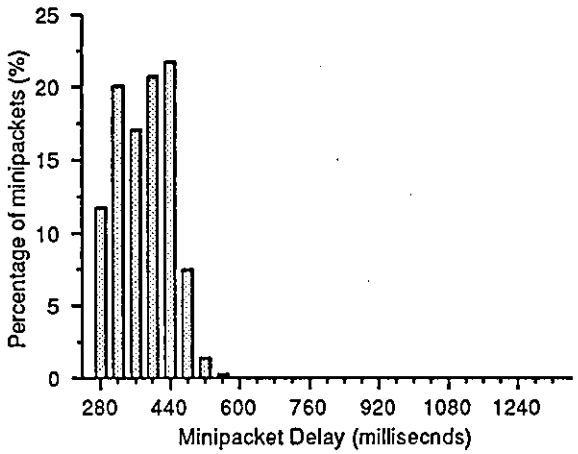


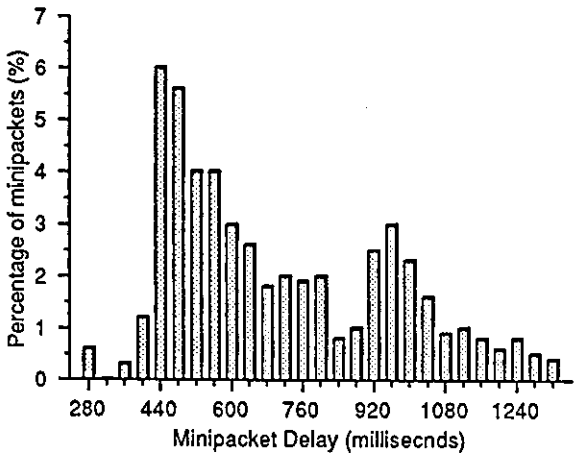
Fig. 6.5 Relative Frequency of Minipacket Delay at Typical Loadings

minipacket delay. However, at 6000 minipackets/sec, the spread has increased significantly along with an overall increase in the average minipacket delay.

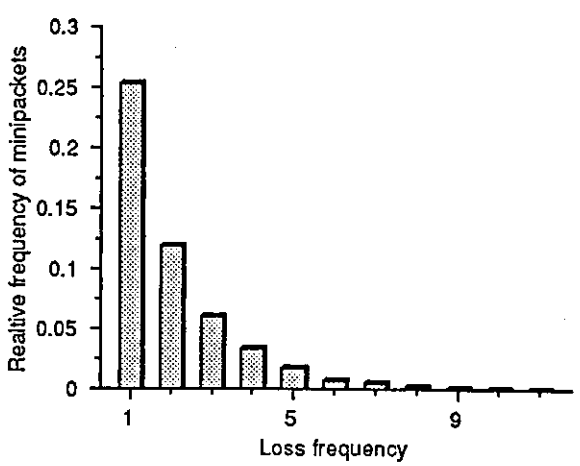
Fig. 6.6 shows the throughput-delay and traffic loss characteristics of the portal for a voice connection at typical loading conditions. Only one software retry was allowed for the priority traffic while the hardware retries were kept at 11 retries. When the loading is below the saturating point of the portal, the graph shows that a high percentage of the minipackets experienced less than 0.5 milliseconds delay. Above saturation, the spread has increased significantly and delay has increased up to 1.2 milliseconds. The figure also shows the packet loss characteristics when the portal is loaded above its saturation point. As can be seen, the majority of minipackets are dropped individually rather than in chunks. However, a very minor percentage of minipackets suffer loss as a small group.



(a) Below saturation (load = 5688 minipackets/s)



(b) Above saturation (load = 5861 minipackets/s)



Packet loss characteristics

Fig. 6.6 Throughput-delay and Loss Characteristics of the Portal for a Voice Traffic Stream at typical Loading Conditions

6.3.3 Congestion and Flow control mechanism

Congestion and flow control is an important issue in the design and operation of packet-switched communication networks. Flow control relates to the control of the flow of information between a sender and a receiver to prevent the sender flooding the receiver; the main goal is to prevent the overflow of buffers dedicated to a connection. Congestion control is required to prevent more traffic entering a network than can be handled; a global procedure is carried out by the bridges to prevent network congestion and the controlling action may be exercised on many source/destination pairs indiscriminately and simultaneously. Flow control and congestion control mechanisms are required to avoid congestion and deadlocks in the bridges, ensure fair sharing of network resources among competing users, and achieve speed-matching between the network and its users.

6.3.3.1 Window Flow Control

The bridging components which switch traffic at the MAC-layer are generally ill-equipped for the provision of effective flow or congestion control facilities [Gerla 88]. This is because these bridges do not have access to the network and transport layers where flow control is generally embedded. In X.25 packet-switched networks, a window flow control [Tenenbaum 88] is commonly used in which a node forwards a fixed number of packets to the destination after which it requires an acknowledgment from the remote end before forwarding the next chunk of packets. The number of packets in a chunk or the window size, is negotiated the connection set up time. However, these windows can be dynamically adjusted (increased or decreased) depending upon the instantaneous loading on the participating machines. By applying window flow control on a link-by-link basis, the individual traffic streams can be throttled by backpressure. If an upstream node along a path is congested, it can reduce the window size (or even shut off the window) and start a chain reaction which will eventually cut the supply of new packets entering the network. Such a mechanism is most appropriate for transferring non-real time data such as file transfers. A file transfer can tolerate being slowed down so long as the total transfer time is within given a limit. However, window flow control is not considered suitable for real time applications since it would increase delays and processing times at the

bridging components. In the Unison network, the bulk of the inter-LAN or inter-site traffic is from real time application services, therefore a window-type flow control is not appropriate due to the reasons described above.

6.3.3.2 Packet Dropping Flow Control

The easiest way to prevent congestion in the bridges would be to drop packets when the buffers become full. This approach has been employed in the Unison network to relieve congestion in the Portals. The only advantage of adopting such a simple approach is that there is very little processing overhead in the bridges. This approach is suitable for those networks which support light-weight protocols and are based on 'best effort' delivery mechanisms for transporting user data. The main drawback in using this congestion control method is its non-suitability for non-real time applications. Indiscriminate dropping of packets can be counter productive because it would trigger retransmissions (provided in the end-to-end transport protocol for loss-sensitive traffic) thus allowing the near-congestion situation to persist. In the portal, such a drawback can be avoided (to some extent) by dropping priority traffic in preference to non-priority traffic since the former will not trigger retransmissions. Priority services, such as voice, do not have retransmissions on an end-to-end basis in the case of packet loss.

The packet dropping type flow control has not been found to be very effective in practice. In the Unison network, most of the real-time traffic sources, such as voice and video, generate traffic with a constant packet rate. The portal has no facility to throttle individual traffic streams and does not provide any feed back to the sources so that traffic may be reduced to avoid congestion. The problem becomes more complicated if a number of bridging components are involved in the communication chain. The sources, not knowing the fate of the traffic in transit, would keep pumping traffic into the network unless an end-to-end feed back mechanism exists which periodically checks the safe arrival of traffic at the receivers.

6.3.3.3 Call-oriented Flow Control

It appears that neither window nor packet dropping type flow and congestion control techniques are sufficient for MAC-level bridges such as the portals in the Unison network. For such networks, a call-oriented flow control [Ohnishi 88] has been

suggested. The network resources are allocated to each call according to its throughput requirements. The bandwidth requirement is negotiated by the network clients at connection set up time. The allocated bandwidth would remain available for the duration of the call. This mechanism is very similar to circuit-switched telephone networks in which a new connection is accepted only if a free channel is available between the caller and the called party. Otherwise it is rejected. Such a mechanism although considered appropriate for an integrated services packet-switched network, has its own implementation problems.

The call-oriented flow control approach faces two main problems. Firstly, there is a need to have some form of bandwidth management entity or entities which have an up-to-date global knowledge of the available network resources. Obviously, achieving this in a wide area network in which a number of bridges are involved is an extremely complex problem. Secondly, it would be necessary to accurately characterise the traffic sources. At connection set up time, the traffic sources would provide a few traffic parameters, such as average and peak bandwidth requirements, which would be used by the network bandwidth management to allocate the network resources. This means that efficient network resource utilisation would depend on the accurate characterisation of the various traffic sources. If the traffic sources are under-characterised, this would result in poor utilisation of the network. If over-characterised, the required quality of service cannot be guaranteed.

In the Unison network, although each client provides a bandwidth parameter in the RPC at connection set up time, it has not been utilised for network bandwidth allocation for individual traffic streams. No effort has been made towards providing a call-oriented flow control.

6.4 Current Problems with the Portal

The portal provides bridging between the prototype high-speed CFRs. The CFR medium access layer has been implemented mostly in hardware [Acorn 87b]. For instance, the CFR semi-custom station chip performs all of the network functions concerned with the transmission and reception of minipackets in firmware. It is almost impossible to achieve all the functionalities in such a complex chip without some minor bugs. The portal also uses transputer technology. The Unison project was perhaps among the first few who were making use of this technology with little

experience in using such complex systems. To achieve an optimum performance from the portal was the main target, other targets were to make the portal as reliable as possible.

The traffic generators were used extensively to investigate the reliability problems of the portal. Most of the software and hardware bugs found when using the portal were corrected in the portal's software to enhance its reliability. Obviously, bug-fixing in software introduces unavoidable adverse effects on the performance of the portal, thus reducing the overall network performance. The following sub-sections discuss the important existing problems concerning the portal along with the solutions adopted for their remedy.

6.4.1 Ring Reframing

Each CFR station possesses framing circuitry used to keep the stations in synchronism with the slot structure of the CFR. The monitor station is responsible for maintaining the slot structure. This station initialises the slot structure on power up or when a serious error occurs on the ring.

The framing circuitry of the ring stations should be able to lock to the slot structure of the CFR in case of ring reframing and the station chip should function normally after such an event. Unfortunately, it was found that the station chip does not function properly and the receive and transmit parts of the chip do not work as desired unless the chip is initialised. For some reason, ring reframing occured quite often and caused a lot of reliability problems to various network components. However, ring reframing was significantly reduced by reducing the ring clock frequency. Reframing presented a serious problem to the Portal; the portal code has to detect such a condition and take appropriate action. The problem was resolved by designing a piece of code to detect reframing. The portal periodically (after every 30 sec) sends a packet to itself. If the packet fails to return, another packet is sent to confirm a reframing event. If the second packet returns successfully, no action is taken, otherwise the chip is re-initialised. This would obviously affect the on-going traffic and also result in the loss of a few packets. Furthermore, the extra code necessary for reframing detection introduces overheadswhich affect the overall portal performance.

6.4.2 Portal Locking

Quite recently, it was found that the portal occasionally locks up if it is operated in overload conditions especially when the ring size is more than two slots. Using traffic generators, heavy loading conditions were emulated. Some debug messages were placed in the portal code to display the status of the various CFR chip registers on the terminal. The debug messages have shown that the transmit FIFO of the station chip was creating the occasional chip lock-up problem. At certain high loading, it was found that the transmit FIFO did not clear itself (which should occur automatically once the minipacket has been launched onto the ring) thus blocking the passage of other traffic. This problem was fixed temporarily by providing a timeout in the transmit process so that if no interrupt comes from the transmit FIFO indicating a FIFO clearance, a reset is forced to return the portal to normal working.

6.5 Discussion

The performance problems associated with the portal have been identified with the help of the measurement system. The identified problems have provided the basis for upgrading and fine-tuning of the portal switching algorithm.

Although the adjustment to the portal resulted in a dramatic increase in minipacket throughput, this was achieved by removing the code responsible for buffer management. Without buffers, the portal relies on backoff to the CFR chip which in turn has a backoff effect on the traffic sources (the clients). In fact, this is a built-in flow control mechanism within the CFR minipacket protocol. In a variable bit rate traffic environment, there is a likelihood that the portal would occasionally experience overload conditions. The portal must be equipped with some buffers in order to cache small-duration overload conditions. If the buffers are provided using the same hardware, a significant proportion of the portal performance would have to be sacrificed. One possible solution would be to take advantage of the transputer's parallel architecture. The transputer architecture provides the necessary flexibility and simplicity in terms of adding extra processing power to an existing system. If another transputer chip is incorporated within the existing portal hardware, this could take care of the processing overhead which would result from the buffer management code.

Another possible way of reducing the portal processing time is to incorporate dedicated hardware for traffic switching. A hardware implementation would probably involve special-purpose hardware to perform per-packet processing. Hardware implementation is a challenging task since a hardware implementation is hard to debug completely. On the other hand, a software (micro-processor) based implementation is unlikely to be fast enough to achieve high throughput.

An important issue in an integrated services environment is the provision of an effective flow and congestion control mechanism. Such mechanisms are required to achieve performance controllability for various traffic types. Where these mechanisms should be implemented is another issue; should these reside in the packet switches (hop-by-hop basis) or should they become part of the network global management (end-to-end basis). A recent study has suggested that these mechanisms should be part of the network management if a real integrated services environment is to be supported [Ohnishi 88]. It is hoped that effective flow and congestion control mechanisms incorporated in the networks would be able to help in overcoming problems such as throughput degradation and loss of quality of service in overload conditions.

Chapter 7

Discussion and Conclusions

7.1 Discussion

A performance measurement methodology has been described for the performance analysis of integrated services networks such as the Unison network. A number of performance issues associated with the Unison network have been considered and investigated in some detail. There are two aspects which are of special interest to this research work. The first relates to network performance from a user perspective, and the second relates to the performance optimisation of the Unison network in relation to network bridging components. Another important issue explored concerns the synchronisation of clocks between distant machines. The Chronological Timestamping technique has been introduced which enables one-way packet delay to be measured accurately across various Unison sites.

A general discussion of each of the main parts of the research is given here along with possible extensions to the present work.

(1) Traffic generation and end-to-end user performance : The traffic generation facility enables the network performance to be observed from a number of perspectives. In previous networks,the use of a traffic generation facility was limited only to the analysis of network protocols such as routing protocols. In the Unison network, its use has been extended to provide end-to-end performance information for

the various application types supported by the Unison network. The Unison applications are mainly voice, video and computer data. The voice and video applications are mainly of the Constant Bit Rate (CBR) type. Although Variable Bit Rate (VBR) traffic emulation for voice and video traffic was provided in the traffic generators, CBR traffic was used predominently in determining the end-to-end performance for voice and video applications. In particular, the performance of a voice connection has been rigorously studied in various network configurations. The main interest was to determine how the performance of a voice connection is affected when the network bridging components are stressed with high traffic loads.

In order to determine the end-to-end performance under heavy loading conditions, it is essential to know first the maximum traffic handling capacities of the various network bridging components. This enables the communication chain involved in the experiments to be saturated with the precise traffic loads. In most of the experiments, two pairs of traffic generators and receivers had been used to conduct performance tests; one pair was used to stimulate the network with the desired load and traffic pattern while the other pair emulated the voice or video traffic stream. Some tests were also carried out by exerting traffic generated from real application components. In fact, the use of traffic generators would be more beneficial if the tests were conducted when a number of application components are using the network. When the emulated traffic stream (generated from the TGR) share the network resources along with other application streams, the analysis of the emulated traffic stream provides a much better and realistic picture of the network performance. Due to the experimental nature of the Unison network, there were limited real application traffic sources available on the network, particularly on the CFR based network. However, the two-pairs of traffic generators and receivers worked very efficiently in determining the end-to-end performance under various traffic loadings. The traffic loading capacity of a single pair of TGRs was high enough to overload any of the network bridging components.

In the CR based Unison network, measurement experiments have shown that the delay characteristics of voice connections are severely affected by the bottleneck effect of the CR-CFR portal. In addition to the bottleneck effect, there are two other major factors associated with the portal. The first relates to the handling of the traffic on the CR side of the portal. The portal switches application traffic at the block level and cannot multiplex application blocks in transmission or reception. This introduces

contention when accessing the portal; applications which use large blocks for communication would occupy the portal for longer periods in comparison with applications which use small blocks. It was noted that a 2048 byte block (the video application uses this block size for transferring video data) would typically occupy the portal for around 25 milliseconds. This is unfair particularly to the voice applications which use small blocks in order to reduce packetisation delay. At higher traffic loads, the contention effect would be worse when voice traffic streams (with small block size) are travelling in parallel with the other traffic streams which use large blocks. The second factor which affects the voice traffic streams is related to the queueing strategy of the portal. The portal switches traffic on a first-in-first-out basis without any traffic class distinction. Again in a mixed traffic environment, the voice traffic experiences a large spread in block delays when the portal is overloaded.

In the case of the video traffic, the throughput performance is restricted by the maximum point-to-point bandwidth of the CR in the CR based Unison network. The video traffic requires high bandwidth across the network in order to support a high frame update rate. The traffic generators and the video applications used the high performance user interface which can pump data onto the CR up to the maximum point-to-point bandwidth of the CR. For a typical CR size of 2 slots and 20 gap bits, the maximum point-to-point bandwidth that can be achieved is around 600 Kbits/sec. Without using any video data compression, the maximum frame rate for a typical video connection (frame size of 256 * 256 pixels, and a pixel depth of 16 bits) would be around 1.7 frames per second. Obviously, some video compression technique has to be used if higher frame rates are to be supported.

In contrast, the CFR based Unison network encompasses features which overcome most of the problems encountered by the real-time applications in the CR based network. Due to an increase in ring speed and slot size, the maximum point-to-point bandwidth in the CFR is theoretically over 10 Mbits/sec. This has a dramatic effect on the throughput performance of the network to user interface. The CFR based traffic generators can pump data traffic into the network at speeds of up to 2.5 Mbits/sec. The video application was the first to take advantage of this efficient network user interface to support high frame rates [Lu 89]. The CFR-CFR portal, being a minipacket bridge, overcomes most of the problems which existed in the CR-CFR portal. The CFR-CFR Portal switches traffic in fixed-size small packets and allows the multiplexed transmission and reception of packets from a number of clients.

This attractive feature not only increases the packet switching rate of the portal but also provides a reasonable degree of fairness in accessing the portal. Furthermore, the CFR-CFR portal has a two-level traffic handling mechanism; priority traffic is handled first in preference to non-priority traffic. This was an immediate lesson learned from the measurement results carried out on the CR-CFR portal, that is, some services (such as voice) require a distinct performance from the switching components in order to provide an acceptable quality of service under varying load conditions. The measurement results over the CFR based network have shown that voice traffic does not suffer from the adverse characteristics experiented in the CR based network. Also, voice traffic does not suffer significantly from the statistical loading of the CFR-CFR portal.

Although the measurement methodology has proved its effectiveness in practice, there are various problems associated with this technique. The first obvious problem is that it is costly to implement. The cost will be proportional to the number of generators and receivers provided over the network. The experience with the Unison network however, has indicated that the user equipment can be used intermittently for the purpose of traffic generation. It can therefore be programmed to perform other operations when measurements are not being taken. In the CFR based network, apart from the basic measurement infrastructure, the user equipment was programmed to act as traffic generators and receivers. Another problem associated with this measurement methodology is the dependency of the test results on the overall network traffic. When an experiment is in progress in a particular section of the network, the precise network load other than the test traffic must be known for a better correlation and analysis of the test results. Any hidden or unknown traffic will affect the test traffic and will eventually lead to a poor analysis of the network section under study. In experimental networks, this problem can be avoided by carefully planning the test experiments. In particular, the inter-site experiments have to be carefully coordinated.

This thesis does not address the performance evaluation of two heterogeneous networks (employing different communication protocols) using the measurement methodology based on TGRs. In the Unison network, although dissimilar LANs (CFR and CR) were used, no attention was given to the development of protocol converters which could have enabled communication to be achieved between heterogeneous networks. However, it would have been very interesting to carry out experiments between heterogeneous LANs using the TGR based measurement strategy. In

principle, the measurement strategy is adoptable to conduct experiments between heterogeneous networks. Another possible extension to this work could be to develop an expert system based on the real-time performance information collected by the TGRs. A similar strategy has been used in Magnet [Lazor 85] in which an expert system composed of a traffic generator and a system monitor was used to investigate the protocol performance. In the expert system, the traffic generator was used to load the network with various traffic patterns while the monitor recorded the network events for off-line and on-line examination as well as to provide a mechanism for setting protocol parameters.

(2) Network Optimisation : One of the main advantages of using the measurement methodology based on traffic generators is that it can be used in a 'measure and tune' fashion to improve the performance characteristics of network bridging components. As described in Chapter 6, it has been extensively used to improve the CFR-CFR portal performance. The traffic switching algorithm of the CFR portal has been studied in some detail. The experience with the CFR portal has indicated that traffic generators are essential tools for the component designers in order to understand how the component will interact with the application streams. This does not only help in debugging the component code but also helps to tune precisely various parameters which can improve the component performance.

Some experiments were also conducted with the ramp and the multicast data server. The performance of these components was also optimised and the results reported in Chapter 5 demonstrate the effectiveness of the traffic generators. The performance characteristics of the ramp were investigated by placing two pairs of generators and receivers on the exchange rings and configuring the ramp in a loop back configuration. This allowed the ramp to be tested independently as an individual component. The traffic loop back facility provided in the network proved extremely useful. Any number of components can be selected in a communication chain and performance tests can be performed over various network sections. The delay characteristics of the ramp were investigated and improved. This was achieved by making adjustments to the buffer serving process of the ramp as described in Chapter 5.

A network may perform very well in a balanced loading condition, yet perform poorly in an unbalanced loading condition. The performance of the network in an unbalanced loading condition depends on the effectiveness of the flow and congestion control

schemes provided in the network. Effective flow control is required to ensure that the network performance remains consistent in all loading conditions. In the CFR-CFR portal, packet dropping flow control was initially implemented. In this type of flow control, the packets are dropped selectively in the case of overload; delay sensitive traffic (priority traffic) is forwarded first but it is dropped first if the incoming traffic exceeds a certain limit in relation to the loss sensitive traffic (non-priority traffic). Later, in improving the portal performance, the flow control provided by the low-level minipacket protocol was utilised as described in Chapter 6. The flow and congestion control is a complicated issue and should be dealt with in greater depth.

(3) Chronological Timestamping : Another important aspect which has been addressed in this thesis is the provision of a one-way delay measurement facility in the measurement system. Traditionally, delay measurements have been carried out either using round-trip delay measurement techniques or by employing timestamping techniques within the switches. These techniques have been reviewed in Chapter 3. Neither of these delay measurement techniques provide the accuracy which has been achieved using chronological timestamping. To carry out one-way delay measurements, the problem is one of synchronisation. This problem has been successfully resolved by interfacing the Rugby time clock receivers to the traffic generators and receivers. The Rugby time clock receivers provide synchronisation signals almost continuously to the traffic generators, so problems like crystal clock drift and uncertainties in message delivery times (in exchanging clock values) which are sources of inaccuracies in other synchronisation methods, are entirely eliminated. As described in chapter 3, the most important feature of the chronological timestamping technique is that the clock drift does not have a cumulative effect over the measurement results. Furthermore, this technique is much easier to implement; the hardware clocks provided in the single board computers are used to count events (computer ticks) and simple software procedures are used to timestamp the packets.

The use of the chronological timestamping technique can be extended to other applications such as packet voice synchronisation. In packet-switched networks, packet voice synchronisation is required to reconstruct a continuous stream of speech data from the set of packets that arrive at the receiver where each packet may encounter a different amount of buffering delay in the packet network. A number of packet voice synchronisation techniques have been proposed [Montgomery 83]. It is

believed that chronological timestamping could provide the necessary ease of implementation and the desired synchronisation accuracy which other techniques lack.

7.2 Conclusions

This thesis has described a performance measurement methodology, its underlying philosophy and its implementation problems over integrated services networks. Based on the measurement methodology, a complete measurement system has been developed and implemented over the Unison network and the use of the measurement system has enabled a number of performance issues to be explored.

The measurement system was based on a number of dedicated traffic generators and receivers along with auxiliary components which provide system control and statistics collection functions. Each generator was equipped with a chronological timestamping facility which enables one-way transit delay to be measured accurately across the network. The problem of clock synchronisation between distant machines was resolved by interfacing Rugby time clock receivers to each generator and receiver. The traffic generating functions, which form the core of the measurement system, enabled a variety of traffic patterns to be generated. The attempt was to emulate traffic patterns that would be generated by real network users.

The traffic generators were dispersed across the Unison network to enable a number of performance experiments to be carried out in various network configurations. Much effort was aimed at providing a picture of the network performance from the user perspective. This was directed towards providing information about the network characteristics to the designers of network applications, especially real time services such as voice and video. The measurement system has been used to characterise the operational performance of various network routes, for different mixes of traffic, and to determine the capabilities of the network switching components. The system had also been used to analyse and improve the performance of various network components such as the CFR-CFR portal, ramp and multicast data server. The strategy adopted has thus provided an input to both application designers and also to the network architecture implementors.

The practical experience of using this measurement approach has revealed that it is almost an essential requirement in order to help the network development process,

particularly of experimental networks. In general, these networks evolve in stages. Firstly, a few network components are developed to verify the fundamental principles. After the functionality and expected performance characteristics of the individual components have been properly tested, further components are developed to extend the network to cover more of the planned infrastructure leading eventually to a fully fledged network. The same is true of the development of network applications. Such a step-by-step network development process requires a measurement strategy which can provide performance information at the various network evolution stages. The extensive use of the described measurement strategy on the Unison network has demonstrated its effectiveness in providing inputs at various stages of the network development cycle.

References

[Acorn 87a] *"Programmer's Reference Manual"*
Acorn Computers Limited, Cambridge, England 1987.

[Acorn 87b] *"CFR Station Chip Datasheet"*
Acorn Computers Limited, Cambridge, England, 1987.

[Adams 84] Adams, C.J.
"Universe Network Protocol Architecture"
Project Universe Report, No. 2, 1984.

[Adams 85] Adams, C.J. and Ades, S.
"Voice Experiments in the Universe Project"
Proc. of IEEE Int. Conf. on Communications, New York, 1985

[Amer 82] Amer, P.D.
"A Measurement Center for the NBS Local Area Computer Network"
IEEE Transactions on Computers, Vol. C-31, No. 8, Aug. 1982

[Becker 85] Becker, J.
"On Measurement of Packet Switching Networks"
Proc. of the IFIP IC6 Working Conf. COMNET 85, 1985.

[Blair 82] Blair, G.S. et al
"A Performance Comparison of Ethernet and Cambridge Digital Communication Ring"
Computer Networks, North-Holland, May 1982.

[Brady 68] Brady P.T.
"A Statistical Analysis of On-Off Patterns in 16 Conversations"
Bell System Technical Journal, Vol. 47, January 1968

[Burren 89a] Burren, J.W. and Cooper, C.S.
"Project Universe - An Experiment in High-Speed Computer Networking"
Oxford University Press, 1989.

[Burren 89b] Burren J.W.
"Flexible Aggregation of Bandwidth for Primary Rate ISDN"
ACM SIGCOMM'89 Symposium, 1989.

[Cambridge Kits] *"MSF Clock / 60KHz Receiver Kit"*
Cambridge Kits, 45 Old School Lane, Cambridge, England.

[Chin 88] Chin, H.S. , et al
"Statistics of Variable Bit Rate Video Signals for View Phone Type Pictures"
IERE 5th Int. Conf. on Processing of Signals in Communications, Loughborough, England, 1988.

[Chin 89] Chin, H.S.
"Transmission of Variable Bit Rate Video Over an Orwell Ring"
Ph.D Thesis, Electronic and Elect. Eng. Dept.
Loughborough University of Technology, England, 1989.

[Clark 86] Clark, P. et al
"Unison - Communication Research for Office Applications"
IEE Electronics and Power, Vol. 32, No. 9, 1986.

[Gan 86] Gan and Davidson
"Speech Comparison for Storage and Transmission using Silence Detection"
Proc. IEEE ICC Conf., Toronto, 1986.

[Gerla 88] Gerla, M. and Kleinrock, L.
"Congestion Control in Interconnected LANs"
IEEE Network Vol. 2, No. 1, January 1988.

[Gruber 82] Gruber J.G.
"A Comparison of Measured and Calculated Speech Temporal Parameters relevant to Speech Activity Detection"
IEEE Trans. on Communications, Vol. com-30, No. 4, April 1982.

[Hamilton 84] Hamilton, K.G.
"A Remote Procedure Call System"
Ph.D Thesis, University of Cambridge, December 1984.

[Hopper 86] Hopper, A. et al
"Local Area Network Design"
Addison Wesley Publishing Company, 1986.

[Illyas 85] Illyas, M. and Mouftah, H.T.
"Performance Evaluation of Computer Communications Networks"
IEEE Communication Magazine, Vol. 23, No. 4, April 1985.

[Inmos 86] Inmos Limited
"Reference Manual - Transputer"
Inmos Limited, 1986.

[Inmos 88a] Inmos Limited, edited by Hoare, C.A.R
"Occam 2 Reference Manual"
Prentice Hall International , 1988.

[Inmos 88b] Inmos Limited
"Transputer Development System"
Prentice Hall International, 1988.

[Jacobs 78] Jacobs, I.M ,et al
"General Purpose Packet Satellite Networks"
Proc. IEEE, Vol. 66, No. 11, November 1978.

[Kieth 86] Kieth, G.
"The Alvey High Speed Network"
British Telecom Research Labs, United Kingdom, 1986.

[Klienrock 76] Klienrock, L.
"Queueing Systems"
Computer Applications Vol. 2, New York: Wiley 1976.

[Lazar 85] Lazar, A. A. et al
"MAGNET: Columbia's Integrated Network Testbed"
IEEE Journal on Selected Areas in Communications
Vol. 3, No. 6, November 1985

[Lazar 87] Lazar, A.A. and White, J.S
"Packet Video on MAGNET"
Optical Engineering Journal, Vol. XXVI, July 1987

[Leslie 83] Leslie, I.M.
"Extending the Local Area Network"
Technical Report No. 43,
University of Cambridge Computer Laboratory, 1983.

[Logica 81a] Logical VTS Limited
"Polynet Network Manual"
86, Newman St., London, England, 1981.

[Logica 81b] Logical VTS limited
"Multibus Intelligent Interface Unit VMI-1"
86, Newman St., London, England, 1981.

[Lu 89] Lu, G.J.
"Graphics and Video Communications over an Integrated Services Network"
To be submitted as Ph.D Thesis, Loughborough University, 1989.

[Montgomery 83] Montgomery W.A.
"Techniques for Packet Voice Synchronisation"
IEEE Journal on Selected Areas on Communications,
Vol. 1, No. 6, December 1983.

[Murphy 88] Murphy, B.J. et al
"Multi-media Applications over a Wide Area Network"
UK IT 88, Conf. Publications,
University College swansea, July 1988.

[Murphy 89] Murphy, B.J.
"Multi-media Services in a Distributed Office"
To be submitted as Ph.D Thesis, Loughborough University, 1989.

[Needham 87] Needham R.M. and Ades S.
"Integrated Services Using a Baseband Local Network"
Proc. of IEE INCUT 87 Conf., London, June 1987.

[NPL] National Physical Laboratory
"Standard Frequency and Time Signal Transmission for for MSF Rugby"
U.K. National Physical Laboratory, Teddington, Middlesex.

[Ohnishi 88] Ohnishi, H. et al
"Flow Control Schemes and Delay/Loss Tradeoff in ATM Networks"
IEEE Journal on Selected Areas on Communications,
Vol. 6, No. 9, December 1988.

[Parish 88] Parish, D.J. and Siddiqui M.H.
"Performance Measurements of the Project Unison Multimedia Network"
IERE 5th Int. Conf. on Processing of Signals in Communications,
Loughborough, England, September 1988.

[Reiser 82] Reiser, M.
"Performance Evaluations of Data Communication Networks"
Proc. of IEEE, Vol. 70, No. 2, February 1982.

[Schneider 86] Schneider, F.B.
"A paradigm for Reliable Clock Synchronisation"
Dept. of Computer Science Cornell University, Ithaca, New York,
February, 1986.

[Seitz 83] Seitz, N.B. et al
"User-Oriented Performance Measurements on the ARPANET"
IEEE Communications Magazine, August 1983.

[Shrimpton 87] Shrimpton, D.
"Multicast Services: Initial Ideas"
Unison Document, UR043, Rutherford Appleton Laboratory, 1987.

[Siddiqui 89] Siddiqui, M.H. et al
"Performance of Local and Remote Bridge Components in Project Unison - A 50 Mbits/sec ATM Network"
Proc. of the 1989 Singapore Int. Conf. on Networks,
SICON'89, Singapore, July 1989.

[Temple 84] Temple, S.
"The Design of a Ring Communication Network"
Technical Report No. 52, University of Cambridge, 1984.

[Tanenbaum 88] Tenenbaum, A.S.
"Computer Networks (second edition)"
Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[Tennenhouse 87] Tennenhouse D. et al
"Exploiting Wideband ISDN: The Unison Exchange"
Proc. of IEEE Infocom'87, San Francisco, March/April 1987.

[Tobagi 78] Tobagi, F.A. et al
"Modeling and Measurement Techniques"
Proc. of the IEEE, Vol. 66, No. 11, November 1978.

[Treadwell 80] Treadwell, S.W.
"Measurement Methods in Packet-switched Networks"
INDRA Technical Report No. 65
Dept. of Computer Science, University College London, 1980.

[Wilber 84] Wilber, S.
"Network Monitoring, Management and Control"
Project Universe Report No. 3, 1984.

Appendix 1

Traffic Generating Functions

This appendix provides a brief summary of the traffic generating functions supported by the traffic generators.

Deterministic Traffic Generating function

This function generates traffic with a constant data rate. In the CR based network, the rate can be varied from a few blocks to the maximum point-to-point bandwidth of the CR. In the CFR based network, the rate can go as high as 10,000 minipackets/sec (over 2.5 Mbits/sec). These generation rates were high enough to overload any of the network bridging components. The following traffic parameters can be selected before initiating a performance measurement test.

Traffic type:	priority/non-priority
Block size:	number of bytes in a block
	(in CFR based network, fixed-size blocks were used)
Timestamping:	enabled/disabled
Inter-packet time:	delay between two consecutive packets (microseconds)
Test time:	duration of traffic generation (seconds)

Exponential Traffic Generating Function

The traffic generated by this function follows an exponential distribution. The traffic related parameters are similar to the deterministic traffic generating function except that the inter-block delay is now exponentially distributed. Table 1 shows the traffic throughput (averaged over the test time) achieved for various mean inter-block times. The tests were conducted over the client CFR.

Burst Traffic Generating Function

This function periodically generates a burst of traffic. It emulates the computer data traffic since computers interact with each other by exchanging bursts of blocks. The traffic related parameters include:

Mean Inter-minipacket Time (milliseconds)	Achieved Generation Rate (minipackets/sec)
0.5	1268
1	780
1.5	558
2	444
2.5	364
3	299
3.5	269
4	228

Table 1 : Traffic Throughputs Achieved for Various Mean Inter-minipacket times

Traffic type: priority/non-priority

Block size: number of bytes in the block

Timestamping: enabled/disabled

Burst Size: number of blocks in each burst

Inter-block delay: delay between the emission of two consecutive blocks in a burst

Inter-burst delay: delay between the emission of two consecutive bursts

Test time: duration of traffic generation

Performance Accounting Function

This function accounts for the performance statistics yielded during test experiments. The performance statistics are normally based on histograms which are later used to prepare statistical distributions which facilitate performance analysis. The performance parameters include:

Achieved generation rate: blocks/sec (averaged over the test time)

Traffic lost statistics: histogram based traffic lost distribution

Single-way delay statistics:	histogram of single-way block delay distribution
Inter-arrival time:	histograms of block inter-arrival time distribution
Block corruption:	number of blocks corrupted
Sequence errors:	sequence error counts
Retransmissions:	retransmission counts

Voice Traffic Emulation

The deterministic traffic generating function can be used to generate constant block rate voice traffic. Block size and block rate can be selected such that the effective data rate corresponds to 64 Kbits/sec. The performance parameters which are important for a voice connection include block delay statistics, block inter-arrival time statistics, block discard rate, etc. The measurement of these parameters determine the quality of service available between the user ends

Silence suppressed voice traffic can be emulated by studying the voice temporal parameters. If one listens carefully to a typical telephone conversation, it can be observed that the conversation is actually a mixture of talkspurts and silence durations. Recent studies into the characteristics of voice signals show that there can be as much as 65% silence duration over the total conversation period [Gan 86]. In packet-switched networks where bandwidth is scarce, these silence intervals can be exploited in order to reduce the per voice connection bandwidth. In order to generate silence suppressed voice traffic, it is essential first to study the distributions of talkspurt and silence durations. The analysis of a number of speech conversation has shown that the distribution of talkspurt and silence durations follow a geometric distribution [Brady 68]. From the literature, a suitable algorithm along with the associated parameters was selected to emulate silence supressed voice traffic [Gruber 82]. During the talkspurt interval, constant block rate traffic (corresponding to 64 Kbits/sec) is generated while nothing is generated during the silence durations.

Video Traffic Emulation

Constant block rate video traffic can easily be emulated by defining the number of pixels per frame, pixel depth (number of bits per pixel), and the required frame rate. These parameters would determine the effective traffic generation rate. The traffic generated would be similar to the traffic generated from a deterministic traffic

generating function. By exerting such traffic on the network, the performance parameters, such as achieved frame rate, block loss rate, block delay statistics can be determined.

Variable block rate video traffic can be generated by studying the characteristics of the video signals. Over the years, a number of research groups have been conducting research into picture statistics [Chin 88]. The aim of these studies was to understand the behaviour of video signals so that these signals can be transported over bandwidth-limited networks effectively. These studies have shown that there is a lot of redundancy associated with the video signals which can be removed at the video traffic sources. In a recent study [Chin 89], head-and-shoulder type video signals have been investigated. This study has shown that the average inter-frame changes were around 15% - 20% after studying a number of frame sequences.

In order to generate variable block rate video traffic, the statistics of inter-frame changes were obtained from a real coder for a number of sequences of video frames (see Fig.1). In order to reduce the complexity, the data rate within a frame was

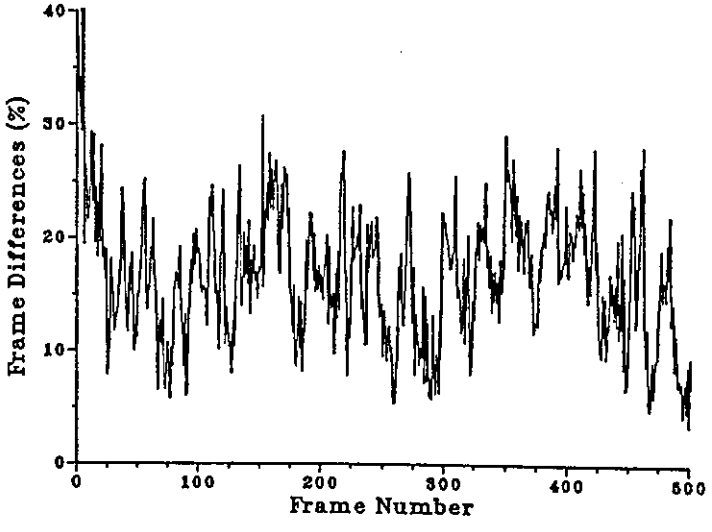


Fig. 1: Variation of Frame Differences for a Sequence of 500 Frames

considered constant. The figure shows that the average inter-frame changes for a sequence of 500 frames is around 20% with a peak-to-mean ratio of 2.

In order to generate traffic corresponding to such changes, it is essential first to know the maximum inter-site bandwidth achievable over the network. In the CR based Unison network, the maximum inter-site bandwidth which can be achieved was around 450 Kbits/sec. A mean of 200 Kbits/sec along with a peak-to-mean ratio of 2 was selected to generate variable block rate traffic in accordance with the inter-frame changes. The experiments conducted over the Unison network have shown that the network can support one such video traffic emulation load between the Unison sites without any traffic loss. Unfortunately, there were no other real variable block rate application components available on the network and it was difficult to produce any reasonable statistics from a single source.

Appendix 2

Rugby Clock Receiver Interface

This appendix presents the details of the interface between the Rugby time clock receiver and the TGR. It also describes the calibration procedure required to obtain accurate synchronisation between distant clocks.

A commercially available Rugby time clock receiver kit [Cambridge Kits] was used to receive time signals transmitted on 60KHz from MSF, Rugby, England. The receiver kit contains the radio clock receiver and decoder circuitry. As shown in Fig. 1, the radio clock receiver and decoder generate two signals : the SEC signal and DATA

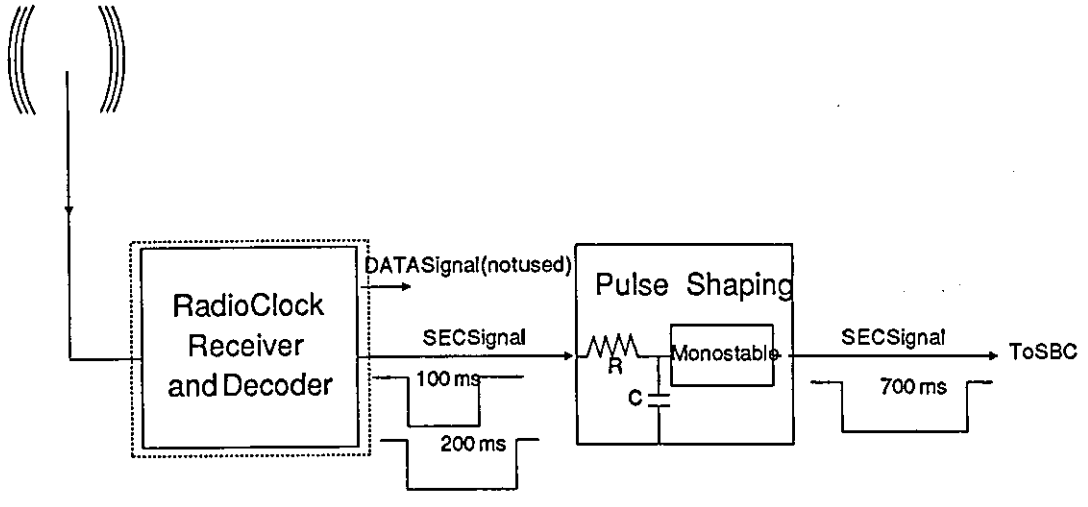


Fig. 1 : Block Diagram of the Radio Clock Receiver and Pulse Shaping Circuit

signal. The former is the second pulse whose leading edge marks the beginning of every second while DATA is the actual decoded signal which carries the time code. Since there was no intention to provide real-time clock values to the measurement system, the DATA signal was not used. The SEC signal can have variable pulse durations; a 100 millisecond pulse duration (voltage level low for a duration of 100 milliseconds) represents a bit 'zero' while a 200 millisecond pulse duration represents a bit 'one'. However, for the clock synchronisation, only the leading edge of the SEC signal is required.

Due to difficulties in reception, the radio clock receiver and decoder box (the dotted portion in the Fig. 1) have to be placed away from the rest of the interface circuitry. At LUT and RAL, it was found that the box had to be placed at least 2 metres away from the rest of the equipment (particularly from computer monitors) in order to avoid any interference to the signal reception. The SEC signal is carried over the twisted cable to the Single Board Computer (SBC) of the TGR. Before feeding the SEC signal to the SBC, it is passed through a pulse shaping circuit; this is required in order to filter out any interference on the SEC signal. Furthermore, in order to avoid the effect of double pulses and fast code [NPL], the SEC signal is passed through a monostable which produces a 700 millisecond pulse from the start of each second.

Fig. 2 shows the circuit diagram of the interface board for the transputer based SBC. A link adaptor chip [Inmos 86] along with the auxiliary circuitry provide the interface to the transputer. I_{valid} and I_{ack} signals of the link adaptor chip provide the necessary handshaking between the SEC signal and the transputer. The SEC signal is attached to I_{valid} via a D-type flip-flop which takes I_{valid} high at the start of every second. The link adaptor chip commences handshaking with the transputer when I_{valid} goes high. It sends a byte-wide data to the transputer via the serial transputer link. When the transputer process responds to this serial link by fetching the byte, it acknowledges the link adapter chip by taking I_{ack} high. I_{ack} is also used to clear the flip-flop in order to make I_{valid} low again. This completes the handshaking process of the link adapter chip.

Calibration of the Rugby radio clock receivers: In the radio clock receiver, an Automatic Gain Control (AGC) feedback loop is provided to overcome problems of signal strength variations. Without AGC, the variation in signal strength would cause errors in synchronisation. In order to calibrate this accurately, current monitoring circuitry exists in the AGC loop. This circuitry is used to monitor the current in the AGC loop when the radio clock receiver receives the timing signals. The first requirement in the calibration procedure is to ensure that the ferrite rod of the radio receiver box is placed broadside to Rugby. This would be essential in order to receive good radio signals. In the Rugby clock receiver, an audible signal is provided to which a high impedence loadspeaker can be connected. By tuning the local oscillator of the Rugby clock, a once-a-second "beep, beep" signal can be heard on the speaker. This is useful to test whether the Rugby clock transmission is on the air or not. Periodically, the Rugby transmission is switched off for the purpose of maintenance.

Furthermore, the use of the speaker gives a rough idea about the quality of the received signal.

An ammeter (which can measure current in uA) is required to monitor the current in the AGC loop of the radio receiver. By connecting the ammeter in series with the AGC loop, the local oscillator of the receiver is tuned for the minimum AGC current, typically 1 uA for every 100 kilometre from Rugby. The minimum AGC current would ensure a very tight AGC control (i.e. gain is adjusted such that the output signal remains within pre-defined limits). This is essential because without tight AGC control, the received signal levels would fluctuate significantly in the adverse 'weather' conditions which would affect the synchronisation accuracy. At LUT and RAL, the AGC current was adjusted such that minor signal deterioration would not affect the output clock signals. In practice, a pair of Rugby clock receivers was used to calibrate the relative synchronisation accuracy between the SEC signals. At LUT and RAL, the relative accuracy was found to be around 1 millisecond. The relative accuracy was checked by sending a chronologically timestamped packet from one TGR to another TGR at the start of every second. Under light loading conditions, the variations in the one-way delay calculated from the timestamped packets determines the relative synchronisation accuracy. The delay measurement experiments which involved only one Unison site (local experiments) were carried out by synchronising the TGRs from a single Rugby clock receiver. This eliminated any synchronisation error in the local delay measurement experiments.

It is assumed that the propagation time from Rugby to RAL and LUT is almost the same since both sites are roughly equal distance from Rugby.

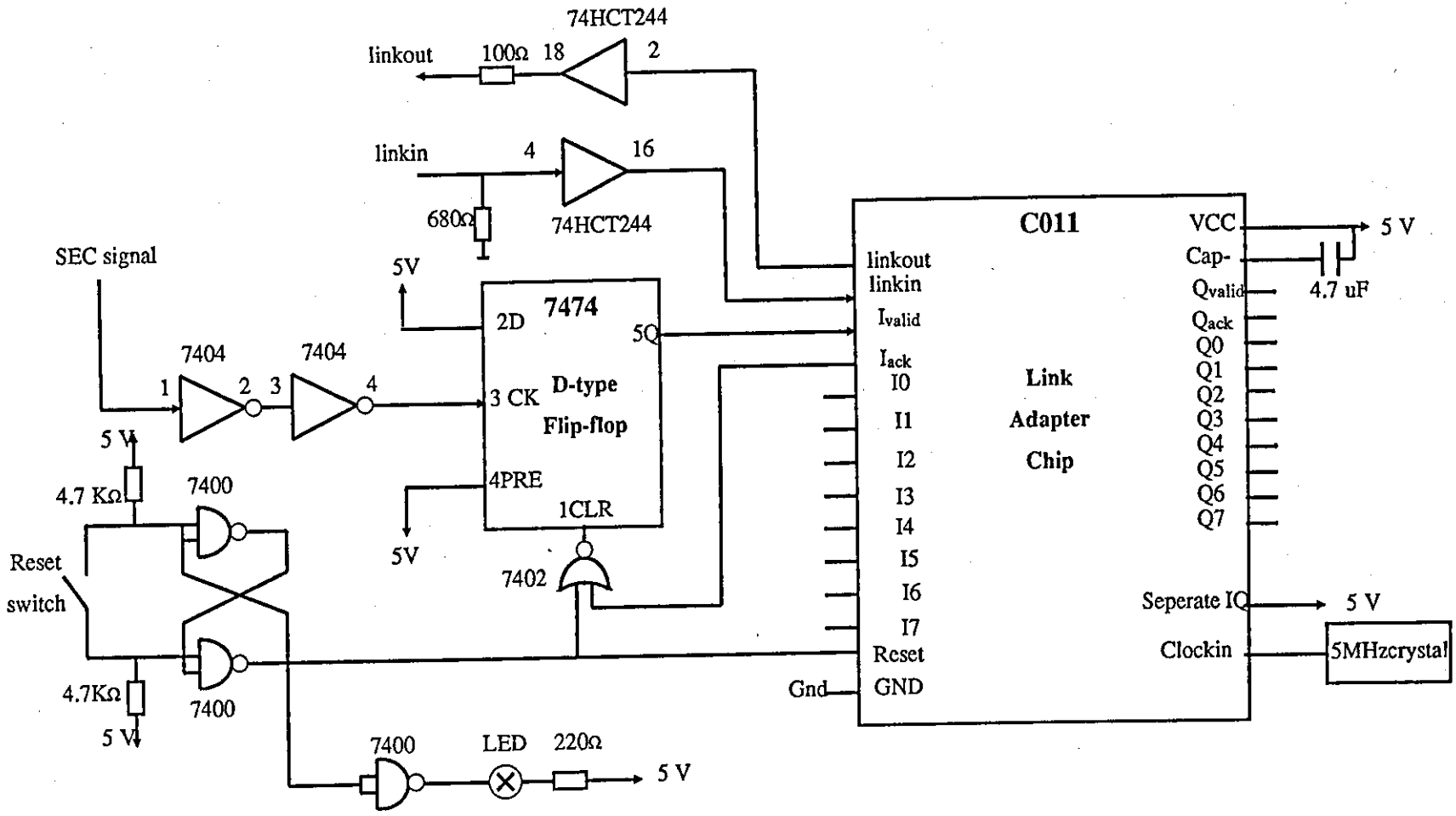


Fig. 2 : Circuit Diagram of the Interface between the Rugby clock receiver and the transputer based SBC