ANALYSIS AND MODELLING OF A COMPUTER NETWORK TO CONTROL AN X-RAY MICRO-ANALYSER

A thesis submitted for the Degree of
Doctor of Philosophy
of the
University of London
June 1979

by

Amr Ali Baghdadi

Department of Computing and Control
Imperial College of Science and Technology
University of London

ANALYSIS AND MODELLING OF A COMPUTER NETWORK TO CONTROL AN X-RAY MICRO ANALYSER

Contents

Page

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to both Professor S.J. Goldsack and Mr. M.P. Jones for their invaluable guidance and help throughout the project.

I am very grateful to Dr. B.K. Penney for the initiation of this project and his guidance and support through most phases of the project.

I am also very grateful to Professor M. Woodside for his invaluable help on mathematical analysis and queueing theory.

Thanks are also due to Department of Mineral Technology and its staff for supplying some technical and financial support throughout the project.

Finally, I would like to thank Mrs. I.E. Phare-Sinha for her patience in typing this thesis.

Introduction

This thesis presents an investigation into the design and the use of distributed computer networks in control and automation. This particular class of computer application includes process control (of steel mills, various forms of production lines, chemical reactors, etc)., industrial automation (synonymous with process control) and laboratory automation (control of, and data acquisition from, laboratory apparatus of various kinds).

The work is concerned with the development and analysis of the techniques which are to be used in implementing a computer network to control an electron probe X-ray micro-analyser.

An X-ray micro-analyser is used for quantitative and qualitative analysis of mineralogical specimens. The analysis work is repetitive and is governed by a complex set of control variables and data signals. It therefore forms an excellent vehicle for research in distributed computer control systems. A micro-analyser known as the 'Geoscan' in the Department of Mineral Technology, Imperial College, has been used for this purpose. Chapter 1 describes the operating principles of X-ray micro-analysers, their applications and their automation.

Initially, the computer control system for the 'Geoscan' consisted of a 'MINIC' (MCS) but this was limited in the speed of analysis, the data storage capability and the availability of high level programming languages for developing powerful mathematical programs for further data analysis.

As a first step, this system was enhanced by linking it to a 'GEC 4080' to provide additional storage and control. Chapter 2 describes the implementation of this system and investigates the requirements and the implementation of protocols for such control environments. The practical implementation of the system is fully described elsewhere. (0.1.)

Further investigations into the requirements of the control system for an X-ray micro-analyser, (Chapter 3) revealed the need for concurrent processing to achieve higher speeds and more powerful control. Models for the process, its operations and its controls confirmed this need and provided estimates of the interprocess data traffic and flow patterns. It was therefore necessary to investigate distributed processing networks suitable for this control application.

A study of broadcast networks (mainly loops and bus type networks) leads to the introduction of a set of criteria for the evaluation of broadcast networks in control environments (Chapter 4). Surveys of bus networks (e.g. ALOHA, Ethernet) and loop networks (e.g. Pierce, Newhall, DLCN) were carried out (0.2). Within the economic, reliability and performance requirements of the application, a loop network with broadcasting capability was chosen for the control of the X-ray micro-analyser.

Mathematical models that compare the different loop access methods (Chapter 4), (0.4) and (0.5), tend to show that 'delay buffer insertion' techniques give the best delay performance compared to slotted loops, control passing and polled loops.

A mathematical model, (Chapter 5), investigates the various possibilities of utilizing 'delay buffer insertion' techniques. The study leads to the introduction of a novel method for loop network communications which provides optimum overall message delay; the optimum delay method, 'ODM'. At certain traffic patterns the performance improvement in delay over conventional buffer insertion techniques, such as 'DLCN' (O.3), approaches 80%.

The method is based on sample averages of message durations measured at each node on the network. Each node then dynamically assigns priority for the use of the communications channel, either to its own generated traffic or to the loop traffic passing through, according to an assignment rule which produces an optimum average delay over the network.

Message priorities on a loop in a control environment could be a necessity. Chapter 6 introduces a mathematical model of a loop network with priority classes of messages, based on the optimum delay method 'ODM'. At each level of priority the total average delay is minimized. The model is based on a modified conservation law for priority queues. A new assignment rule was deduced (to be used by each node independently) for optimum scheduling which produces minimum average delay.

The actual network to be used for the control of a new X-ray micro-analyser will be based on four micro computers (DEC, LSI-11). Standard specimens are used in estimating the traffic rates on the proposed network (Chapter 7). The performance of the network has been examined on the basis of the mathematical models developed in Chapters 5 and 6 and the interprocesser traffic rates and patterns.

References

O.1. Beaven, C., Penney, B.K., and Baghdadi, A.A. "A Computer Network for the Extended Analysis of Data From a Computer Controlled Electron Probe Microanalyser". 15th APCOM Symposium, Brisbane, Australia, 1977.

O.2. Penney, B.K., and Baghdadi, A.A. "Survey of Computer Communication Loop Networks" Research Report 78/42 - C.C.D., Imperial College, London, 1978.

O.3. Liu, M.T., Babic, G., and Pardo, R. "A Performance Study of the Distributed Loop Computer Network (DLCN)". Proc. 1977 National Telecommunications Conference, Los Angeles, California (Dec.1977), 31:5/1-31:5/7.

O.4. Liu, M.T., Pardo, R., and Babic, G. "A performance study of distributed control loop network". Proc. 1977 Int. Conf. on Parallel Processing (Aug.1977), 137-138.

0.5. Hopper, A.
"Local area computer communication network"
Technical report No.7, University of Cambridge,
U.K., 1978.

CHAPTER I

Microbeam Analytical Instruments

1.1. Introduction to electron probe micro-analysers and their applications.

This chapter provides a general description of instruments in use for electron probe analysis; it also describes the applications in various fields and the measures taken to automate their use. The electron probe micro-analyser is described below, followed by a brief account of its applications in mineralogy.

The ability of electron probe micro-analysers to obtain mineral compositional data forms the basis of all applications of this equipment to mineralogy. Results from volumes as small as 1 cubic micrometer are possible.

1.1.1. The Electron Probe Microanalyser

In the electron probe a high energy electron beam is focussed onto a small area (~ 1 micro-m^2) of a sample. X-rays characteristic of the chemical composition of their volume will then be emitted. These X-rays contain information in terms of wavelength (or energy), which depends on the element which emitted the X-ray photon, and in the intensity of the signal which depends on the concentration of the element. The emitted X-rays are measured by one of two methods:

(a) A wavelength dispersive spectrometer (WDS): X-rays emitted from the sample are dispersed by a suitable crystal which then reflects a monochrome X-ray wave to a detector. The intensity of the diffracted X-ray lines is measured quantitatively using a counter and a signal amplification system.

(b) Energy Dispersive System (EDS):

In this system the X-ray detector is lithium drifted silicon operated at liquid nitrogen temperature. Each X-ray photon absorbed in the silicon frees a number of electrons which form a charge signal whose amplitude is a direct function of the X-ray photon energy.

The energy produced represents a wide spectrum of elements and is useful for exploring instantly all the constituents of a specimen. However, when applied to the analysis of a single mineral, it does not produce such reliable statistics as is the case with the WDS. This is due to the fact that the X-ray quanta per unit time obtained from a WDS represent information on a single element while the same number of quanta per unit time from an EDS represent all the elements at the analysed point. Thus the analysis of a single element is much more accurate due to more counts per element time in the WDS than by using an EDS. However, the number of elements distinguished per traverse in an EDS system is much greater; thus analysis is faster.

In a WDS system we could obtain the same speeds by implementing a multitude of spectrometers each set to detect a particular element.

The WDS was chosen for the system in the Department of Mineral Technology; thus in the following sections there will be more emphasis on this type of system.

1.1.2. Applications

The various types of electron probe micro-analysers have been used to produce quantitative analyses of specially prepared specimens with known elements. In addition they have been also used to identify unknown regions and particles in mineralogical specimens as an adjunct to optical microscopy. They have also been used to obtain the location of various elements and the location of phases on a specimen.

In applications depending on prepared specimens with known elements, reasonably accurate quantitative analysis to aid in the interpretation of reactions are made. This makes it possible to analyse the grain boundaries when different metals are diffused. Applications in biology (e.g. Calcium variations in bone sections) and examining sections of integrated circuits are typical.

The identification of unknown regions using electron microprobes help in the detection of one or more particles of foreign matter in materials such as electronic solid state devices (e.g. transistors or integrated circuits), which could sometimes seriously affect the operating characteristics. Also the examination of the pigment particles in oil paintings could prove whether they are genuine or fake.

Other applications are referred to in (1.1) and (1.2).

1.1.3. Automation

Many components of an electron probe micro-analyser could be operated under some form of automatic control. Microprobe analysis tends to be repetitive, where large number of analyses are made on similar materials.

The repetitive work can be carried out most rapidly under automatic control of one or more computers. Such a system could also provide the facility for storing and analysing the data obtained from each analysed sample.

The computers can be used to sense signals from the microanalyser or generate signals to control the process.

The various components of a typical electron probe instrument which could be controlled automatically are listed below, refer to fig. 1.1.

I. Input Signals (from instrument to computer):

a. Analogue Sensors:

- Shaft encoders for stage and spectrometers (1). (3 for the stage and one per spectrometer).
- Vacuum gauges (2)
- Beam/sample currents (3)
- Filament Voltage (4)
- Backscattered electrons (5)

b. Digital Signal Sensors:

- Scalers for counting X-rays (6) (one per spectrometer)

(Up to 17 input signals may be automatically detected in a 4 spectrometer electron probe micro-analyser).

II. Output Signals (from computer to instrument)

a. Switch controlled operations (device actuators)
  - Selecting the appropriate analysing crystals (7)
  - Inserting collimating slits in front of the X-ray detectors (8).
  - Controlling the aperture width setting (9).
  - Operating vacuum valves (10).
  - Switching pumps (11).

b. Mechanical Movements (motor controllers)
  - X, Y and Z axis positioning of the specimen stage (12).
  - Angular positioning of the spectrometer units (13).

c. Analogue Signal Operations (D/A converters).
  - Electron beam position (scanning coils) (14)
  - Setting the pulse height analyser (15) (one per spectrometer)

(Up to 24 output signals may be automated in a 4 spectrometer micro-analyser). By controlling the abovementioned components, the computers can perform automatically functions such as:

- Spectral scanning and searches for peaks.
- Selection of the appropriate standard specimen.
- Selection of the analysis time or counts required.
- Normalising the data derived for beam current.
- Analysing for elements selected by the operator at various points of interest on the specimen.
- Other computer output operations such as plotting results of spectral scans and peak searches, and printing results with associated statistics.

Figure 1.1
Schematic diagram of an electron probe micro analyser

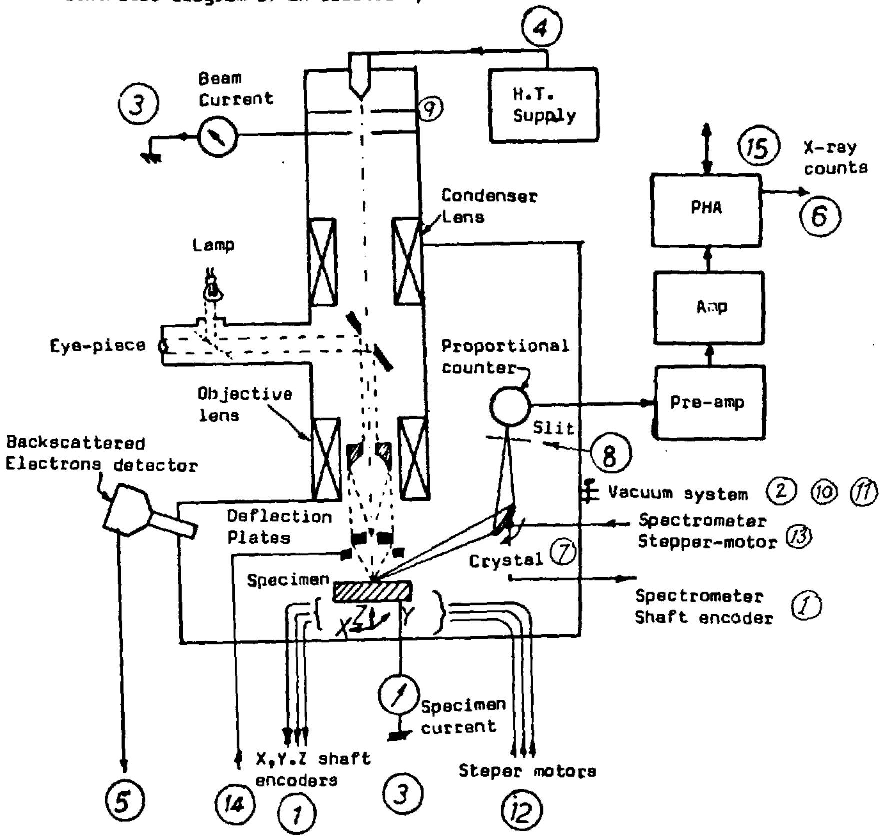
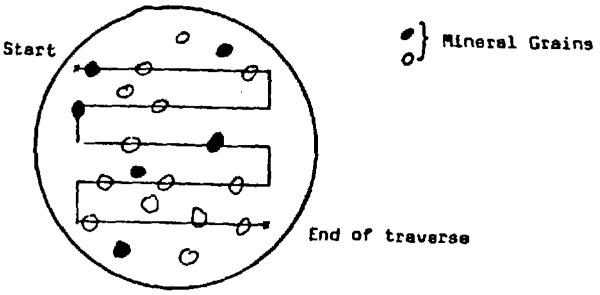


Figure 1.2
Scanning pattern on a mineral specimen



1.2. The electron probe micro-analyser in the Department of Mineral Technology, Imperial College.

An electron probe micro-analyser known as the 'Geoscan' controlled by a dedicated mini-computer, has been operating at Imperial College since 1970.

In the 'Geoscan' suitably prepared specimens are subjected to electron bombardment and the characteristic X-rays produced from the surface layers of the specimens are analysed to provide both stereological and chemical information. (Stereology is the deduction of 3-dimensional information from lower dimensional data). Figure 1.1 illustrates the principle of operation and figure 1.2 shows the scanning pattern normally employed.

The techniques of deriving the required mineralogical data from the experimental results have been fully described in Jones and Gavrilovic (1970) (1.3) and in Barbery (1974) (1.4).

In this system a particular mineral species (known as a 'phase') is identified by the simultaneous reception of appropriate signals from the two available spectrometers and the specimen current integrator. The specimen, which is held in a stage, is driven by two stepper motors underneath the beam. As the specimen moves under the electron beam, different mineral phases are encountered and the frequency distributions of intercept lengths on the individual phases form the basic secondary data from which the mineralogical parameters are deduced.

Many of the mineralogical materials that are examined by this method are extremely heterogeneous and the total length of scan on a 25 mm diameter specimen may be around 1 m. The scanning speed varies from 100 μm/s to 450 μm/s and a set of six specimens may take up to 15 hours to examine. In order to achieve the maximum utilisation of the probe, automatic unattended operation was essential. A small dedicated mini-computer having a storage capacity of 8K 8 bit bytes (a 'Minic' manufactured by MCS Ltd) was chosen for the control, and in addition it was able to undertake limited data reduction.

1.2.1. Data Analysis

A major proportion of the work for which the measurement system was required is concerned with the determination of the mineralogical parameters of rock specimens. These parameters must be known for the design and monitoring of mineral preparation and treatment processes.

The specimens consist mainly of 25 mm diameter flat discs polished on the upper surface and coated with carbon to a thickness of approximately 20 nm. This coating is necessary to prevent build-up of charge on the surface.

Most analyses are concerned with complex aggregations of mineral species of which the grain size could vary from less than 1 μm to 10 mm or more. Because of the method of selecting the sample, the distribution of grain sizes of different mineral phases may be treated statistically, except when pronounced systematic features are found, as shown by Barbery (1974), (1.4).

Under computer control the specimen is moved under the electron beam in steps. A real time clock having a 10 ms period governs the data collection process. Output counts from each of the spectrometer X-ray detectors are accumulated in separate registers in the interface. The separate totals together with the time integral of the specimen current for the same period are then passed to the 'Minic' memory for subsequent processing.

The scanning pattern is adjusted so that the probability of the path crossing a grain more than once is small. Under these conditions the error introduced by counting larger grains twice is also small.

In the most usual mode of operation the spectrometers are set at the X-ray wavelengths corresponding to major elemental components of the mineral for which the analysis is sought. Preliminary work establishes the expected count range for each element and the program then establishes the presence of the particular mineral within specified statistical limits.

From these data, frequency distribution sets of the intercept lengths of the beam passage across the grains of the various mineral species in the specimen are determined.

The total path length covers ranges from 0.2 to 1 m for each specimen, with the mean intercept length varying from less than 20 μm to greater than 200 μm. The total number of intercepts ranges up to 10^5.

In this initial system the data is processed as it is received, primary information being discarded owing to the storage and processing limitations of the mini-computer. Discarding the primary information, however, means that important interrelations between mineral species could not be determined thereby reducing substantially the utility of the system.

1.2.2. The Automation of the Microprobe

Fig. 1.3. shows the hardware connections to the controlling computer, the 'Minic'. Table 1.1 shows the input and output signals between the GEOSCAN and the Minic computer.

Output signals are defined as those passing from the Minic computer to the Geoscan. Input signals pass in the reverse direction.

The stepper motors are driven by an 8-phase waveform derived from an input pulse supplied by the computer. Each step turns the motor shaft through 3.75° (96 steps/rev) which in turn causes the specimen stage to move 1 μm.

1.2.3. The computer system

The 'Minic" has an 8-level vectored priority interrupt system. Each of the peripherals and the real time clock is connected to different interrupt levels, the clock has the highest level of interrupt. All interfaces, shown in fig. 1.3. are connected either to the 8 bit input bus or to the 8 bit output bus. The software operates at several levels of priority. Synchronisation of the system is achieved by arranging for the clock, which is connected to the highest level of the interrupt system, to cause an interrupt every

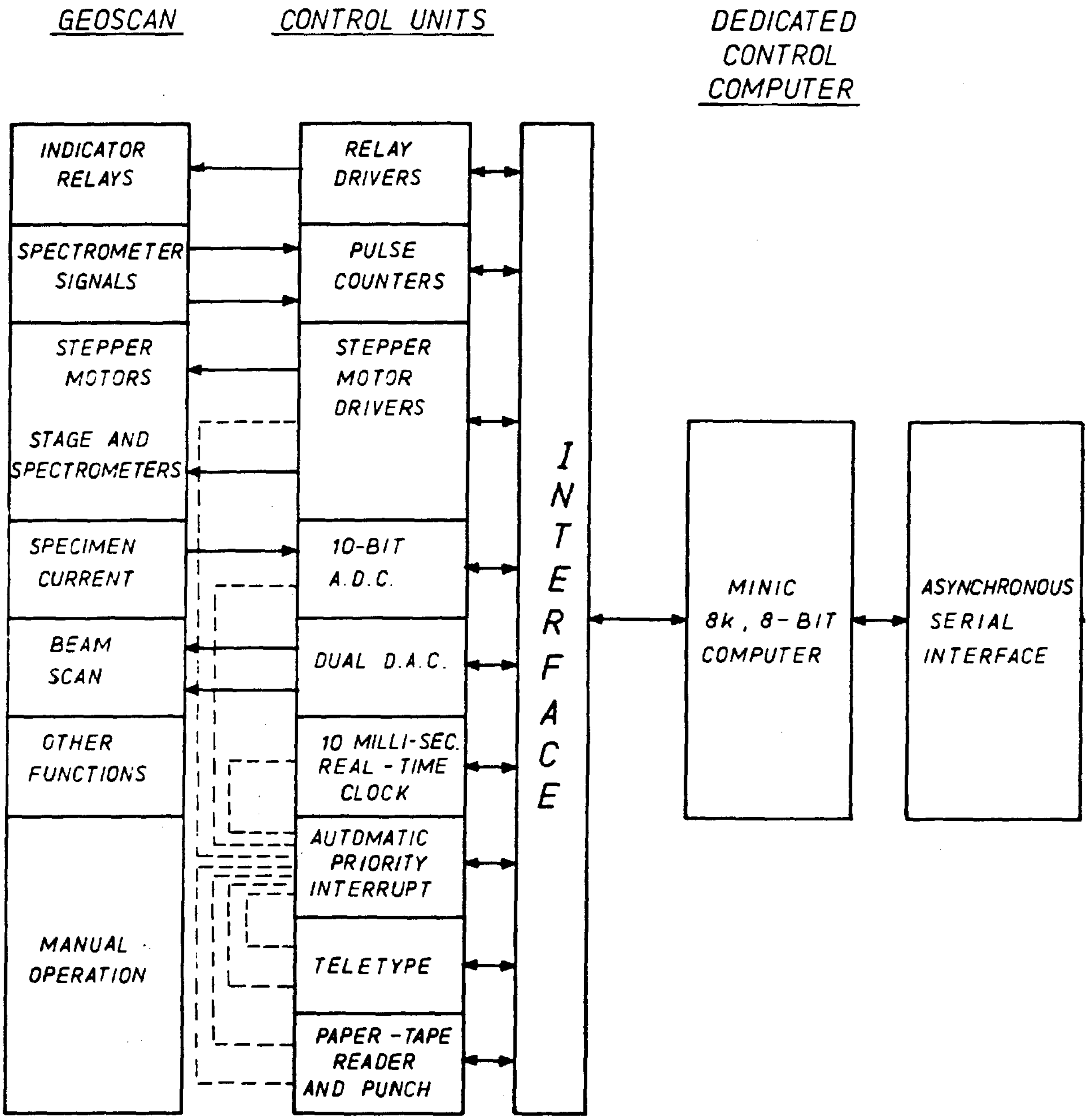


Figure 1.3

The Geoscan and Minic hardware interfaces

10 m.sec., this having been calculated as the minimum time needed to accumulate statistically significant sums in the X-ray counters. On interrupt, the corresponding service routine reads and clears the pulse counters and then deposit the consequent values into appropriate locations in main store.

The analysis program is not connected to the interrupt system and it runs continuously. This program takes as input the data which is collected when the clock causes an interrupt. In the majority of cases the analysis is completed before the next set of data arrives at the next 10 m.sec interval. The possibility exists, however, that the analysis is not completed at this time, and for this reason the clock service routine inspects a completion flag which is set by the analysis routine. If this flag has not been set, indicating that analysis is still in progress, then the input data is simply discarded. If it is set, the input data is stored normally and the completion flag is cleared. The analysis program on completion of the current data set, sets the completion flag and then waits until the flag has been cleared before starting the analysis of the next data set.

The program at the beginning was able to identify a single selected phase, and the measurements were confined to this phase. At the end of scanning each specimen, or on operator request through the teletype, the analysis program in this system prints on teletype the particle distribution, the number of particles and the first four moments as described by Jones, Beaven and Shaw (1.5) (1972). Similar data is dumped onto paper tape every 100 particles giving

ITEM	FUNCTION	DIRECTION OF SIGNAL	LIMITING FACTORS	RESOLUTION	SIGNAL	PULSE RATE	FUNCTION LIMITS
Stepper Motor	Left Hand Spectrometer	To Geoscan	Speed Motor "Pull-in" Capability	0.002°	Drive signal	450p/s max.	0.9°/s
Stepper Motor	Right Hand Spectrometer						
Stepper Motor	Specimen Stage X Direction		Position Mechanical Tolerance	1 µm	Direction Signal '1' or '0'		0 to 450 µm/s
Stepper Motor	Specimen Stage Y Direction						
Electronic Beam Scan	X Direction	To Geoscan	"Minic" Processing time;	0.5µm	Analogue Signal Converted from 10 Bits of 16 Bit word	16 Bit in 25µs	$5\text{x}10^4$ µm/s
Electronic Beam Scan	Y Direction		Deflection Current Stability				
Phase Indicator Relays	Left Hand Spectrometer	To Geoscan	Relay Operating time;	±1count/s	ON/OFF		150 count max.
Phase Indicator Relays	Right Hand Spectrometer		X-Ray Statistics Amplifier Noise and Drift				500nA max
Phase Indicator Relays	Specimen Current			5nA			
X-Ray Counter	Left Hand Spectrometer	To Minic	X-Ray statistics: Counter 'dead time'	±1count	8 Bit staticiser	15000/s max.	190 count max.
X-Ray Counter	Right Hand Spectrometer						
Current Integrator	Specimen Current	To Minic	Noise and Amplifier drift	5nA	10 Bit A.D.C.		500nA max.

Table 1.1

Geoscan/Minic (Front end)
System Signals

the possibility of later, off-line analysis on a more powerful computer.

Limitations on the analysis arose from several factors, these mainly being the size of the main store, the time available for analysis, the programming difficulties which existed when working with a primitive assembler language for such a small computer, the lack of backing storage and the limited processing speed.

1.2.4. Improvements to the analysis software

The initial program controlling the GEOSCAN and analysing its data was designed to obtain measurements on single mineral species. Consequently when a specimen contained many mineral species, a complete analysis required a number of sequential examinations.

New programs were developed which enabled the GEOSCAN to measure up to 5 mineral species simultaneously and could easily be extended to detect 9 species.

The multiple phase detection program detects the individual phases from various combinations of X-ray signals. In the earlier, single phase, detection program, a 'filter' subroutine 'dropped-out' (and, so, neglected) the effects of small cracks, pores and polishing artifacts on the specimen. Multiple-phase specimens normally show very rapid, but genuine, phase changes which would be ignored by the dropout filter. Consequently, a 'dynamic dropout filter' has been incorporated which neglects cracks, pores and polishing artifacts, but still remains to measure rapid phase changes.

This filter routine is switched off when the rate of change of phases exceeds a certain threshold for a certain length. The program is designed to show the operator on a light indicator panel the phase currently being detected. The computer produces the statistical data on paper tape for further analysis on a larger computer. At a later stage, as will be shown in Chapter 2, the computer passes this data 'on line' to a larger computer in the Department of Computing and Control through a communications link. This software has been used to produce valuable information from an iron ore and the results have been published (1.7).

References

1.1. Birks "Electron Probe Micro Analysers", John Wiley, 1963.

1.2. Smith, D.G.W. "MAC Short Course in Microbeam Techniques", Mineralogical Association of Canada, Edmonton, 1976.

1.3. Jones, M.P. and Gavrilovic J. (1970), "Automatic Quantitative Mineralogy in Mineral Technology", Rudy 5: pp. 189-197.

1.4. Barbery, G. (1974) "Determination of Particle Size Distribution from Measurements on Sections", Powder Technology, 9: 231.

1.5. Jones, M.P., Beaven, C.H., and Shaw, J.L. (1972) "The Measurement of Stereological Parameters for the Mineral Industry by a Computer Controlled Electron Probe Micro Analyser". Proceedings

Industrial Measurement and Control by Radiation Techniques Conference, IEE, London.

1.6. Baghdadi, A.A. "Research Report - July 1977 - Dept. of Min. Tech., Royal School of Mines, Imperial College, London.

1.7. Jones, M.P. and Horton, R. "Recent Developments in the Stereological Assessment of Composite Particles by Linear Measurement". Paper No.52, XIth Commonwealth Min. & Metall. Congr. - Hongkong, May 1978.

CHAPTER 2

First Steps Towards a Computer Network for Control

Because of the limitations of the processing power and storage capabilities of the 'Minic' computer (which initially controlled the Geoscan), primary information was discarded. This meant that important interrelations between mineral species could not be determined and therefore the effectiveness of the system was reduced.

A laboratory automation project was initiated in the Department of Computing and Control at Imperial College based on a GEC 4080 computer. The GEC 4080 was intended to act as a central computer in a star-shaped network, with other laboratory computers linked to it through communications lines. The 'Minic" was the first node to be connected to this network, (see refs. (2.1), (2.2)).

The system is shown in block form in fig. 2.1. There has been three main stages of computing. First, the MINIC serves as a "front-end" to the Geoscan. Secondly, data from the MINIC is passed to the GEC 4080via communications lines and the latter machine processes the data and stores it on disk. Finally, more extensive analysis is performed by transmitting data under the 4080 program control via a synchrononous communications lines to the Imperial College CDC 6400, or via an asynchronous communications line to an IBM/135.

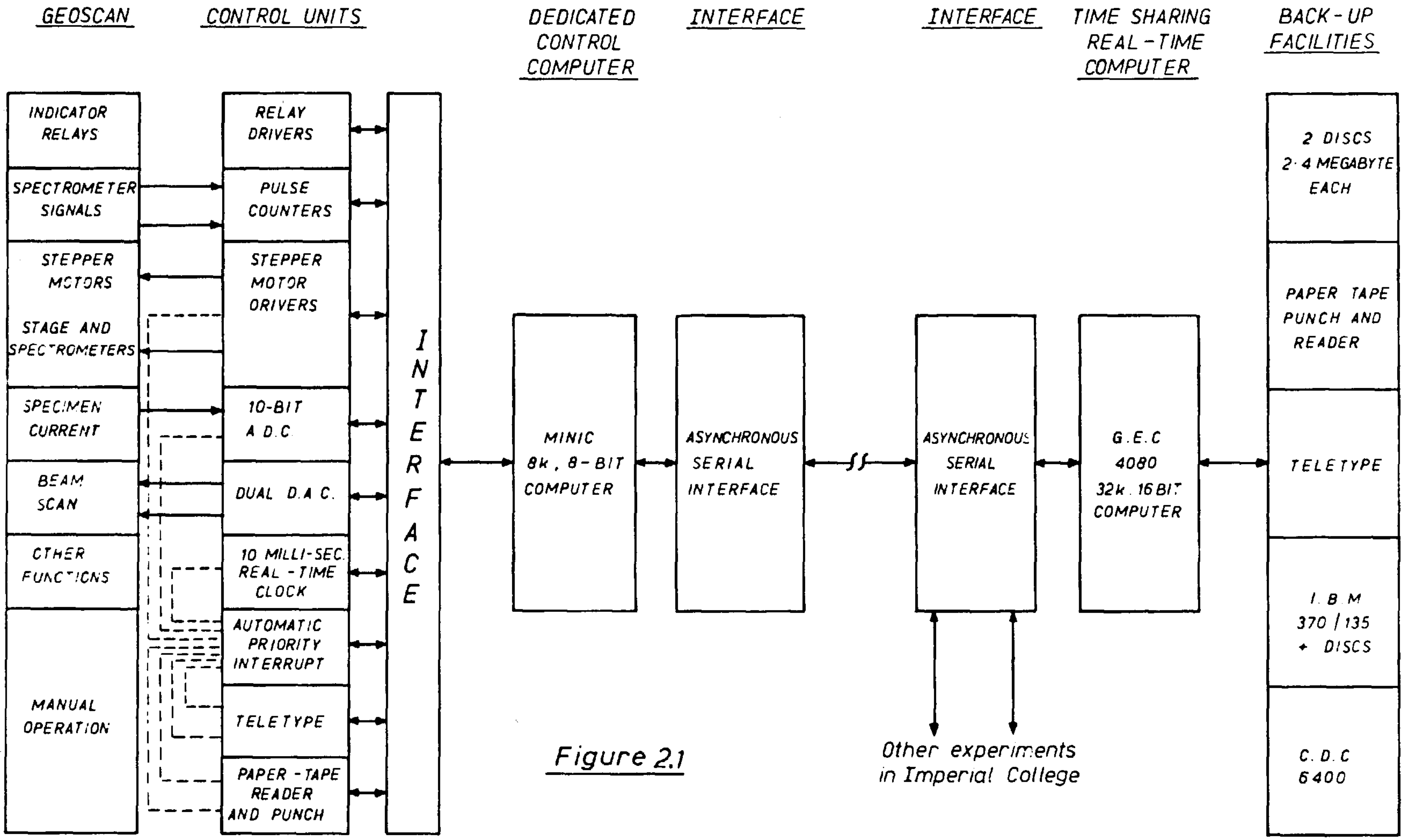


Figure 2.1

GEOSCAN IMAGE ANALYSING SYSTEM

2.1. System Requirements

The MINIC controls the specimen in the Geoscan by two stepper motors in steps of 1μm at 10 m.sec. intervals. The data needed for analysis at each step is represented as information in terms of byte length as shown in table 2.1.

TABLE 2.1

Primary Data Collected from the Geoscan

Item	Length (byte)	Max value	Data rate (Byte/s)
X-position	3	8×10^4 μm	300
Y-position	2	2.4×10^4 μm	200
A-spectrometer	1	150 counts	100
B-spectrometer	1	150 counts	100
Specimen Current	1	500 nA	100
Σ	8		800

In table 2.1, the X and Y values represent the 2-dimensional position of the specimen under the electron beam. The A and B spectrometers read the quantized characteristic X-rays, produced by electron bombardment of the specimen, into two 8-bit counters. The specimen current is the leakage current from the specimen.

As shown, the basic primary data rate is of the order of 800 bytes per sec. (or 6400 bits/sec). This rate is beyond the capacity of the original MINIC and much of the information is discarded. With the larger storage capability of the GEC 4080 (64 K bytes of main store + 2 x 2.4 M.byte disk stores), however, it was favourable to keep most of the primary data and process it after storage. The limiting factor then became the data handling rate of the interface between the MINIC and the 4080, which is 4800 bits/sec. This meant that an initial compression of the amount of data had to be performed by the MINIC before transmission to the 4080. These compressed data are to be held on disk files in the 4080 or alternatively on the CDC-6400 or the IBM.370/135.

2.2. Data preparation and reduction

To compress the amount of data for transmission to the 4080 the MINIC has to process the data collected from the Geoscan to reduce the inherent redundancy.

This raw input data is analysed to detect a specified phase and measure its intercept length. The type of phase and its intercept length forms a data message which is placed in a circular buffer (fig.2.2) for later transmission to the GEC 4080. Thus the only data transmitted to the 4080 are those formed whenever a new phase is met. Table 2.2. shows that the resulting data rate is typically 200 times less than the primary data rate given in table 2.1.

TABLE 2.2.

Data Rates and Values after Reduction

Item	Numerical value	Length (Byte)
Function code	0-8	1
Length of phase or of Y jump	$1\text{-}10^6$	4
<u>Data rates</u>:		
Phase/s (maximum)	10	
(typical)	1	
Byte/s (maximum)	40	
(typical)	4	

Total storage required: 4×10^5 Byte maximum
4×10^4 Byte typical

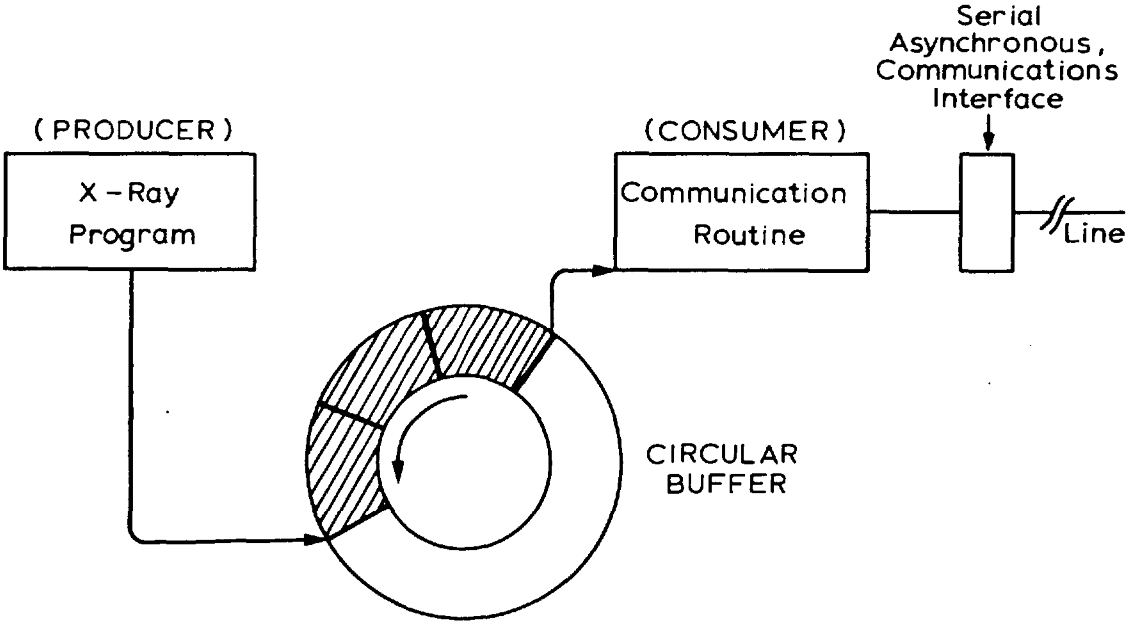


Figure 2.2. MINIC COMMUNICATIONS SOFTWARE

The Minic phase-measuring program drives the entire mechanism of the Geoscan and processes the collected signals in order to detect a new phase. The latest version of the program is designed to detect up to 9 different phases. The communications routine is responsible for the message exchange (protocol) on the Minic side.

However, the Minic program and the communication software act as a producer and a consumer, each running independently and dealing with a circular buffer which contain the data to be transmitted. If the buffer is filled, the Minic stops moving the specimen and discards any collected data until the circular buffer begins to be emptied. The consumer always tries to empty the circular buffer independently from the producer.

Data are generated by the MINIC program at ten m.sec. intervals (Table 2.2) yet meaningful data is only encountered whenever a new phase is found on the specimen; this occurs at an average interval of 1 second.

2.3. The Communication Link

The two computers (the Minic and the GEC 4080) are linked by a serial half-duplex asynchronous communications line over a distance of about 500m. The line is allowed to run at the maximum speed that the Minic interface is able to support, i.e. 4800 bits/s. Because the link runs through an electrically noisy environment, a balanced line driver is supplied at either end of the line.

Communications took place using 'messages'. A data message consists of a header, a data section and a trailer. Other messages contain only a header and a trailer (zero length data) for control purposes. The message format is shown in figure 2.3. The source and destination codes are not used in this system but are included to cope with the expected expansion of the network and of its facilities. Communication proceeds by exchanging messages according to a set of rules known as the protocol.

Header	Start of header
	Message type
	Length of text (0-127 bytes)
	No. of Message (modulo 128)
	Source Code
	Destination Code
Data	Up to 127 bytes
Trailer	End of message character
	Checksum

Figure 2.3. Message Format.

A communications protocol was implemented to ensure error-free transmission of data between the MINIC and the 4080. It was essential that the protocol should allow the 4080 to transmit control messages to the MINIC in case of remote control operation, and allow the re-initialization of the system with the minimum losses of data in cases of failure. Also, the ability to perform error checking and error correction functions during data transmission was of primary importance.

Several standard protocols such as X-25, HDLC and Decnet protocols (2.3), (2.4) and (2.5) were examined but they proved to be inadequate for this system due to their costly overheads and due to the storage limitations in the MINIC (and small automated systems in general). Hence a new protocol which takes into account the previous precautions was designed.

2.4. The Communication Protocol

A local computer network, used for control purposes and based on small mini or micro-computers, can provide economy and higher efficiency in a traditional centralised control environment in which control and monitoring lines are brought to a central computer (2.7).

The information (referred to as data) generated at a certain control point is transmitted to another processor for analysis. The amount of data transmitted can be reduced by partial analysis and formatting, and possibly by using data compression techniques. In addition, the memory of the mini or micro-computers can act as a buffer to smooth out the bursts of data produced. This will result in lower bandwidth requirements and may permit the use of serial rather than parallel transmission lines. Distributed processing systems of this type require very reliable communication between processors, so a communication protocol has to be implemented in the computers to handle the message exchange.

2.4.1. Communication Protocol Requirements

Although the initial implementation consisted simply of the MINIC and the 4080, it was taken into consideration that the design of the protocol should match the requirements of other small distributed systems for control.

The main requirements of the communication protocols for small mini/micro computer-based distributed systems are as follows:

Simplicity - the protocol should be easy to implement on a mini/micro computer with its limited memory and processing power;

Flexibility - the protocol should be independent of the transmission methods and network configuration;

and

Reliability - industrial environments are likely to be electrically noisy, requiring automatic error detection and correction techniques.

Several other communication system requirements should also be satisfied:

1. Addressing Capabilities:

   In order to cater for various network configurations, every "packet" must identify both source and destination node addresses.*

2. Error Detection and Correction:

   The protocol has to be able to handle the following errors:

---

*A packet is a sequence of control and information characters used to enclose some information to be transmitted between computers via a communications line (sometimes referred to as messages).

- Corrupted Data due to interference:

In an electrically noisy environment errors may appear in the data packets. These errors can be effectively detected with a cyclic redundancy check (CRC) or a longitudinal check (2.8) and correction is possible through the use of the automatic repetition (ARQ) method (2.8). Upon the arrival of each error-free packet, the receiver returns a positive acknowledgement. If the sender does not receive the acknowledgement within a specified period the packet is re-sent.

- Loss of a Packet:

A packet may not reach its destination because the line or receiving node is temporarily out of action, or the destination address is corrupted. The ARQ method overcomes this problem by ensuring that the sender retransmits the packet.

- Incomplete Packets:

If any packet was partially lost due to a line or node failure the receiver should be able to recognize that it is faulty. This is ensured by recognizing the end of the packet as well as its beginning. The end of the packet is located either by using the length field as in the DDCMP protocol (2.5), or by a special end of packet flag as in the HDLC protocol (2.4) or the bipolar violation technique (2.8).

However, the flag may become corrupted, or the sender may abort the packet or specify the wrong length. Thus the receiver must also initiate a time-out at the start of a packet and, when failure to complete a transmission is recognized, should revert to the mode where it could accept more incoming packets. Also it could send a negative acknowledgement requesting retransmission of the last packet.

- Packet sequence error:

If an acknowledgement is lost, or does not arrive on time, the packet should be sent again. All packets must therefore contain a sequence or identification number, so that the receiver can detect duplicates.

2.4.2. Existing Protcols:

Some of the commercially available protocols are DEC's DDCMP (2.5), IBM's SDLC (2.6) and ISO's HDLC (now part of CCITT X25 protocol (2.4)). All these protocols were designed for point-to-point communication or in the case of SDLC and DDCMP, multipoint with a single master which polls slave nodes. The packet headers do not allow for both source and destination addresses. In addition, DDCMP is the only one which specifies a length field in the packet header, the others rely on bit insertion to ensure that the start and end of a packet flags are unique. The bit insertion technique is used in serial transmission to prevent the end of packet flag (01111110) from occurring in the packet (2.3).

On transmission a "0" is inserted after all sequences of 5 contiguous "1" bits. On receiving, a "0" which follows 5 contiguous "1" bits is discarded. Any detected sequence of six "1" bits is the flag or an error.

The X25 protocol uses the concept of a virtual call between two processors, i.e. a point-to-point circuit which may traverse several nodes. It is not necessary to specify source and destination addresses as these can be obtained from the virtual call number, but the nodes then have the time overhead of translating the call number into actual addresses. There is also the overhead of messages required to set up and clear a virtual call for a short transaction, which in a control application may consist of only two messages, a request and a reply.

In general, the abovementioned protocols are far too complex with many unnecessary message types, not needed in a local network of cooperating mini and micro-computers.

2.4.3. A Protocol for Industrial Applications

A protocol consists of 4 main levels:

1. User level
2. Message level
3. Communications level.
4. Transmission level.

A node in a computer network may be defined as a station (which could be a computer, a terminal or an instrument) plus a communication interface (which could be another computer. This twin processor system could be substituted by a single processor in simple cases.

A communications system can be viewed as a number of independent levels with a protocol defining the rules for sending messages between the corresponding levels in different nodes, and the interface between adjacent levels in the same node.

The user level:
This is application dependent and provides a uniform interface for exchanging packets with remote processes or programs. The process residing in the Minic is the X-ray program. The user protocol generates packets of limited length in order not to delay other nodes from accessing the network. Also the buffer size has been optimised using the circular buffer.

The message handling level:
This level generates and removes the message headers, manages the buffer allocation and handling ("consumer" in the MINIC case), handles timeouts if required by user level; is responsible for routing if the system implies a routing mechanism such as store and foreword type networks.

The message format implemented in the MINIC/GEC 4080 system is seen in figure 2.4. and message types are shown in Table 2.3. Messages are either user level commands or communication level acknowledgements or commands (used by the communication level).

The communication level:
This level recognizes the start and end of messages, makes CRC or longitudinal error checks. It is also responsible for the automatic request for resending a packet (ARQ) and its timeouts. Thus, ultimately, this level hands error free packets to the 'packet handling

level'. This level can issue packets of its own for acknowledgements and requests to retransmit packets with errors. This level is also responsible for resolving contention between nodes in random access networks.

The transmission level:
depends on the communications lines technology and is responsible for bit and character synchronization.

TABLE 2.3.

Types of Messages in the MINIC/GEC protocol.

1. User level messages
(Data field $\neq 0$)

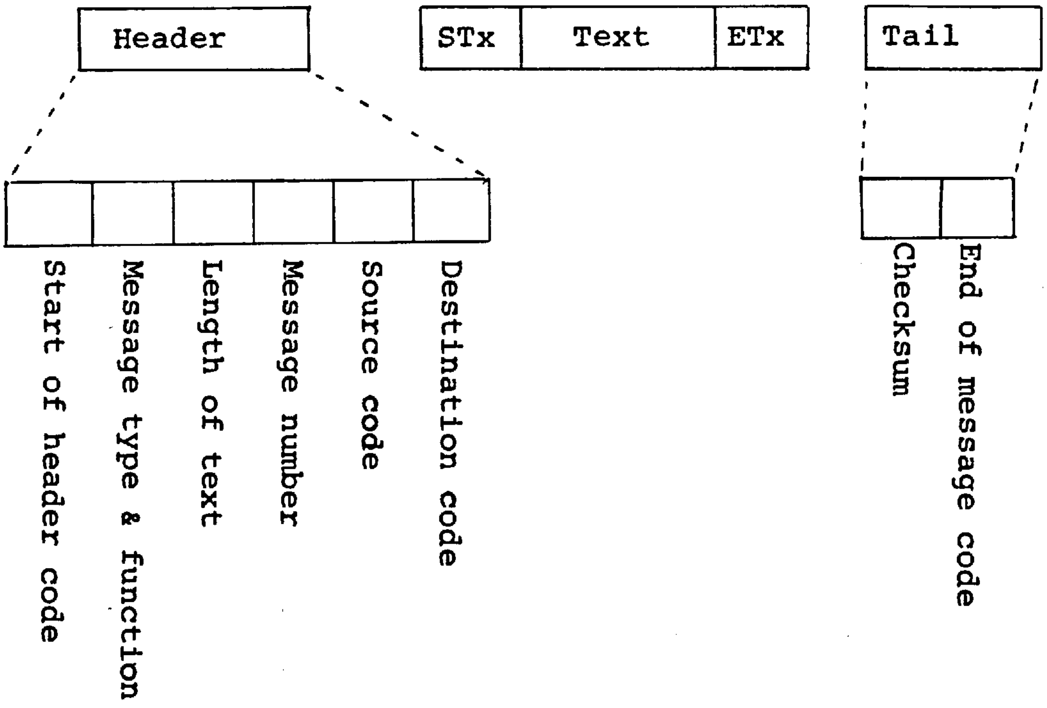
Message	Function
CONNECT	For initialization and reinitialization
CONTROL	Carry information for remote control
DATA	Carry user information for processing or storage
CONNECTED	Positive acknowledgement for initialization

2. Communications level messages
(Data field = 0)

Message	Function
ERROR	Indicates type of unrecoverable error.
MESSAGE RECEIVED	Positive acknowledgement
RE-SEND	To resend a lost or corrupt message

2.5. Conclusions

Although the general principles of the protocol designed above were designed as guidelines for the laboratory automation network protocol in Imperial College, they were oriented to serve a more general type of application. It is not a protocol that could suit all types of control networks or all types of network configuration. However, members of the same research group have managed at a later stage to produce a more global protocol which suits most network configurations (2.9) and which is directed towards control application.



STx = start of text

ETx = end of text

Figure 2.4.
Message format for the MINIC/GEC 408 Protocol.

References

2.1. C. H. Beavan, B. K. Penney and A. A. Baghdadi, "A Computer Network for the Extended Analysis of Data from a Computer Controlled Electron Probe Microanalyser". 15th APCOM Symposium, Brisbane, Australia, 1977.

2.2. A. A. Baghdadi, "Design and implementation of a Node in a Laboratory Automation Computer Network", M.Sc. Thesis, London University 1976.

2.3. CCITT 1975, "Recommendation X.25", COM VII, No.262E.

2.4. Sloman, M.S. "An Explanation of the CCITT X.25 Recommendation for Accessing a packet switched Network". Research report 78/34 - CCD, Imperial College, London (1978).

2.5. Digital Equipment Corp. 1974, "Specification for digital communications message protocol (DDCMP)".

2.6. Donnan, R.A., Kersey, J.R. "Synchronous Data Link Control: A Perspective", IBM Systems Journal No.2, 1976, pp. 140-162.

2.7. Oswald, H. 1976, INFOTECH Distributed Systems, 363-374.

2.8. Martin, J. 1970, "Teleprocessing Network Organization", Prentice Hall.

2.9. Sloman, M.S., Penney, B.K. & Marsland T. A. "A communication protocol for distributed microprocessors", IEE publication no. 153.

CHAPTER 3

The Control Processes in an Electron Probe Microanalyser

3.1. The Requirements and Needs for Enhancing the Automation and Speed

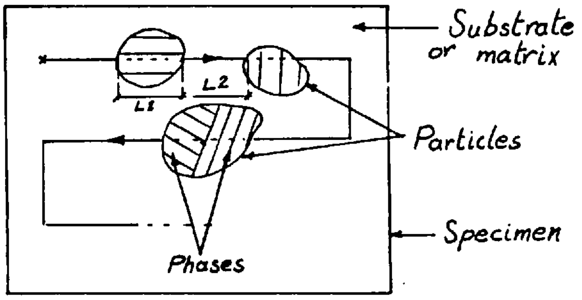


Figure 3.1.

The conventional operation of quantitative mineral analysis through line traverses, fig. 3.1, relies on collecting X-ray counts accumulated during a finite time period from a multiplicity of spectrometers. The time period is limited by the duration of each individual pulse and by the minimum number of counts needed to identify the mineral spectrum (up to 190 counts). This time interval is found to be about 10 m.secs in all electron probe microanalysers. This means that a traverse of 1 metre on the specimen with 1 micro-metre increments (i.e. a speed of 100 micro m./sec) may take as much as 2.8 hours for a complete analysis. A set of 6 specimens will require over 16 hours for analysis.

It was found that the uninteresting analysis regions (i.e. the substrate or matrix), which constitute about 75% of the specimen could be detected by a direct measurement of the specimen current and/or the backscattered electrons. If the specimen is driven in these regions at a much higher speed, e.g. 100 micro m/sec., the analysis time of 1 metre traverse across a specimen is reduced to 0.9 hrs. A higher speed along the matrix (2000 micro m/sec) will still further reduce the analysis time to 0.79 hrs. These speeds will reduce the analysis time of a set of 6 specimens to 5.4 hrs. and 4.8 hrs. respectively.

The following sections deal with the system parameters which will allow us to control the system in such a way in order to achieve the required speed and analysis capabilities.

3.2. Description of the process and the control

The system under consideration can be broken into three identifiable parts which could be modelled separately:

1. The controlled process - which comprises a set of variables, some of which are measured and some of which are controlled.
2. The operation of the controlled process - this is concerned with the rules governing the process variables to perform a certain function.
3. The control system - which manages the operation of the system by measuring, manipulating and controlling the process variables.

Models for each of these concepts are described for an electron probe microanalyser. The control system itself can be broken into several control processes dedicated to parts of the operation yet coordinating and communicating to accomplish the overall operational task. This coordination between the control processes is also described in terms of traffic patterns and the rates of information flowing between them (section 3.3 and Chapter 7).

Consequently, these control processes are allocated to controllers which could be processors, human operators, or mechanical or electronic controllers.

The various types of processor networks are sought to act as the control system (Chapter 4). The requirement for high speed data flow between the various control processes will lead to the choice of a suitable computer network, as will be seen in the following chapters.

3.2.1. Model of the controlled process variables

The controlled and measured variables in an electron proble microanalyser have already been listed in (1.1.3). However, further classification of these variables clarifies their operational involvement. The variables can be regarded as members of two sets.

The first set of variables is involved in the start-up and the close-down operations of the machine.

The second set of variables is involved in the steady state operation and analysis.

The two sets overlap and the common variables in the intersection are used in both the start-up/close-down procedures and the steady state operation.

The two sets of variables are listed below along with the type of components they relate to (mechanical, switches or electrical). Some of the common overlapping variables will be referred to in the control processes. Variables related to mechanical components are indicated by '*', switch operation related variables are indicated by '**', otherwise they are related to electrical components.

I. Variables used in start-up/close-down operations:

a. - Measured variables:

- Filament heating current
- HT voltage (operating the electron gun)
- Vacuum gauge

b. - Controlled variables:

- Water taps for cooling the pumps (if water cooled).*
- Vacuum valves. **
- Pumps (rotary and diffusion type pumps).**
- Pulse height analyser (PHA).
- Aperture width.**

II. Variables used in steady state operations

a. - Measured variables:

- Beam Current
- Shaft encoders for stage position (x, y, and z positions).*
- Shaft encoders for each spectrometer.*
- X-ray counters (for each spectrometer).
- Specimen (or sample) current.
- Back scattered electron current.

b. - Controlled variables:

- Selection of appropriate analyzing crystals.**
- Electron beam positioning (scanning coils).
- X, Y and Z axis positioning of the specimen.*
- Angular positioning of the spectrometer units.*
- Inserting collimating slits in front of the X-ray detectors.**

3.2.2. Model of the Process Operation

The operation of the electron probe microanalyser consists of two sets of procedures. The start-up/close-down operation of the instruments occurs only at the beginning and the end of all operations or in the event of a filament, or other, failure. The steady state normal operations consist of many procedures depending on the type of analysis the instrument is performing. One procedure, which is the major interest in the Department of Mineral Technology at Imperial College, is concerned with high speed quantitative linear analysis of mineral specimens. Other procedures are listed in section (1.1.3).

The start-up/close-down operations followed by the high speed quantitative analysis procedures are described below.

a. Start-up/close-down operations

- The spectrometer cavities and the electron beam column are evacuated (low vacuum followed by high vacuum procedures).
- The mineral specimens are inserted.
- The electron beam is switched on and the lens

currents are adjusted to focus the beam onto the specimen surface.

b. High speed quantitative analysis

The purpose is to deduce 3-dimensional information about the mineral specimens from line traverses. The various operations are described below.

- Setting the instrument to recognize a set of mineral compositions.
- Moving the specimen along a chosen pattern (fig.3.1)
- Recording the intercept length of each mineral particle (L1) and each mineral phase (each particle may be composed of several phases).
- Recording the interparticle distances (L2)
- Information must be produced for the analyst both during and after a traverse. This requires capacity for storage of the information and for analysing it, as well as being able to report instrument status information and specimen analysis information in some visual form.

3.2.3. Model of the control processes

The control processes are responsible for performing the required system operations by measuring, manipulating and controlling the process variables. The major target of the steady state control system is to achieve a maximum speed in analysing a specimen. Physical limitations imposed by the controlled system limit the speed of analysis as will be seen below.

The control of the start-up/close-down operation:

The start-up operation proceeds sequentially as follows:

- The water system is switched on for pump cooling.
- The magnetic valves on the pipes for the rotary pump are switched on.
- The rotary pump is switched on to evacuate the system.
- The vacuum gauge is measured constantly until the required vacuum is achieved.
- The magnetic valves on the pipes for a diffusion pump are switched on.
- The diffusion pump is allowed to heat for a certain period.
- The stage is driven to the position where the specimen can be inserted.
- The specimen is now installed in its chamber.
- The specimen chamber is evacuated.
- The filament warming current and the HT (electron accelerator) switched on.

The close-down operation, either at the end of operations or in case of emergency, is the exact reverse of the start-up operation.

The control of the high speed quantitative analysis:

- The selection of an appropriate analysing crystal takes place whenever a monochromatic wavelength corresponding to a required mineral phase is needed.
- The collimating slit is inserted in front of the X-ray

detectors, 'open' or 'shut' depending on maximum count rates. If the maximum count rate swamps the detector, i.e. signals overlap, then the slit must be inserted to reduce what is known as the 'dead-time' effect.

- The angular re-positioning of the spectrometer units follows a crystal change; the detectors are mechanically moved so as to fall in focus and on the Rowland circle, in relation to the crystal and the specimen.
- For particular specimens it is necessary to speed up the operation so that the matrix (non-phase) regions of the specimen can be detected rapidly by measuring the specimen current and/or the backscattered electron current. Measuring the backscattered electrons using two detectors simplifies the detection of cracks and topographical features. Since these measurements can be carried out almost instantaneously, the specimen is driven at the maximum speed when the beam is on matrix (which may constitute up to 75% of the area of the specimen in many cases).
- The controller drives the specimen stage along the operational path (Fig.3.1) in one micro-meter increments using three stepper motors (for x, y and z directions).

When the beam is stationary over a point, the X-ray counts reading on each spectrometer (up to 5 spectrometers) are measured. The combination and the values of the X-ray counts indicate various mineral compositions or phases. The length of each phase (which could vary from several micro-meters to a few hundred micro-meters) has to be recorded together with the type of phase.

The speed of the stepper motors in this case is limited since a minimum period is required to accumulate sufficient X-ray counts to discriminate between the mineral phases. The braking operation (switching from high speed to slow speed) may be accompanied by overshooting the phase starting point and will require readjustment.

Meanwhile, other operations have to be performed by another controller if possible, such as:

- Measuring the beam current and readjusting the filament voltage to keep it constant.
- Checking the various components related to the safety of the system such as the vacuum gauge, the water flow, the valves, etc.
- Preparing statistical information from the collected particle (or phase) information.
- Sending the particle (or phase) information to be stored in a processor with powerful computing capability which would be able to calculate 3-dimensional information from the one-dimensional particle information.
- Monitoring information must also be sent regularly or on request describing the current status of the instrument.

The various control processes are shown in fig. 3.2. Each row of processes represents a group of operations which might have to occur simultaneously. The vertical sequence, however, does not represent a sequential flow of events.

3.3. Traffic Patterns and Communications Requirements

The concurrent operation of some of the control processes, as shown in fig. 3.2, is best handled by concurrent control (or processing) to attain the highest possible speed of operation.

3.3.1. Traffic rates

There are several groups of control processes because of the necessity to handle high data rates.

Group 1 (Processes (5) and (6)). These processes are responsible for the control of the x, y and z stepper motors, the measurement of the x, y and z shaft (or position) encoders, the measurement of up to five spectrometers (x-ray counts) and the identification of phases and the matrix by comparing the measured results with pre-set tables. These 11 variables are measured and controlled at 10 m.sec intervals.

Experiments on the Geoscan with a processor of 4 μsec average instruction time has shown that 10 m.seconds could not accommodate extra control or computational tasks without exerting delays. The useful information generated by this group of processes is produced at a much slower rate. Experiments have shown that information (whenever the matrix begins) regarding changes of phase is produced on the average every 300 m.sec. This period, however, is the total period spent by both groups 1 and 2 before identifying a new phase or matrix. The actual period spent by the processes in group 1 correspond to the average

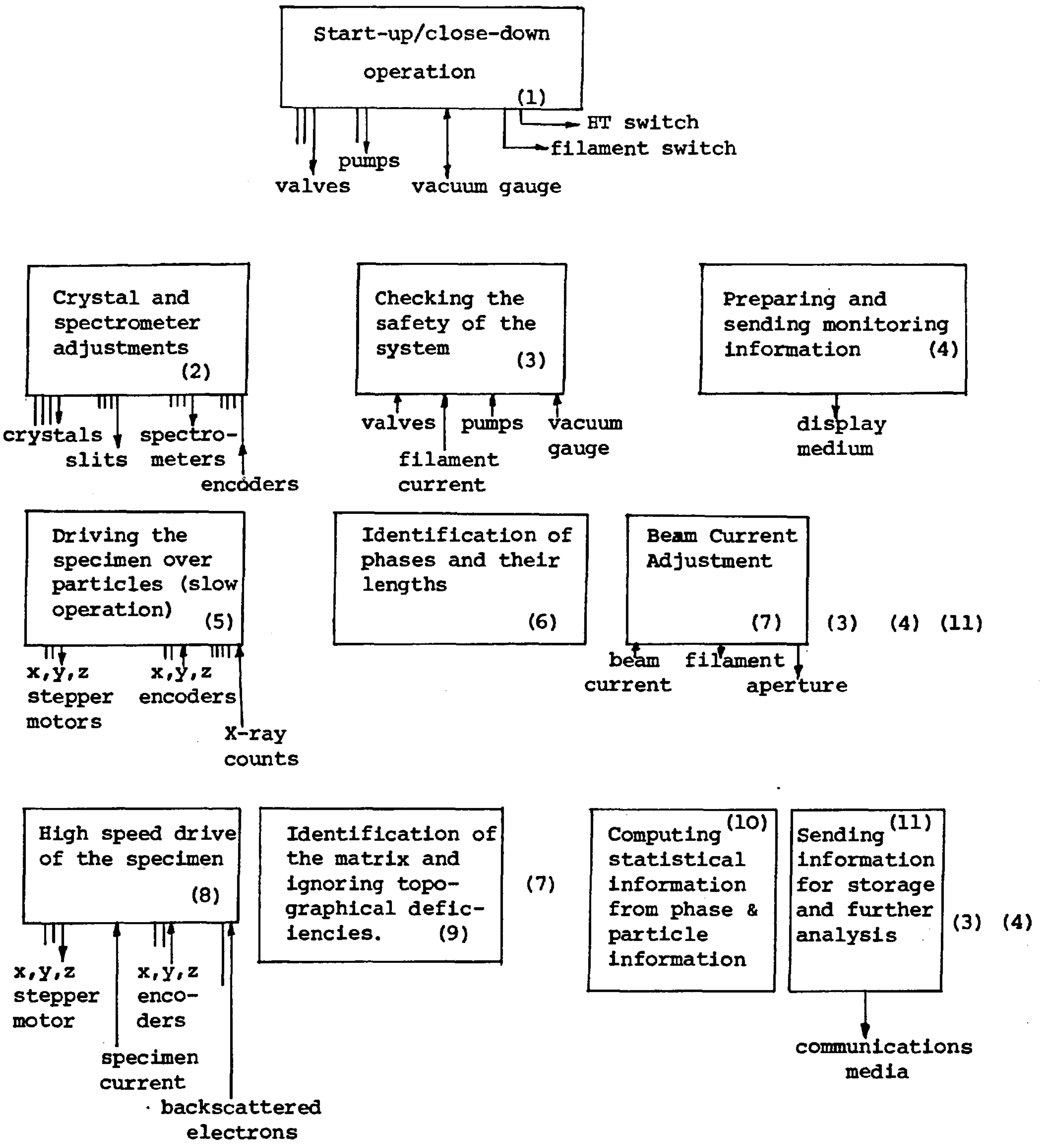


Fig. 3.2. The control processes controlling the start-up/close-down operations and the high speed quantitative analysis operation.

particle length and have been seen to be 250 m.sec. on average (see L1 in fig. 3.1).

The above results correspond to specimens with no phase transformations within the particles. Some specimens contain particles with rapid phase transformations as shown in fig.3.3. Phase information in this case is generated at average intervals of 40 m.sec.

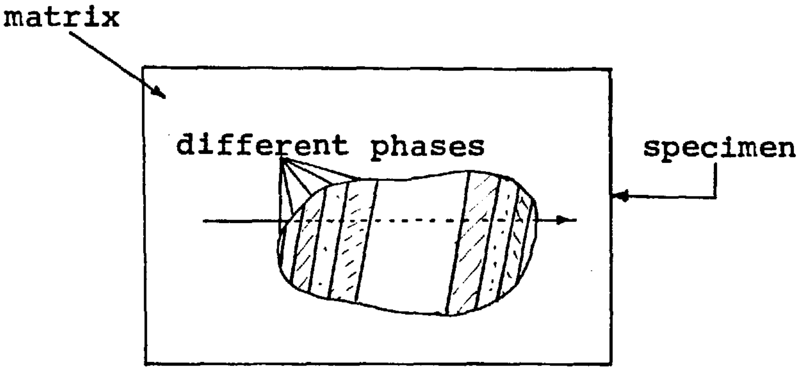


Fig.3.3. A particle with rapid phase transformations.

<u>Group 2:</u> (Processes 8 and 9).

These processes are responsible for the control of the x, y and z stepper motors, the measurement of the x, y and z position encoders, the measurement of the specimen current and two measurements of the backscattered electrons, in addition to analysis of the measured variables in order to identify the matrix and the start of a new phase. This operation occurs when the beam is incident on the specimen matrix and the stepper motor in the travelling direction is operated at its full speed (~ 1500 micro meters per second in most commerical systems) i.e. at 0.66 m.sec. intervals or steps.

Experimental results have shown that for the type of quantitative analysis performed the average interparticle distances to be 75 micrometers (see L2 in Fig. 3.1.) Phase information (when a new particle is met) is also produced on the average at 300 m.sec. intervals. This period again is the total period spent by groups 1 and 2 before identfying a new particle (L1 + L2). The actual period spent by group 2 over the matrix identification averages to 50 m.sec.

Although the operations of groups 1 and 2 are interleaved, yet information belonging to each group must be processed during the undedicated period e.g. process (10) handles phase information resident in the controller of group 1 and could be processed during group 2's operation.

The other processes concurrent with groups 1 and 2 occur at a less frequent rate and thus can be handled by the undedicated controller at any time.

These controllers are processors capable of control, measurement and computation. They have to allow the resident processes to communicate in order to convey information, e.g. information exchanged between groups 1 and 2 to switch from one speed and identification process to another.

Other processes such as (11) may form a group of their own since the communications with a remote storage/high powered computational facility require an almost dedicated processor to handle the protocols which ensure error-free transmission and reception.

3.3.2. Process assignments and traffic patterns

Table 3.1 shows the assignment of processes to 4 processors which require a communications facility to exchange control and information in the form of messages.

TABLE 3.1.

The assignment of process to processors

Processor	Processes dedicated control period	 undedicated control period
Processor 1:	(8) (9)	(4) (9)
Processor 2:	(5) (6)	(4) (7) (10)
Processor 3:	(1) (2) (3) (4) (11)	
Processor 4:	is dedicated to monitoring the processes and performing local data analysis by the user of the instrument. It also handles a variety of peripherals for print-outs, graphic displays and storage media.	

The flow of information between processors is in the form of messages. Five basic categories of message are distinguished:

1. Control messages:

    a. Messages exchanged between processors 1 and 2 to increase or decrease the speed of specimen movement.

  b. Messages from processor 1 to processor 2 for coordinating phase identification and cancelling topographical effects.

2. Information messages:
  a. Messages carrying phase information for remote filing.
  b. Messages carrying instruments status information.
3. User/operator request messages.
4. Filing, line printer and display information messages.
5. Alarm messages: in case of component failure or total system failure.

The direction of flow of these messages between the 4 processors is shown in figure 3.4.

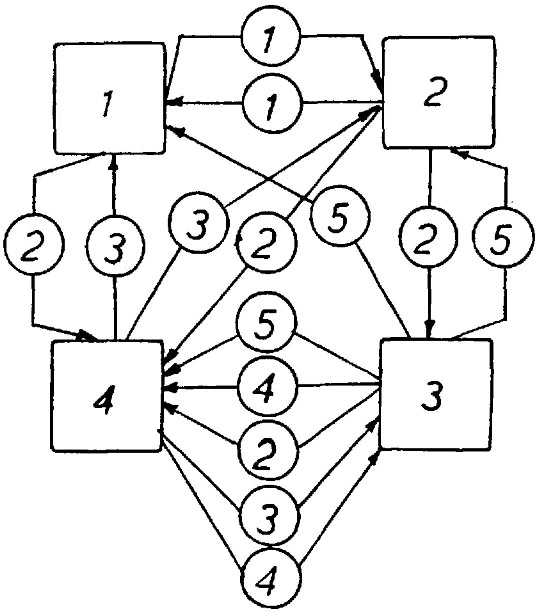


Fig.3.4. Types and directions of messages flowing between different processors.

The rate of flow of each type of message depends on several factors such as the specimen, the rate of operators requests, the sizes of information files to be transmitted, the speed capabilities of the peripherals, etc. Typical examples of flow rates and message formats will be shown in chapter 7.

The major objective now is to seek a form for communications between the four processors. This network of processors (computer network) should be capable of handling the high speed required for exchanging messages with minimum delays. This requirement must fall within the economic structure of the overall system.

The objectives considered when designing a computer network to control an X-ray micro-analyser are discussed in chapter 4. These could be briefed in the following:

1. Economy.
2. Capability of incremental growth to accommodate more processes and related instruments.
3. Reliability.
4. Stability.
5. High performance in terms of overall message delays.

The following chapter introduces the major classifications of computer networks from the point of view of localized control systems.

CHAPTER 4

Review and Evaluation of Broadcast Networks for Control

4.1. Introduction to computer networks for control

If several computers are to cooperate in sharing a set of tasks, then a communications system must be provided. The interconnection method determines the topology of the system. We recognize three fundamental topologically different approaches, as seen in fig. 4.1; these are the store and forward mesh (including the star configurations), the common bus or highway, and the loop configurations. The mode of operation determines whether the network is a 'broadcast' type or a 'store and forward' type.

A mesh is a network of computers partially connected by a number of point-to-point links, messages from one computer to another being routed via a number of intermediate computers. At each intermediate computer, a message is read in, stored and sent on, or forwarded, to the next computer on the route. This technique is usually associated with the properties of high reliability, long distances (on a national scale or larger), and complex communications protocol and, hence, complex software. A star network is the simplest form of a store and forward mesh and utilises one central switch to route the information between all nodes.

In a highway, all stations are connected by a common bus, whose capacity is shared amongst all stations. Three techniques are used to share access to a highway; contention, polling and random access. Contention access is usually used over very short distances only. Compared with the mesh, highways are easier to implement on a small scale due to the lack of any routing problems, but suffer from problems of reliability and, for longer distances, of performance. Speed of operation is limited by the electrical loading of a highway, which depends both on the number of stations attached and on the length of the bus.

A loop may, in principle, be considered as a sequence of point-to-point links closed in on itself. However, it has very different properties from a mesh network, arising mostly from the mode of working employed in most loop networks known as check and forward. In this, messages are routed around the loop from point-to-point, en route to their destination, but instead of being completely read in and stored at each station before onwards transmission, are subject to a delay of only a few bit times. Another major difference between loops and meshes is broadcast operation. In broadcast operation, each message is placed on the communications medium and is presented to all stations, and it is the responsibility of each station to determine whether the message is intended for itself or not. In this respect, loops are similar to highways which also use broadcast operation.

This broadcast operation of loops and highways has a number of advantages over the store and forward operation of mesh networks.

(a) Because messages (or packets) are not completely read in and stored at intermediate stations before being forwarded, buffer storage is not required, thus reducing the overall system cost. A partial exception to this arises in the delay buffer insertion technique (see Section 2.5).

(b) Routing algorithms are either absent or very simple, hence fewer resources are used and software is simpler.

(c) Delays are small as no store and forward operation is needed.

(d) Multiple destination addressing may be implemented easily and efficiently.

To select a most suitable configuration to suit a particular application we have devised a set of criteria upon which the applicability of a certain network configuration is judged.

These criteria are intended for localised control applications. Due to the cost effectiveness of 'broadcast' configurations, the set of criteria is also suitable for this type of networks.

In the following sections a description of the criteria is given and it will be seen that loop networks are more appropriate than common bus networks for our particular control application. However a short review of 'common bus' techniques is given, followed with more emphasis by a review and evaluation of loop networks.

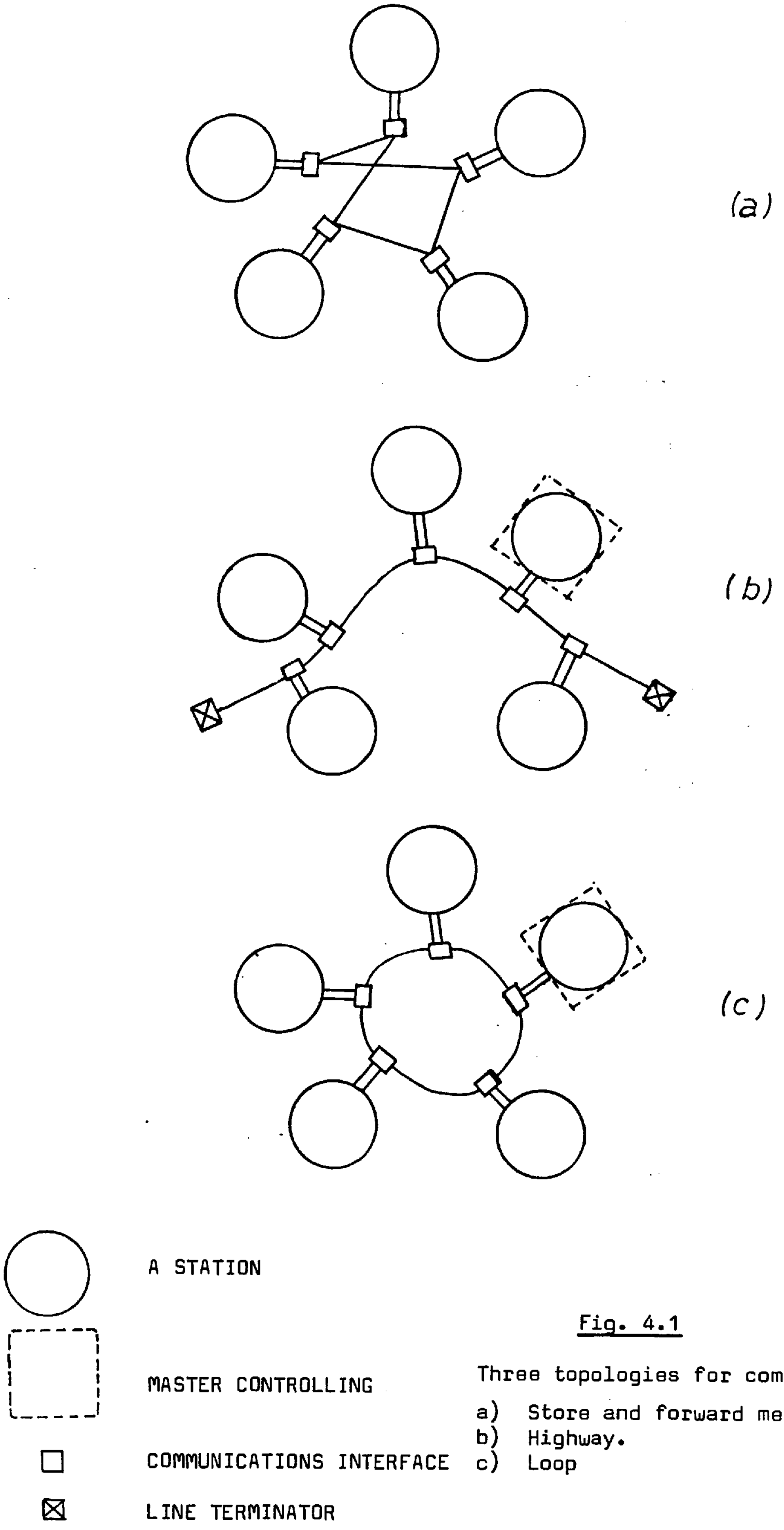


Fig. 4.1

Three topologies for communications.

a) Store and forward mesh.
b) Highway.
c) Loop

4.2. Criteria for Evaluation of Broadcast Networks

The set of criteria listed and explained below will help in evaluating and analysing broadcast networks.

1. Configuration and topology.
2. Sharing the network capacity.
3. Timing, synchronization and data transmission.
4. Distribution of control.
5. Reliability.
6. Performance.

The criteria attempt to categorize the properties of computer networks. Each criterion contributes a factor to the global cost. The cost depends on the users applications.

The following subsections gives a brief description (by no means exhaustive), of each of the abovementioned criteria.

4.2.1. Configuration and Topology

The configuration and topology of a network defines the layout and the method of interconnection of nodes.

The configuration and topology of any individual network characterizes the performance, cost, limitations and reliability.

Our concern in this section is to present the basic topologies which give broadcast features. These could be summarised as:

1. Common Bus Networks: where a common medium is used for nodes to exchange messages (e.g. radio or cables).
2. Loop Networks: The basic loop is unidirectional with nodes connected in tandom via loop interfaces.
3. Floodrouting Networks: Such networks are basically 'store and forward' systems with each node being able to forward each message to all other nodes connected to it. Unlike the bus and the loop configurations, this will store whole messages before rerouting them to other stations.

Limitations on the size of a network are affected by many factors such as the physical size of the site, addressing capabilities, electrical constraints, performance and reliability considerations. Many of the constraints could be overcome by various techniques, but at the expense of cost and performance. Common bus networks exhibit all the above constraints, yet in refs. (4.2), (4.34) methods are given to avoid these constraints in cases where expansions are desired. Ref. (4.1) reviews the constraints of loop networks and methods of expansion.

4.2.2. Sharing the Network's Capacity

This is a major issue in networks where high speed and low response times are required. The techniques are normally implemented as the low level protocol, in each station in software algorithms and sometimes partly in hardware. The three main types of network sharing (4.1) are given below:

1. Polling: here a 'master controlling station' polls all other nodes in a given order to grant any requests. This method, despite its simplicity, presents time wastage and the extra cost of a dedicated master controller.

2. Time Division Multiplexing: In this case each station in turn is assigned a time slot during which it is allowed to transmit messages. Although this method shows an apparent improvement in delay performance over the polling technique, it still gives very unsatisfactory performance at low loads especially if the traffic is unequally spread between nodes; in addition to technical complexities for timing and synchronization.

3. Random Contention: Stations requiring transmission contend for the transmission medium. Many techniques have been devised to resolve simultaneous contention (Examples Ethernet (4.2), ALOHA (4.3), DLCN (4.4)).

Many other techniques involve combinations of the three described above, such as the control passing in loop networks (Newhall loop, (4.5)), slotted loop (pierce loop, (4.6)) and slotted bus (slotted Aloha, (4.11)).

4.2.3. Timing and Synchronization

Bit Synchronisation:

For loops operating in check and forward modes (the normal case), or for the single bus operating mode, all transmitters and receivers have to work in synchronisation at the bit level. To achieve this, receivers will have to take their timing either from a central clock via separate lines, or derive timing from the line signals. A practical example of the former method is found in Serial CAMAC (4.8) in which a clock signal is distributed from a network supervisor (a loop network) by means of a separate line in parallel with the main data loop.

Deriving bit synchronization information from the transmission line requires the receiver to observe signal level changes on the transmission line. If the bit coding scheme entails a signal change during each bit time frame, then the network interface could derive its timing from this without, in principle, using a local clock. In practice to avoid jitter, a local clock might be used to give a 'flywheel' effect, but would not need to be very accurate (4.9). A detailed description of various types of clock derivation at bit level is found in ref. (4.1).

In loop and bus networks, there is generally a single master clock, while each other node contains a slave clock which is synchronized to the master clock.

Message Synchronization:

Message synchronization means that a network interface must be able to recognize the start and end of a message (or packet), amongst a continuous stream of bits. This is normally achieved by preceding a packet with a unique bit sequence or control signal which does not confuse with the data. In cases where bit sequences are used to identify the message frames bit insertion (or bit stuffing) techniques are used.

In both cases of bit and message synchronization, the techniques adopted depends on the transmission level protocol. The economics of these methods vary widely according to what is recognized as standard. For example the bit synchronization in the 'Ethernet' uses phase-encoded signals (4.2) which require purpose-built hardware. Most other loop networks implement standard synchronous (or asynchronous) communication interfaces.

4.2.4. Distribution of Control:

In this section we distinguish three levels of control.

a. Timing and synchronization in the network at bit level and generation of empty packets (slots) in time slotted networks.
b. Allocation of bandwidth (or channel allocation) to individual nodes.
c. Higher management functions such as initialization of the network, control of certain types of error (such as lost messages) and reconfiguration in case of node or link failure.

For each of these levels, control may be exercised from a single node (centralized control), it may be spread over all nodes and not be dependent on any individual one (distributed control), or it may be some intermediate arrangement. For maximum reliability, it is preferable to distribute control to avoid reliance on any single node. Also, centralized control of channel or bandwith allocation may lead to inefficient performance. On the other hand, ease of design and implementation usually implies centralized control of functions. Hence there is a trade-off between distributed and centralized control and this trade-off may be different at each level of control (4.1).

4.2.5. Reliability

In broadcast networks errors may arise from a number of sources. Information which is already on the communications channel may become corrupted, possibly due to the presence of electrical noise, either caused externally or arising from connectors, faulty transmitters, etc. Usually this corruption is intermittent and "bursty". Secondly, the links in the network may be damaged or cut and, in effect, isolate some of the nodes. Finally, one or more of the nodes may fail or malfunction in such a way that they cannot accept messages or relay on information (in case of loops) from and onto the network.

These error conditions may cause the following effects on the system:

(a) Loss of synchronization of the receivers and the transmitters in the network interfaces. This may necessitate a re-synchronization procedure.
(b) Corrupted messages on the communications channel.
(c) Incomplete messages on the communications channel.
(d) Faulty messages which seem to be correct.
(e) Isolation of some nodes due to failure in transmission line.

Control of errors and their effects may be exercised by a suitable protocol. This may be considered in four steps:

(1) The use of error detection techniques (e.g. parity, longitudinal and transverse, blocksums and cyclic redundancy checks) to recognize corruption or loss of messages, or failure of a network interface or the transmission line.

(2) The correction of transient errors; this involves correction of data by means of forward error correction or by retransmission, and removal of undelivered messages.

(3) Isolation of failing components, either segments of the network or network interfaces.

(4) Alternative routing, either through use of standby parallel links or multiple interconnections.

4.2.6. Performance

The performance measures of a broadcast network in control environments hinges greatly on message and packet delay. Other performance measures are buffer queue lengths and buffer overflow (or blocking) probability, channel utilization, and stability. These are defined below:

(a) Message Delay:

Delay may be defined as the time elapsing between the generation of a message at a host computer and the arrival of the message at the destination station.

In general this will have several components, thus:

- Queueing time: in the transmission buffer.
- Latency: This is the time taken between a message reaching the top of the queue and being actually transmitted.
- Transmission on service time: which is the division of message length by transmission speed.
- Propagation delay: This has two components.
  1. Line propagation delay, where at 1 μbit/s, each 300 metres contributes a maximum of one bit delay.
  2. Delay at each node (mainly in check and forward networks). This could range from one bit to one or more messages.

Where a message is split into shorter, fixed length, packets, there are two delays of possible interest. Delay per packet and delay per message.

This definition of delay may be extended to include the time taken to send an acknowledgement of receipt of message back to the sender. On some systems, this may be an integral part of the message transmission system. In others, it may be a separate message.

(b) Buffer Occupancy:

When messages are generated, they are placed in a buffer to await transmission. A measure of interest is the distribution of queue lengths for this buffer, as it determines how much buffer storage is required. If service is slow or message arrival rate is high, then a proportion of messages will be lost when the buffer overflows.

Alternatively, the producer process or terminal will be blocked.

(c) Channel Utilization:

This parameter represents the proportion of time that the communications channel is busy. As a network may be split into a number of point to point links, or segments, and as each segment may have different traffic levels, the network utilization must be summed for all segments. This is complicated by the fact that not all information is useful.

(d) Stability:

Some random access networks, where nodes contend for the transmission channel may attain a maximum throughput at a certain traffic load. After this, throughput will start to decline and message delays will increase indefinitely. This is due to the random contention for the channel which leads to a thrashing situation. In the case of single bus networks (e.g. ALOHA, slotted ALOHA, CSMA or Ethernet) (4.2), (4.10), instability problems may frequently occur and certain control procedures have to be implemented. Some networks avoid the regions of instability at the cost of using very high speed communication channels.

The criteria for network evaluation described above are set to help the designer in choosing the most appropriate network for his application. Priorities and orders of importance may differ from one application to another. However, for localized networks with small number of nodes used in control environments, reliability and high performance are essential. For the control environment at hand (i.e. the electron probe micro-analyzer) economy is also of great importance. This will lead to a greater emphasis on loop and bus type broadcast networks in an attempt to choose the most suitable network design.

4.3. A Review of Single Channel and Loop Networks

As networks capable of broadcasting messages, single channel networks and loop networks have several advantages over store and forward networks with floodrouting capability:

- Economy in buffer storage since messages (or packets) do not have to be stored completely before retransmission. Hence, also, shorter delays.
- Economy in the hardware: smaller number of transmitters and receivers (one of each type is needed in the case of loops and single channels) and cabling.

For these reasons, only single channel and loop networks were considered for the application. As will be seen in the rest of this chapter, a loop network was preferred for the application for economy, stability and performance.

The next two sections consider separately each type of networking in the form of a review. An extensive survey of loop networks has also been made (4.1).

4.3.1. Single Channel Networks

The topology of a single channel network necessitates the availability of a bidirectional medium for transmission.

This medium could be the surrounding atmosphere in case of satellite communication, or a single coaxial cable connecting all nodes. There are several means for sharing the capacity of the channel and resolving contention between the various nodes, either for a large-sized widely distributed network, e.g. when a satellite is used for packet switching, or, a large number of local stations connected by a cable or radio transmission.

Satellite packet switching networks are used for geographically dispersed nodes and are characterised by long propoagational delay in a round trip transmission. Methods of sharing the capacity of such a network vary in performance and in complexity, however they are not suited to localized networks for control. The list below gives a quick reference to the capacity sharing methods:

1. Pure ALOHA (4.3)
2. Slotted ALOHA (4.11)
3. Excess Capacity Method (4.12), (4.7)
4. Capture Effect (4.10)
5. Dynamic Reservation Method (4.14)

In local single channel networks, the transmission delays are negligible compared to the message length. Thus it is possible for all nodes to 'sense' or 'listen' to a message being transmitted on the channel almost instantaneously. A wide band radio channel or a passive transmission line (usually a low loss coaxial cable) could be used. Contention for sharing the channel is performed by sensing a carrier signal. The method is known as the carrier sense multiple access method (CSMA) (4.15). The carrier signal could be a carrier frequency in the case of radio transmission or bit encoded pulses in case of coaxial cables (4.3) (4.2). Both methods require specially built hardware for transmission, sensing and interpreting the carrier signals.

There are several mechanisms, all based on the CSMA, for resolving contention and sharing the channels capacity:

1. Non persistent CSMA (4.10): A node will transmit a message if the channel is sensed idle, otherwise it schedules the retransmission of the message to some later time. A slotted nonpersistant CSMA will force all terminals to synchronize in order to fit the fixed size packet into the time slots.

2. P-persistant CSMA: If the channel is sensed idle by a node, it transmits by a short duration (usually a round trip delay period) with a probability (1-p). If at this new point of time the channel is idle, the same process is repeated. Otherwise, it acts as if a conflict has occurred and reschedules the transmission of the packet. If the node senses a busy channel, it waits (or persists) until it becomes idle, then operates as above.

   A slotted version of a 1-persistant CSMA can also be implemented by slotting the time axis and synchronizing the transmission of packets. In this case, a node ready for transmission will always start the transmission of a packet (with probability 1) once the channel goes idle.

   As an alternative to the random access methods described above, time division multiplexing and polling techniques could be used. However these either have high economic overheads or low utilization.

For bit synchronization, some form of phase encoding is used, where the cable is utilized in the base band region of its frequency spectrum (4.1). However, due to random contention and the operational technique, messages may have to be aborted and synchronization is lost. In Ethernet (4.2), each node senses the 'carrier' signal on the channel, and those requesting transmission operate in a similar manner to the 1-persistent CSMA mode.

Thus if more than one node transmits, the collision of packets is certain. Unlike ALOHA or CSMA modes, the nodes could immediately detect a collision and abandon it (instead of running to completion), then retransmit the packet after some dynamically chosen random time period. To allow messages to be received according to this mechanism the communications interfaces to the communications channel must be equipped with five mechanisms:

1. Carrier detection: which serves for all CSMA modes and allows the nodes to wait until a current transmission terminates.
2. Interference detection: allows an early detection of colliding messages.
3. Packet error detection: a receiving node could discard

packets which do not satisfy error checks.

4. Truncated packet filtering: short truncated packets caused by interference detection are filtered out in the hardware.
5. Collision concensus enforcement: when a station determines that its transmission is experiencing interference, it momentarily jams the channel to ensure that all other participants in the collision will not detect intereference.

It is clear from the above discussion, that considerable hardware complexity is required to implement such a system.

Single channel networks operating at high speeds (such as Ethernet) are designed so that messages would be received at their destinations only with high probability (4.10). It is the responsibility of processes in the source and destination nodes to take the precautions necessary to ensure reliable communication of the quality desired. Other reliability problems such as link failure and node failure could be dealt with by cable duplication and by node isolation techniques (4.2, 4.16).

The performance of CSMA modes compared with that of ALOHA modes appears to be superior in terms of delay and throughput (4.15), (4.10). However, we shall not proceed further with the details of performance characteristics of single channel networks since the really serious problem appears to be in stability.

Random access techniques of single channel networks suffer from instability problems when traffic loading exceeds a certain maximum, (4.13), (4.10). For stable operation, the throughput has to be controlled. Yet if it slips to bistable operating points, optimal control procedures are necessary to regain stability (4.2), (4.35), (4.36). If the operation shifts into unstable regions, the system may come to a halt and reinitialization is necessary (4.10).

4.3.2. Loop Networks

Most loop networks take the form of a unidirectional loop with nodes connected in tandom via loop interfaces, as shown in fig. 4.1.

In general, the loop interfaces are tightly coupled and in close physical proximity to the hosts to which they are attached (4.1).

Almost all of the loop systems examined actively regenerate the signals at each loop interface, and hence electrical considerations do not limit the number of nodes in the loop. They may, in principle, affect the maximum distance between nodes, but in the local environments commonly encountered, this would not normally be a serious problem. By comparison, a single channel network relies upon a single transmission path over the whole path length of the network, and baseband signals will suffer more attenuation and distortion than in the shorter individual point to point links of a loop. A single channel highway may overcome the length restriction by use of modems and repeaters, but this slows the operation and increases costs (4.2).

Other topologies such as multiple loops are used for expansion, (4.5), (4.6), (4.18).

In a loop system, all traffic must share a single physical channel. There are four basic approaches for multiplexing the channel capacity amongst the participating nodes on the loop. There are more classifications found in the literature (4.18); however, they are combinations based on the four classifications described below:

1. Static Sharing

In static sharing, or ordinary time division multiplexing, the time axis is divided into equal size slots, and each node is assigned a fixed number of slots (not necessarily contiguously). Figure 4.2a shows an example of time and space diagrams of its operation; 4.2b shows a snapshot of packets in the system at time 2t. A packet from node 1 is already on the loop, and packets from nodes 2 and 3 are waiting to enter. The IBM 2790 system (4.19) utilizes this method and is used to connect peripheral devices to a small computer.

2. Demand Sharing with Fixed Length Slots

The time axis is again divided into fixed length time slots. A demanding node will contend for an empty slot, and will only start transmission when the slot indicates that it is free. Figure 4.3a shows time and space diagrams for a system with four nodes, and figure 4.3b shows a snapshot of the loop at time 2t.

Static Sharing scheme (STDM) for a loop with four stations.

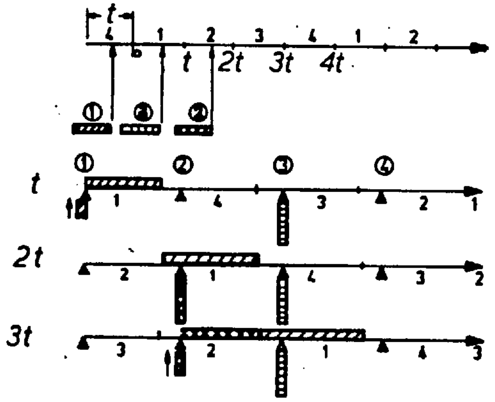


fig 4.2(a) packets from stations 1, 3, and 2 at various times either on the loop or waiting to enter.

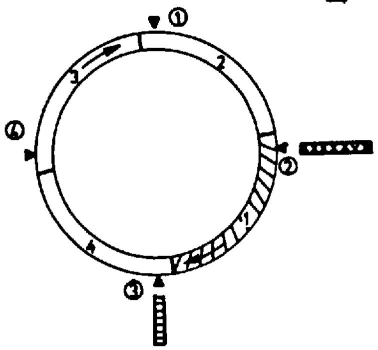


fig 4.2(b) packets on the loop at time 2t.

Demand sharing with fixed length slots for a loop with four stations.

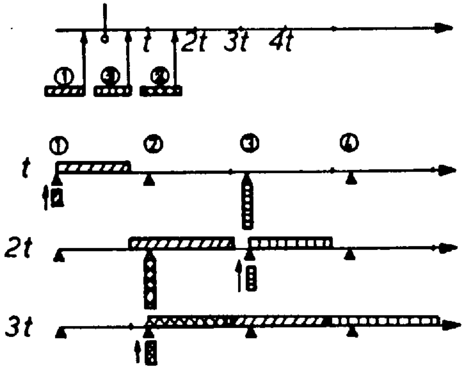


fig 4.3(a) packets from stations 1, 3 and 2 at various times either on the loop or waiting to enter.

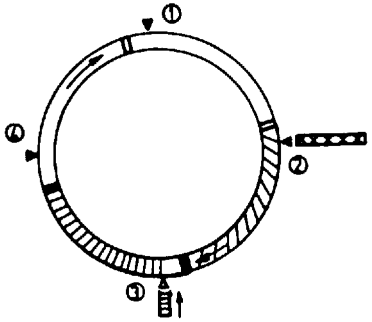


fig 4.3(b) the state of the loop at time 2t.

Demand sharing with variable length messages
(control token method of Farmer and Newhall)

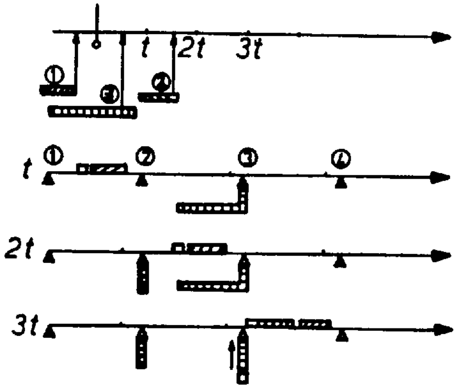


fig 4.4(a) message from stations 1, 3 and 2 at various times either on the loop or waiting to enter.

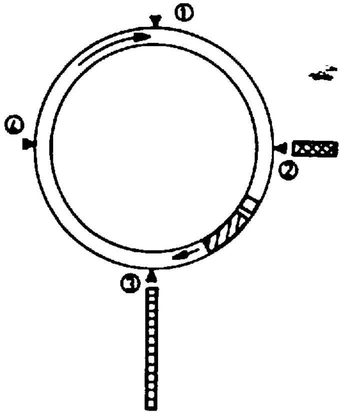


fig 4.4(b) the state of the loop at time 2t, with a message from station 1, followed by the token, actually on the loop.

Demand sharing with variable length messages
(shift register insertion technique of Lui)

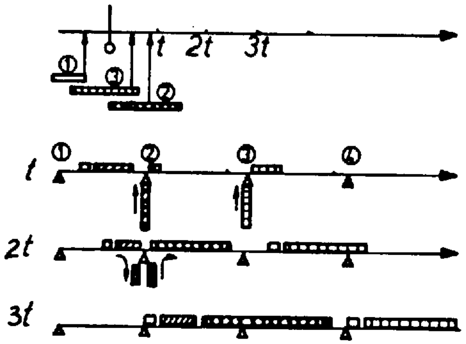


fig 4.5(a) messages from stations 1, 3 and 2 at various times either on the loop or waiting to enter. One message is shown being diverted into the delay buffer at station number 2.

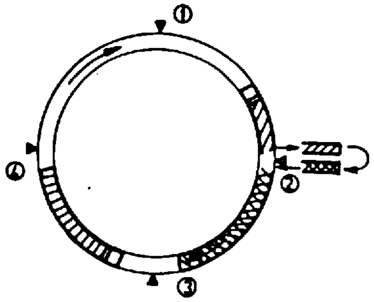


fig 4.5(b) the state of the loop at time 2t.

The 'Pierce and Kropfl' loop (4.6) and the 'Spider' systems (4.20) operate on a similar principle.

3. Demand Sharing with Variable Length Messages:

A loop interface, once it has gained access to the loop, sends a variable length message. The message must indicate to the other loop interfaces when it terminates (for example, by a length field or special end of message character). Two methods allow the implementation of this technique, the control passing and the register insertion methods.

- In the control passing method, a single control token is passed from node to node around the loop. Only when a node is in possession of the token is it allowed to transmit its own messages (Farmer and Newhall loop (4.5)is based on this technique). (See figure 4.4).
- The register insertion technique implements a shift register in each loop interface which can act as a delay buffer for at least one message. If a node wishes to transmit a message, it must first wait until the message, if any, which it is currently relaying comes to an end. Then, even if more messages are received on its input side, it inserts its own message or messages onto the loop. Meanwhile any message arriving on the loop input side is routed through a shift register. When the new message being inserted by the node ends, the shift register may be progressively unloaded until either it is empty or the node again wishes to insert a new message. Figure 4.5. shows the operation of this method with four nodes.

The Distributed Loop Computer Network (DLCN) was the first to implement this technique (4.21), (4.22). Another method which utilizes buffer insertion but with a different assignment rule is introduced in chapters 5 and 6. In this method, each node has to decide dynamically whether to assign the communications channel to its own traffic or to the relayed traffic. The decision is made dynamically after measuring sample averages of message durations and produces a minimal overall average message delay.

4. Demand Sharing by Polling:

In the classical sense, polling involves a master station interrogating all slave stations, in some order, by means of poll messages. A poll is a short message which asks a specific slave station whether it has any data to send. The Weller loop (4.23) and the Serial CAMAC system (4.8) may be interpreted as having polled operation.

The polling technique is most suited for the case where all data traffic is between a master station and a number of unintelligent slave stations. Where communication is to be between two slave stations this must be as a sequence of master slave communications, and is very inefficient. Performance is very poor compared to the other demand sharing techniques due to the overheads of the poll messages.

Reference (4.1) gives a detailed discussion on bit and message synchronization in loop networks. It is worth noting, however, that standard communications techniques such as synchronous transmission (e.g. Tl carrier (4.24)) can be used. The DLCN for example uses a USRT chip, the COM-5025, to perform bit and message synchronization (4.22).

As discussed in section (4.2.5), it is preferable from the reliability point of view to use distributed control to avoid reliance on any single station. The survey given in reference (4.1) shows no adherence to this rule in many loop implementations. The channel or bandwidth allocation as a function of the overall loop control system if implemented in a distributed manner, cuts the overheads (in terms of time) and leads to a better performance (4.1).

In general the distribution of control in a loop system has many advantages such as:

1. Ability of incremental growth.
2. No initial overhead cost for the control station.
3. Higher reliability and no reliance on one control station.

The reliability of loop networks has been intensively studied. Methods and protocols for error detection and deleting stray messages have been pursued by many authors (4.8), (4.19), (4.25). A survey of these methods as well as techniques for detecting link and node failures are extensively explained in ref. (4.1). Methods for error recovery and system reconfiguration are also provided.

Many performance studies have been undertaken on individual types of loop networks. The parameters and assumptions used in the different published analyses vary but some authors have undertaken the task of unifying the assumptions to obtain comparative results (4.26), (4.28), (4.37).

The principal input parameters used in mathematical modelling are:

- Buffer sizes in nodes (usually single or infinite)
- Message arrival discipline (e.g. Poisson, exponential, fixed).
- Message length distribution.
- Error rates (usually ignored).
- Treatment of acknowledgement messages (usually ignored).
- Number of terminals in the loop.
- Packet sizes.

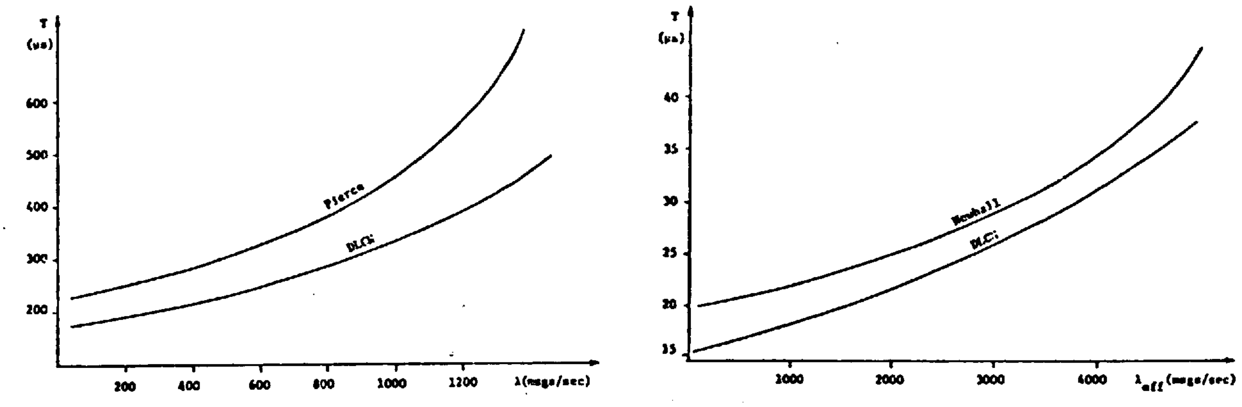
Performance measures of greatest interest in control environments are message and packet delay. Other measures which are important in aiding the design of the system are the buffer queue lengths and buffer overflow probabilities and the loop channel utilization.

Richardson (4.26) compared three loops for an enquiry-response type environment. The mechanisms that he considers are the control passing technique, a multiple slot technique (called SD, or slot deletion), and a variant called slot-no-deletion (SND). The difference between SD and SND is that in the former, packets that are marked empty by a terminal may be re-used as they continue along the loop whereas in the latter they may not be re-used but must continue back to the sending, or control computer, in a manner similar to that of the Cambridge loop (4.27).

Liu, Pordo and Babic (4.28) compared the delay buffer insertion technique, implemented by DLCN(4.29) with the slotted loop studied by Hayes and Sherman (4.30) and the control passing method studied by Kaye (4.31). Figure 4.6 shows the average message delay in the three networks as a function of offered load, and it is clear that the DLCN technique is superior for all loads.

Hopper (4.37) has also compared the performances of the preallocated (static sharing) slotted, control passing and delay buffer insertion techniques. His mathematical models dealt with both the cases of an infinite buffer and a single buffer. He also dealt with two methods of buffer insertion where subsequent transmission is considered: (a) when the next packet can be loaded and transmitted instantaneously, and (b) when this takes a finite number of bits delay. His results has shown that buffer insertion with instant replacement always has the least delay, while the normal insertion technique and the control passing methods are comparable and are still superior to the slotted and the preallocated systems, see figure 4.7. The line utilization of the buffer insertion and token techniques again seems to be the highest and the best, see fig. 4.8.

It is obvious from the mathematical models considered above that the delay buffer insertion technique leads to smallest delays specially with increasing loads. This appears to be due to the immediate access to the loop by new messages causing reductions in queueing time and latency.



(a)

(b)

Fig. 4.6
a)Pierce(slotted).vs.DLCN (Buffer insertion)
b)Newhall(control passing).vs.DLCN.

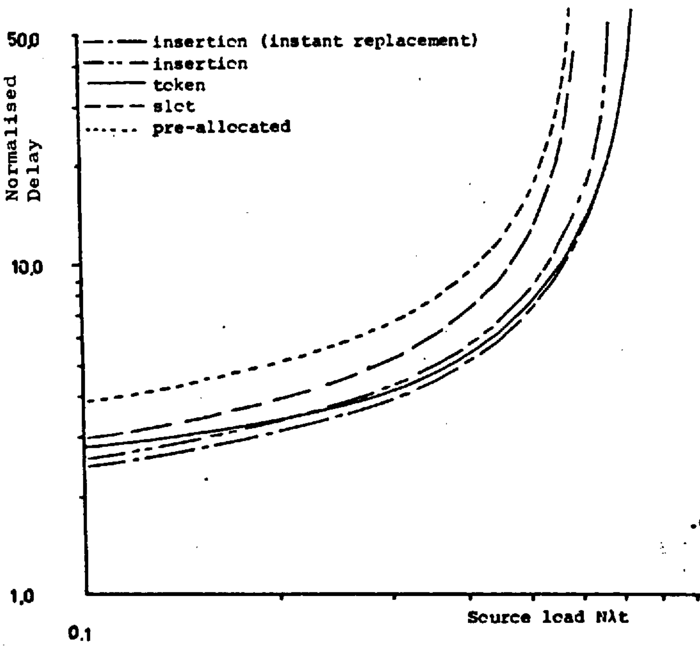


Fig. 4.7

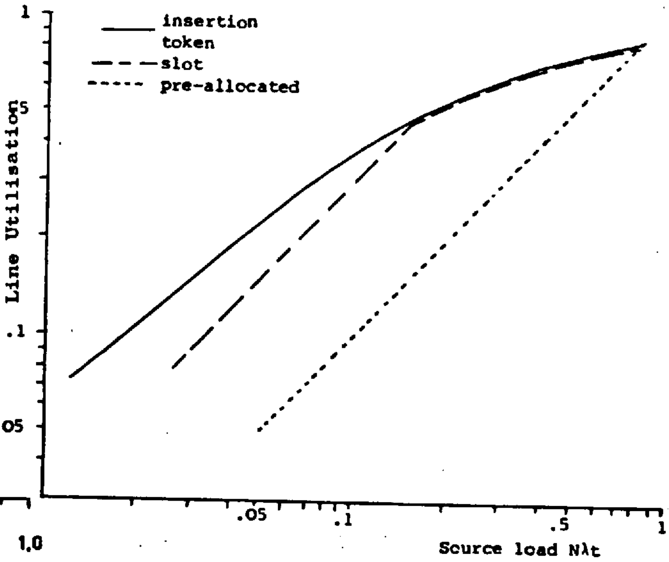


Fig. 4.8

It is also worthwhile noting that systems which allow variable length messages are more efficient as otherwise messages that do not precisely fill an integral number of packets lead to partially empty packets being transmitted. Splitting messages into several packets also means extra delays due to the multiple occurrence of latency times and the additional overheads of adding a header in each packet.

4.4 Comparisons and Conclusions:

In a local environment, loop networks, single channel networks and star networks (as a straightforward case of the store and forward mesh) may be compared. The main features for comparison are cost, reliability and performance.

Cost may be characterised by the number of receivers and transmitters (input ports and output ports), by cable lengths (more generally, the lengths of the communications channels) and by special master stations for control. Both the loop and the single channel may have only one receiver and transmitter per station, whereas a star configuration needs two per station (one pair in the station and the other in a central switch). It is self evident that cable length will be much larger for a star than for a loop or a single channel, particularly if the stations are a long way from the central switch. In a star, a high performance central switch is required, and this represents an additional cost penalty, particularly for smaller systems. Many loop and single channel networks do not need an expensive control switch, hence have lower start-up costs.

Loops may suffer from a vulnerability to system failure if a station or link fails (this could be overcome by special reconfiguration procedures (4.1)). However, a star is totally dependent on the central switch, which makes the system of higher vulnerability in the case of its failure.

Kaye and Richardson (4.32) have analysed and compared the performance of a control passing type loop with a single channel and a star network with polling. They have shown the loop to have a superior performance (in terms of response time) over the polled star and single channel. Hayes and Sherman (4.33) compared a loop (slotted) with a polled bus and a random access bus of the ALOHA type (1). Their analysis found that the message delay performance of the bus polling was ultimately limited by modem synchronization and, for similar traffic levels, was significantly worse than for the slotted loop. They also found that random access performance was worse than for the slotted loop particularly at higher traffic levels.

The apparent simplicity and economy of loop networks together with their satisfactory reliability, has lead our research into methods for enhancing the loop performance.

References

4.1. B. K. Penney and A. A. Baghdadi. "Survey of Computer Communication Loop Networks", Research Report 78/42 CCD, Imperial College, 1978.

4.2. Metcalfe R.M. and Boggs D.R. "Ethernet: distributed packet switching for local computer networks". Comm. ACM 19, 7 (July 1976), 395-404.

4.3 Abramson, N. "The ALOHA system - another alternative for computer communications". AFIPS Conf. Proc. 37 (Nov. 1970), 281.

4.4. Reames, C.C. and Liu, M.T. "A loop network for simultaneous transmissions of variable length messages". Proc. 2nd Annual Symposium on Computer Architecture, Houston, Texas (Jan. 1975), 7-12.

4.5. Farmer W.D. and Newhall E. E. "An experimental distributed switching system to handle bursty computer traffic". Proc. ACM Symposium on Problems in the Optimization of Data Comms. Systems (Oct. 1969), 1-33.

4.6. Pierce, J.R. "Network for block switching of data". Bell System Tech. Journal 51, 6 (July-Aug.1972), 1133-1145.

4.7. Abramson, N., "The throughput of packet broadcasting channels", IEEE Transactions on Communications, vol. COM 25, No.1 (January 1977).

4.8. IEEE Standard serial highway interface system (CAMAC). IEEE Standard 595- 1976; ESONE Standard EUR 6100.

4.9. Hippert R.O. "A pulse-code modulated transmission loop". IEEE 6th Int. Conf. on Comms., (June 1970), 36/10-36/15.

4.10 Kleinrock, L. "Queueing Systems: Vol.II". John Wiley & Sons, 1976.

4.11 Kleinrock L. and Lam S., "Packet switching in a satellite channel". Nat. Comput. Conf. Proc. Vol.42, Montvale, N.J.: AFIPS Press. 1973, pp. 703-710.

4.12. Retberg R. "Random ALOHA with slots-excess capacity", ARPANET Satellite Syst. Note 18, NIC Document 11865 - Stanford Res. Inst., Menla Park, CA (October 11, 1972).

4.13 Kleinrock L. and Lam S.S. "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation", IEEE Transactions on Comm. Vol. COM 23, No.4, April 1975.

4.14 Roberts L.G., "Dynamic Allocation of Satellite Capacity through Packet Reservation" AFIPS Conference Proceedings, 1973 National Computer Conference, 42, 711-716.

4.15 Kleinrock L. and Tobagi F.A. "Packet Switching in Radio Channels: Part 1 - Carrier Sense Multiple-Access Modes and their throughput-delay characteristics", IEEE Trans. on Comm. Vol. COM 23, No.12, Dec. 1975.

4.16 West A.R. "A Broadcast Packet-Switched Computer Communications Network" Design Progress Report, Queen Mary College, University of London, London.

4.17 Graham R. L. and Pollack H. O. "On the addressing problems for loop switching". Bell System Tech.Journal 50, 8 (Oct.1971), 2495-2519.

4.18 Heger, D. "Computer Control System with distributed microcomputers". Presented to 5th meeting of European Distributed Intelligence Study Group, Brussels, Belgium (Nov. 1977), unpublished.

4.19 Hippert, R.O. "IBM 2790 digital transmission loop". IBM J. Res. Develop., (Nov. 1970), 662-667.

4.20 Fraser A.G. "Spider - a data communications experiment". Computing Science Technical Report 23, Bell Laboratories, Murray Hill, New Jersey (1974).

4.21 Reams, C.C. and Liu, M.T. "A loop network for simultaneous transmission of variable length messages." Proc. 2nd Annual Symposium on Computer Architecture, Houston, Texas (Jan. 1975), 7-12.

4.22 Liu M.T. and Reams C.C. "The design of the Distributed Loop Computer Network". Proc. 1975 Int. Computer Symposium, Vol.1, Taipei, Taiwan, (Aug.1975), 273-282.

4.23 Weller D.R. "A loop Communication System for I/O to a small multi-user computer". Proc. 5th Annual IEEE Conf. on Hardware, Software, Firmware and Tradeoffs, Boston, Mass., (Sept. 1971), 49-50.

4.24 Fultz, K.E. and Penick, D.B. "The Tl carrier system", Bell System Tech. Journal 14, 7 (Sept. 1965), 1405.

4.25 Zafiropulo, P. "Reliability - a Key Element in Loop Systems". 1974 Int. Zurich Seminar on Digital Comms., Zurich, Switzerland (March 1974), D2(1) - D2(6).

4.26 Richardson, T.G. "The effect of protocol on the response time of loop structures for data communications". CRC Report 1261, Department of Communications, Ottawa, Canada (Aug. 1974).

4.27 Hopper A. "Data ring at computer laboratory", University of Cambridge. Conf. on Local Area Networking, NBS Spec. Pub. 500 - 31, Washington D.C (April 1978).

4.28 Liu, M.T., Pardo R., and Babic G. "A performance study of distributed control loop network". Proc.1977 Int. Conf. on Parallel Processing (Aug. 1977), 137-138.

4.29 Liu, M.T., Babic G. and Pardo R. "Traffic analysis of the Distributed Loop Computer Network (DLCN)." Proc. 1977 National Telecommunications Conf., Los Angeles, California (Dec. 1977), 31:5/1-31:5/7.

4.30 Hayes J.F. and Sherman D.N. "Traffic analysis of a ring switched data transmission system", Bell System Tech. Journal 50, 9 (Nov.1971), 2947-2978.

4.31 Kaye A.R. "Analysis of a distributed control loop for data transmission". Symposium on Computer Communications Networks and Teletraffic, Polytechnic Institute of Brooklyn (April 4-6, 1972), 47-58.

4.32 Kaye A.R. and Richardson T.G. "A performance criterion and traffic analysis for polling system". Infor. II, 2 (June 1973), 93-112.

4.33 Hayes J.F. and Sherman D.N. "A study of data multiplexing techniques and delay performance". Bell System Tech. Journal 51, 9 (Nov. 1972), 1983-2011.

4.34 Kohn R.E. "The Organization of computer resources into a packet radio network". Trans. IEEE on Comm., vol. COM-25, No.1, January 1977.

4.35 Lam S.S. "Packet Switching in a multiaccess broadcast channel with applications to satellite communication in a computer network." Computer Science Dept., School of Engineering and Applied Science, Eng. report UCLA-ENG-7429, March 1974.

4.36 Lam S.S. and Kleinrock L. "Dynamic control schemes for a packet switched multi-access broadcast channel". National Computer Conf., AFIPS Conf. proceedings, Vol.44, Montvale, N.J. 1975 pp. 143-153.

4.37 Hopper, A. "Local area computer communication network". Technical report No.7 - University of Cambridge, U.K. 1978.

CHAPTER 5

A Loop Computer Communication Network with Optimum Priority Scheduling

In this chapter a delay buffer insertion type unidirectional loop with distributed control i.e. no central control station, is analysed to find the method of loop access which minimizes the average message delays.

The computer communication network consists of N nodes linked by a communication channel in the form of a loop, see fig.(5.1):

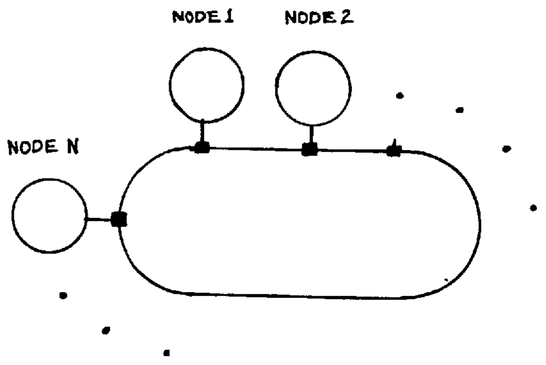


Figure 5.1.

The analysis carried out shows the way nodes on the loop should behave in order to minimise message delays. Expressions for average message delays are also derived. The analysis also assumes no errors in transmission.

5.1. Introduction:

Each node consists of a host, which could be a terminal or an intelligent computer, and a loop interface.

The loop interface, see fig. 5.2, receives messages to be transmitted from the host and delivers messages to the host; it also relays messages addressed to and by other nodes. Three buffers are used for this purpose; one will hold the outgoing messages, buffer "A", another one will act as a delay buffer, "B", to delay any messages which arrive through the transmission channel and are to be relayed while buffer "A" is transmitting. The third buffer "C" holds any messages to be delivered to the host.

Messages which require transmission at a certain node may have access to the transmission channel according to some rule of channel assignments to resolve any contention with traffic relayed through buffer "B". The DLCN (5.1) loop has preset a fixed rule for channel assignment which gives priority to messages from buffer "A" if there is no message from "B" being serviced at the loop interface at the instant of "A"'s message arrival; otherwise any channel traffic is delayed in buffer "B" and will retain priority only if the buffer is about the overflow.

A node with its 3 buffers and various parameters is shown in Fig. 5.2:

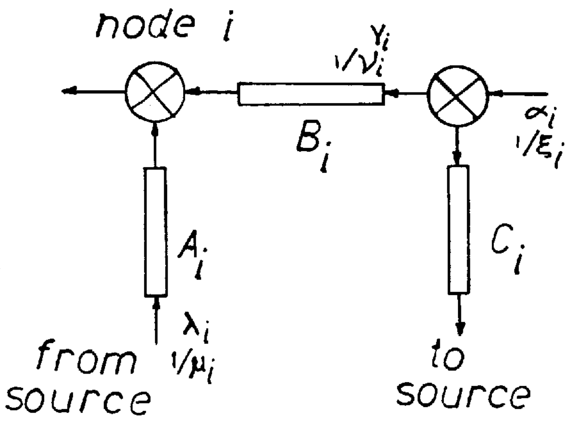


Figure 5.2.

The notation used for our analysis is listed below:

C	channel speed in characters/sec.
N	number of nodes.
λ_i	source message rate to buffer "A" at the ith node (messages/sec).
$1/\mu_i$	average message length from source at node i (characters/message)
γ_i	message rate to buffer "B" (messages/sec).
$1/\nu_i$	average message length to buffer "B" (characters/message).
α_i	message rate to buffer "C".
$1/\xi_i$	average message length to buffer "C".
λ_{ik}	proportion of messages generated at node i and destined to node k.
H	number of characters in the leader.
W_{A_i}	average time spent in the system by a message from buffer "A" of node i.
W_{B_i}	same as above, but from buffer "B".
W	average message delay due to buffer delays only.
T_h	message header inspection time.
T	total average message delay.
$\overline{X}_{A_i}$	average message duration (seconds) from buffer "A" of node i.
$\overline{X}_{B_i}$	average message duration (seconds) from buffer "B" of node i.
$\overline{X}^2_{A_i}$	second moment of message duration from buffer "A" of node i.
$\overline{X}^2_{B_i}$	second moment of message duration from buffer "B" of node i.

5.2. The Model:

The model described here starts by looking at the behaviour of each node independently. However, the traffic pattern arriving to the "B" buffer depends entirely on the other nodes.

Two protocols for handling messages on the communications channel are dealt with:

- Protocol 1. Messages are removed at their destination; the destination is responsible for sending acknowledgements.
- Protocol 2. All messages sent by a node are removed from the loop by the same node after they have made a complete cycle of the loop. This allows a broadcasting mode.

In our analysis, protocol-2 will be treated as a special case of the more general case, protocol-1.

We shall first consider a single node (i). If we assume buffer "A" be infinite, the sum of messages in the two-buffers A and B, see fig. 5.3, is the same for every discipline. This is assuming a work-conserving, non preemptive service. By work conservation, we mean that no work (service requirement) is created or destroyed within the system; for example destruction of work could occur if a message was to leave the system before completing transmission, and to create work might correspond to a server standing idle in the face of a non empty queue.

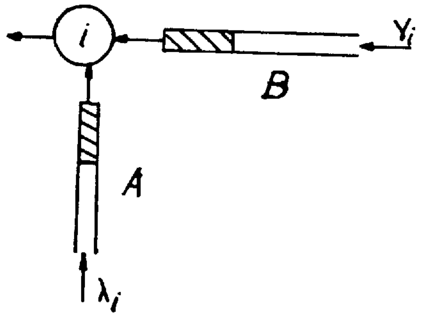


Fig. 5.3.

If we merge the two buffers "A" and "B", in theory, they would form one infinite buffer with input traffic or message rate $(\lambda_i + \gamma_i)$. We shall first consider a general arrival distribution for each "A" and "B", where the processes $[\lambda_i]$ and $[\gamma_i]$ are independent.

The problem of assignment here is to find a decision rule for assigning control to either channel "A" or "B". We shall achieve through the following analysis the best decision rule based on optimizing the average waiting time "W", where "W" is the average delay in buffers "A" and "B" encountered by a message from source to destination.

To find an expression for W, we shall define the conservation laws for G/G/1 (General arrival pattern/General service pattern/single server) system with multiple resources.

By definition:

U(t) $\triangleq$ the unfinished work in the system at time t.

$\triangleq$ the remaining time required to empty the system of all messages present at time t.

U(t) thus represents the interval of time required to empty the system completely if no new customers were allowed to enter after the instant t.

U(t) is also independent of the order of service (as long as the server is busy when a customer is in the system), such systems we have referred to as work conserving.

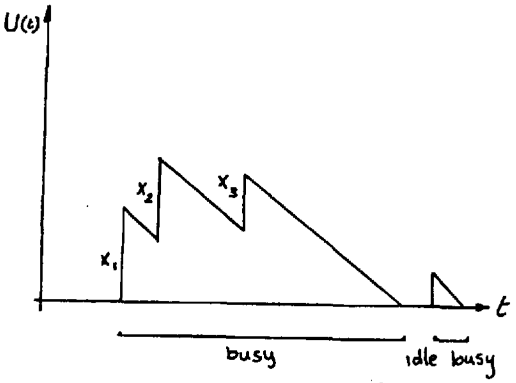


Figure 5.4.

Figure 5.4 shows that U(t) is a function which has the following properties:

1. It decreases at a rate of 1 sec/sec whenever U(t) > 0.
2. Remains saturated at zero when it hits the horizontal axis.
3. Takes vertical jumps at the arrival instants in amounts (x_i) equal to the service requirements brought in by arrivals.

Therefore, regardless of the order of service (service dependent or not), U(t) will not change: this is true for G/G/1 as well as M/G/1 (Markovian arrival pattern/general service pattern/single server) (5.2) and (5.3).

$\bar{U}$, the average unfinished work could be expressed by:

$$\bar{U} = W_O + \sum_{p=1}^{P} \bar{x}_p \, E[N_p]$$

where P sources generate customers into one single queue with one server. Each source p (p = 1,2,...P) has an average duration of messages $\bar{x}_p$. W_O is the residual work (5.4) due to customers in service. N_p is the number of messages in group p and E is the expectation.

Applying Little's theorem so that $E[N_p] = \lambda_p W_p$ we finally get the conservation law for U in the form:

$$\bar{U} - W_O = \sum_{p=1}^{P} \rho_p W_p \qquad (1)$$

Equation (1) gives a conservation law which puts a linear equality constraint on the set of average waiting times W_p. Any attempt to modify the queueing discipline so as to reduce one of the W_p will force an increase in some of the other W_p; however this should not be an even trade since the weighting factors for the W_p are generally distinct.

Applying equation (1) to our two message streams A & B, see fig.3, we get for a node i:

$$\rho_{B_i} W_{B_i} + \rho_{A_i} \cdot W_{A_i} = C_i \qquad (2)$$

where $C_i = \overline{U}_i - W_{O_i}$.

This constant C_i is the general unknown of the G/G/1 system. However we shall not concern ourselves to evaluate it since we shall only be concerned with the linear equality constraint.

Also ρ_{A_i} and ρ_{B_i} are the utilization factors for streams A and B and are given by:

$$\rho_{A_i} = \lambda_i \cdot \overline{x}_{A_i}$$

$$\rho_{B_i} = \gamma_i \cdot \overline{x}_{B_i}$$

5.3. Minimum Average Delay and the Assignment Rule.

In order to obtain an expression for W, we have to observe the loop as a whole. A message originating at node i and destined to node k will encounter a delay in buffer A of node i and a sum of delays $W_{B_{i+1}}, W_{B_{i+2}}, \ldots, W_{B_{k-1}}$. However the values of W_B traversed by the message will depend on two factors:

1. The traffic pattern of the loop affecting the γ's and the $\overline{x}_B$s.
2. The assignment rule which divides the channel between the A and the B traffic. W_B will be minimal if B had absolute priority over A and maximum delay if A had priority over B.

To deal with the first factor we shall introduce a notation which will help in analysing the loop traffic.

Consider λ_{ik} to be the proportion of traffic rate λ_i which is destined and removed by the destination node k.

Thus

$$\lambda_i = \sum_{\substack{k=1 \\ k\neq 1}}^{N} \lambda_{ik} \tag{3}$$

λ_{ik} is the traffic generated at node i which will affect all the nodes intervened between nodes i and k in case of protocol-1 defined earlier, while in case of protocol-2 all nodes other than node i will suffer a traffic λ_i, assuming node i to be the sole active node on the loop.

To be able to calculate the amount of traffic traversing each node we define $\delta_{i_{(kl)}}$ and δ_i. These are expressed by the following equations:

For protocol-1

$$\delta_{i_{(ik)}} = \begin{cases} 0 & \text{if } i \notin J_{kl} \\ 1 & \text{if } i \in J_{kl} \end{cases} \tag{4}$$

where $$J_{kl} = \{k+1,\ k+2,\ \dots\ l-1\} \tag{5}$$

and for protocol-2

$$\delta_i = \begin{cases} 0 & \text{if } i \notin J \\ 1 & \text{if } i \in J \end{cases} \tag{6}$$

Where $$J = \{k+1,\ k+2,\ \dots\ ,\ k-1\} \tag{7}$$

the traffic rate passing through node i (γ_i) would then be expressed by

$$\gamma_i = \begin{cases} \sum_{k=1}^{N} \sum_{l=1}^{N} \lambda_{kl} \cdot \delta_{i(kl)} & \text{for protocol-1} \\ \sum_{k=1}^{N} \lambda_k \cdot \delta_i & \text{for protocol-2} \end{cases} \tag{8}$$

Now we can evaluate $\bar{x}_{B_i}$, the average duration of messages arriving to buffer B of node i.

$$\begin{aligned} \bar{x}_{B_i} &= \frac{\sum_{k=1}^{N} \sum_{l=1}^{N} \lambda_{kl} \cdot \bar{x}_{kl} \cdot \delta_{i(kl)}}{\gamma_i} & \text{for protocol-1} \\ &= \frac{\sum_{k=1}^{N} \lambda_k \cdot \delta_i \cdot \bar{x}_k}{\gamma_i} & \text{for protocol-2} \end{aligned} \tag{9}$$

where $\bar{x}_{kl}$ is the average message duration (in secs.) originating at node k and destined to node l, and $\bar{x}_k$ is the message duration from node k. The former expression for $\bar{x}_{B_i}$ could still however be used for protocol-2 if different message durations are destined to different nodes, except that $\delta_{i(kl)}$ would have to be replaced by δ_i.

Similarly the higher moments of $\bar{x}_{B_i}$ could be calculated by replacing $\bar{x}_{kl}$ in equation (9).

Using Little's theorem, we could now express the average message delay W, due to delays encountered in buffers A and B.

$$\left(\sum_{i=1}^{N} \lambda_i \right) \cdot W = \sum_{i=1}^{N} \lambda_i \cdot W_{A_i} + \sum_{i=1}^{N} \gamma_i \cdot W_{B_i} \tag{10}$$

However, we can express $\lambda_i W_{A_i}$ in terms of W_{B_i} from equation (2).

$$\rho_{B_i} \cdot W_{B_i} + A_i W_{A_i} = C_i$$

$$\lambda_i W_{A_i} = C_i' - \gamma_i' W_{B_i} \qquad (11)$$

where $$C_i' = \frac{C_i}{\bar{x}_{A_i}}$$

and $$\gamma_i' = \gamma_{i'} \frac{\bar{x}_{B_i}}{\bar{x}_{A_i}}$$

from (10) and (11) we obtain:

$$(\Sigma\lambda_i) \cdot W = \sum_{i=1}^{N} C_i' + \sum_{i=1}^{N} W_{B_i} \cdot (\gamma_i - \gamma_i')$$

so $$W = \left[\sum_{i=1}^{N} C_i' + \sum_{i=1}^{N} W_{B_i} (\gamma_i - \gamma_i') \right] / \lambda \qquad (12)$$

where $$\lambda = \sum_{i=1}^{N} \lambda_i$$

Our goal is to minimize the total waiting time W, and this is a linear expression in W_{B_i} as seen in equation (12). The coefficient of W_{B_i} in equation (12) becomes

$$K_i = \gamma_i \left[1 - \frac{\bar{x}_{B_i}}{\bar{x}_{A_i}} \right] = K. \left[\bar{x}_{A_i} - \bar{x}_{B_i} \right]$$

where $$K = \gamma_i / \bar{x}_{A_i} \quad .$$

To minimize W, the right hand side of equation (12) should be optimized. This is achieved by minimizing W_{B_i} when K_i is positive and maximizing it when K_i is negative.

W_{B_i} could have a minimal value if B had priority over A and maximum if A had priority over B.

Looking at the components of K_i (we shall denote as the cost factor), we are fortunate to find that the components $\bar{x}_{A_i}$ and $\bar{x}_{B_i}$ are both terms which could be locally determined at each node.

We could now state our decision rule for assigning traffic to the A and B channels.

Each node i should sample averages $\bar{x}_{A_i}$ and $\bar{x}_{B_i}$ to calculate the cost factor K_i. If K_i is negative ($\bar{x}_{B_i} > \bar{x}_{A_i}$), priority is assigned to A; if K_i is positive ($\bar{x}_{A_i} > \bar{x}_{B_i}$), priority is assigned to B; and if K_i is null it is obvious that assignment has no effect.

However, if the buffers (A or B) had limited capacity, this will impose a boundary to the decision rule in order to avoid buffer overflow. It is necessary not to lose any of the traffic passing through the B buffer. Therefore we shall only be concerned with having buffer limits on the B channel.

To calculate the values of W_{B_i} we shall assume the message arrival distribution at both the A and B channels to be Poisson (M/G/1). In the following section we shall derive expressions for W_{B_i} and the average message delay W, based on our M/G/1 assumption.

5.4. Calculating the average message delay

From the definition of unfinished work, U is independent of the order of service, so, by implementing an FCFS (first come first served) system, the average unfinished work for Poisson arrivals must equal the average waiting time for messages, which we denote by W (5.4). The value for this quantity is given for an M/G/1 system and is known as the Pollaczek-Khinchin (P-K) mean value formula.

$$W = \frac{W_O}{1-\rho} \text{ where } W_O = \frac{\overline{x^2}}{2\bar{x}} \tag{14}$$

Thus: $$\bar{U} = W = \frac{W_O}{1-\rho}$$

substituting in equation (1) we get

$$\sum_{p=1}^{P} \rho_p W_p = \begin{cases} \frac{\rho W_O}{1-\rho} & \rho < 1 \\ \infty & \rho \geq 1 \end{cases} \tag{15}$$

Equation (15) gives a conservation law for any M/G/1 system and any non-preemptive, work-conserving queueing discipline. The constant C_i in equation (2) could now be calculated.

$$C_i = \frac{\rho_i \cdot W_{O_i}}{1-\rho_i} \tag{16}$$

and $$\rho_i = \rho_{A_i} + \rho_{B_i} = \lambda_i \cdot \bar{x}_{A_i} + \gamma_i \cdot \bar{x}_{B_i} \tag{17}$$

$$W_{O_i} = \Sigma \frac{\rho \overline{x_i^2}}{2\bar{x}_i} = \frac{\lambda_i \overline{x^2}_{A_i}}{2} + \frac{\gamma_i \overline{x^2}_{B_i}}{2} \tag{18}$$

The values of W_{B_i} (the average waiting time in the B buffer of node i) could be calculated according to the priority assignment by implementing the well-known expressions for 'head on the line (HOL)' priority queues (5.5).

If the A channel have priority over B at a node i:

$$W_{B_i} = \frac{W_{O_i}}{(1-\delta_{B_i})(1-\delta)} \tag{19}$$

and if B had priority over A

$$W_{B_i} = \frac{W_{O_i}}{(1-\delta B_i)} \tag{20}$$

where $\delta_{B_i} = \rho_{B_i} = \gamma_i \cdot \bar{x}_{B_i}$

and $\delta = \rho_{A_i} + \rho_{B_i} = \lambda_i \cdot \bar{x}_{A_i} + \gamma_i \cdot \bar{x}_{B_i}$

The average waiting time W_{B_i} hence fluctuates between the two average values given in (19) and (20) depending on the priority assignment.

$$\frac{W_O}{(1-\delta_{B_i})} < W_{B_i} < \frac{W_{O_i}}{(1-\delta_{B_i})(1-\delta)} \tag{21}$$

The upper average value could be used on equation (12) if the 'cost factor' was negative and priority is assigned to buffer A_i. In this case the system will operate in a similar fashion to the DLCN described by Liu and Reames in (5.1), provided the B_i buffer does not overflow.

The lower average value of W_{B_i} is that achieved when the cost factor is positive and priority is assigned to the B_i buffer. This could certainly happen long before the B_i buffer is about to overflow. In practice this should happen if a node decides to send a class of long messages ($\bar{x}_{A_i} > \bar{x}_{B_i}$), say a data file, in a control environment characterized by short messages. To simplify the analysis we shall assume that both $\bar{x}_{B_i}$ and $\bar{x}_{A_i}$ never exceed the B buffer length B_i.

So, referring to figure 5.2: A_i will have priority if

$$\bar{x}_{A_i} < \bar{x}_{B_i}$$

and

$x_{i_1} <$ (B_i - the sum of x_{B_i} messages in B_i in characters)) where x_{i_1} is the message on top of the queue A_i.

In the case of a null cost factor $K_i(\bar{x}_{A_i} = \bar{x}_{B_i})$ equation (12) ceases to

$$W = \sum_{i=1}^{N} C_i'/\lambda \qquad (21)$$

which is a constant independent of priority assignment. Therefore the average waiting time is unaffected and any assignment rule could be implemented.

In practical terms, ($\bar{x}_{A_i} = \bar{x}_{B_i}$) means that the message duration distribution is symmetrical around the loop, i.e. the message lengths at all nodes have the same distribution. We shall now justify this result of priority independence by using a different approach.

For a loop with symmetric message length distribution we find that all generated messages have an average length $\bar{x}$. Applying this to equation (9):

$$\bar{x}_{B_i} = \frac{\Sigma_k \Sigma_i \lambda_{kl}.\bar{x} \cdot \delta_{i(kl)}}{\Sigma_k \lambda_k \cdot \delta_{i(kl)}} = \bar{x} \cdot \frac{\Sigma_k \lambda_k \cdot \delta_{i(kl)}}{\Sigma_k \lambda_k \cdot \delta_{i(kl)}} = \bar{x}$$

A special case of the conservation law given in equation (15) appears when $\bar{x}_p = \bar{x}$ for all p and for $\rho < 1$ hence

$$\sum_{p=1}^{P} \lambda_p \cdot W_p = \frac{\lambda W_0}{1-\rho} = \lambda W = Nq \qquad (22)$$

where $W = \frac{W_0}{1-\rho} = \text{Const}\big|_{\text{queue discipline}}$

and $Nq = \lambda W = \text{Const}\big|_{\text{queue discipline}}$

there, in the special case where $\bar{x}_p = \bar{x}$, the average number in the queue and the average waiting time are independent of the queue discipline.

For any node i equation (22) becomes

$$\lambda_i W_{A_i} + \gamma_i W_{B_i} = (\lambda_i + \gamma_i) W_i = C_i \qquad (23)$$

so from (23) and (12) we get

$$W = \frac{\Sigma C_i}{\sum_{i=1}^{N} \lambda_i} = \text{Const}\Big|_{\text{for any queueing discipline}}$$

5.5. The Total Average Message delay (T)

The total average delay is the average time taken between the arrival of a message to buffer A or an originating node, and the arrival of this message to a destination node. The components contributing to this total message delay will be:

1. Waiting time at buffer A of the origin node i.
2. Waiting times in each of the intervening nodes, at buffer B, from node i+1 to node k-1, before arriving at destination node k.
3. Inspection time T_h of the header, of length H, of each message at each intervening node.
4. The propagation delay T_p between origin and destination.

Components (1) and (2), we have obtained in the form of an expression for W and is expressed as:

$$W = \frac{1}{\lambda}\left[\sum_{i=1}^{N} C_i' + \sum_{i=1}^{N} W_{B_i} \cdot K_i\right]$$

In some analyses service time at each node is added; however, in this case we have included service time in the expression for W since W_{B_i} and W_{A_i} are the total times spent in the system.

The header inspection time T_h could be obtained from the relation given below:

T_h = [Average path length]·H/C and the "Average path length"

$$= \sum_{i=1}^{N} (\gamma_i+\lambda_i) / \sum_{i=1}^{N} \lambda_i$$

$$T_h = \frac{H \sum_{i=1}^{N} (\gamma_i+\lambda_i)}{C \sum_{i=1}^{N} \lambda_i} \qquad (24)$$

Then the final expression for the average message delay becomes:

$$T = W + T_h + T_p$$

but we can neglect T_p since it is infinitesimal for localized systems:

$$T = \frac{1}{\sum_{i=1}^{N} \lambda_i} \left[\sum_{i=1}^{N} C_i' + \sum_{i=1}^{N} W_{B_i} K_i + \frac{H}{C} \sum_{i=1}^{H} (\gamma_i + \lambda_i) \right] \qquad (25)$$

where $$C_i' = \frac{[\lambda_i \cdot \bar{x}_{A_i} + {}_i \cdot \bar{x}_{B_i}] W_{O_i}}{[1 - \lambda_i \bar{x}_{A_i} - \gamma_i \bar{x}_{B_i}] \cdot \bar{x}_{A_i}}$$

and $$W_{O_i} = \frac{\bar{x}^2_{A_i}}{2\bar{x}_{A_i}} + \frac{\bar{x}^2_{B_i}}{2\bar{x}_{B_i}}$$

$$K_i = \gamma_i \left[1 - \frac{\bar{x}_{B_i}}{\bar{x}_{A_i}} \right]$$

$$W_{B_i} = \begin{cases} \dfrac{W_{O_i}}{(1-\delta_{B_i})} & \text{if } K_i > 0 \\[2ex] \dfrac{W_{O_i}}{(1-\delta_{B_i})(1-\delta)} & \text{if } K_i < 0 \end{cases}$$

(see equations (20) for values of δ and δ_{B_i}).

For a Symmetric Loop

For a symmetric loop, we have for all i=1,2, ..., N

$$\lambda = \lambda_i$$

and (26)

$$\gamma = \gamma_i = \lambda(N/2-1)$$

and as proven before:

$$\bar{x} = \bar{x}_{A_i} = \bar{x}_{B_i}$$

This leads to an average waiting time

$$T = \frac{\overline{x^2}}{(2-N\ )(\bar{x})} + \frac{H}{2C} \tag{27}$$

which is obviously independent of the priority assignment and hence independent of W_{A_i} and W_{B_i}.

5.6. Practical implementation of the priority assignment rule:

We realize by now that our method of 'minimal delay priority assignment' is based on dealing with averages, while in the case of DLCN decisions are 'instantaneous' and depend on individual message lengths and the size of the B buffer (5.1).

We can further implement the instantaneous switching of the priority to the B buffer if the size of the first message on head of the A buffer queue is larger than the free space in the B buffer.

The procedure of priority assignment is explained in the following:

Providing the condition of expected buffer overflow does not occur, the priority assignment to the A or B buffers is based on sample averages of $\bar{x}_A$ and $\bar{x}_B$ at each individual node. Say with N_A and N_B messages being measured from streams A and B respectively.

Equation (12) could be written as:

$$K_i = \frac{\gamma i}{\bar{x}_{A_i}} (\bar{x}_{A_i} - \bar{x}_{B_i}) = K_i' (\bar{x}_{A_i} - \bar{x}_{B_i}) \qquad (28)$$

Accordingly, we can choose our hypothesis for the decision to be:

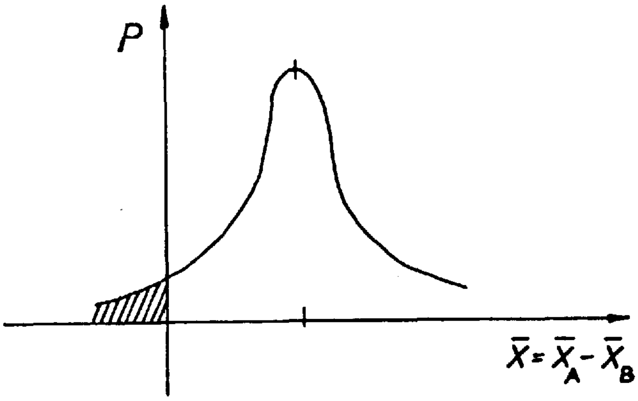
$$[\bar{x}_A - \bar{x}_B] > 0 \quad \text{for B having priority.}$$

To assign the priority to A, say, we will have to calculate the probability of a wrong decision 'P', or the probability of a wrong sign according to our hypothesis. This method is well documented in statistics literature as the "significance of the test" or "power of the test" (5.7)

The values taken by $(\bar{x}_A - \bar{x}_B)$ could be assumed to a normal (Gaussian) distribution, figure 5.5.

Our hypothesis is to test that $\bar{x} = \bar{x}_A - \bar{x}_B \leqslant 0$, where $\bar{x}_A$ and $\bar{x}_B$ are the means of the population.

Figure 5.5.



Each node measures a random sample of n observations x_A^1 , x_A^2 ... x_A^n and x_B^1, x_B^2...x_B^n with means $\bar{y}_A$ and $\bar{y}_B$ giving a difference average $\bar{y}$.The hypothesis we use now is $\bar{y} \leqslant 0$. As a means of assessing our degree of belief concerning the hypothesis that $\bar{y} \leqslant 0$, we could calculate the probability that a sample mean based on n observations would be equal to or greater than the observed value, $\bar{y}$, given $\bar{x} = 0$. If this probability, say α, was very small, we would then conclude that $\bar{x} > 0$. The smaller the value of α, the more likely it is to believe that $\bar{x} > 0$, contrary to our hypothesis.

We shall not proceed further with the calculation of this probability since its derivation is well described in the literature (5.7).

5.7. Comparisons and Results

To evaluate our new model we shall restrict ourselves to a Poisson arrival pattern.

We also take a reasonable assumption of exponential message length distribution whose probability distribution function is B(y) and probability density function is b(y).

If x is the average message length (in seconds), where $x = 1/\mu C$ and $1/\mu$ is the average length (in characters)

$$B(y) = 1-e^{-\mu y}$$

and

$$b(y) = \frac{dB(y)}{dy} = \mu e^{-\mu y} \quad .$$

Therefore, by taking the Laplace Transform of b(x)[B*(y)] we get:

$$B^*(s) = L[\mu e^{-\mu y}] = \frac{\mu}{s+\mu}$$

but

$$\bar{y} = -B^{*1}(0) = \frac{\mu}{(s+\mu)^2}\bigg|_{s=0} = \frac{1}{\mu}$$

is the average message length (1st moment) - Then the average message length (in seconds) becomes:

$$\bar{x} = \frac{1}{\mu C}$$

We then obtain the second moment in the same fashion:

$$B^{*2}(0) = (-1)^2 \cdot \bar{y}^2$$

$$B^{*2}(0) = \frac{2\mu}{(s+\mu)^3}\bigg|_{s=0} = \frac{2}{\mu^2}$$

Therefore

$$\bar{y}^2 = 2/\mu^2 \text{ and } \bar{x}^2 = 2/\mu^2C^2 \ .$$

The mathematical model was implemented as a computer program in FORTRAN. The aim was to calculate the average delay time a message suffers for three different cases.

These were:

1. The DLCN type network where at each node the A message stream always has priority over the B message stream unless the B buffer is almost full.
2. The optimum delay method, (ODM) where at each node priority is given to the stream with the smaller average message lengths, inspected at regular intervals.
3. The worst case delay; in other words, when the priority is always assigned to a channel in such a way that it contributes to a maximum average delay (opposite to the optimum delay assignment method).

The three above cases were considered for the two types of protocol described earlier, (see section 5.2). Results have shown that for both protocols and in most cases of asymmetrical traffic, the optimum delay method was superior to the DLCN and the worst case traffic, where the improvement in delay achieved more than 70% in most cases of high traffic. In the case of asymmetrical traffic, the three cases described above seized to the same delay pattern for each protocol.

The model assumed an M/G/1 traffic pattern at each node, with an exponential message length distribution. The line speed was 1 Mbit/sec. in all cases, and the average message lengths varied between 30 and 100 bits including a 10 bit header.

In the case of protocol 1, where each receiving node had to acknowledge the receipt of each message, the acknowledgement messages were not catered for in the model since this was an unnecessary added complication to the model, and also, since the time taken to generate acknowledgement messages was unpredictable. Therefore the results obtained for protocol 1 are not directly comparable to those of protocol 2. It was undesirable to be concerned at this stage with such a comparison, since the broadcasting capability obtained from protocol 2 was essential for the control applications in our interest.

Figures 5.6 and 5.7 show the delay characteristics for the ODM, DLCN and the worst case delay. The traffic in this case was symmetrical for both protocols 1 and 2 with an average message length of 30 bits and arrival rates varying between 100 and 5000 messages per second. It is interesting to see that the delay in all cases for each protocol coincides exactly. This asserts our previous conclusion that traffic assignment in a symmetric network has no effect on the average delay. The performance of each protocol in asymmetric cases of traffic are dealt with separately.

- Protocol-1:

As soon as the traffic pattern and the average message lengths start to change, the delay characteristics for each case start to behave differently. Figure 5.8 shows an asymmetrical case of a network comprising 6 nodes. In this case, the majority of nodes are transmitting short messages (30 bits) at a relatively high rate (500-7500 messages per sec.), while the minority of nodes (2 nodes) are transmitting long messages (90 and 100) at relatively low rates (100 bits per sec). Figure 5.8 shows a slight improvement in the delay characteristics of the ODM over the DLCN and the worst case which are almost identical. For this pattern of traffic, the average message length of traffic going through the B channel is greater than that of messages going through the A channel for the majority of users. This means that in the case of ODM, only the minority of nodes get their B traffic having priority over the A traffic and even for high traffic rates this would not make a considerable improvement in the delay characteristics of the ODM.

If the traffic pattern was reversed so that the minority of nodes generate short messages at a high traffic rate, while the majority generates long messages at low rates. This means that most nodes will assign priority to the shorter messages passing through the B channel in which case we would expect a considerable improvement in the delay pattern. Figure 5.9 shows an example of this traffic pattern where the minority (2 nodes) generate short messages (30 bits) at a high rate (5000 messages per second) and the majority of users generated long messages (100 bits) at a relatively low rate (100-1600 messages per second). At high traffic utilization we can see improvements in the delay of the ODM over the worst case by more than 35% and over the DLCN by more than 28%.

An increase in the traffic generation rate of the minority (with short messages) should still give further improvement in the delay of the ODM. Figure 5.10 shows the same traffic pattern but with a higher rate for the minority (10,000 messages per sec.) The ODM in this case is still showing further supriority over the DLCN and the worst case by over 46% and 48% respectively at high utilization. The improvement of the DLCN over the worst case at the same utilization is around 3.8% only.

- Protocol-2:

Figure 5.11 shows the delay pattern for protocol 2, where messages are only removed by their origin node, thus have a broadcast capability. In figure 5.11, the majority of nodes generate short messages (30 bits long) at a relatively high traffic rate (500-7500 messages per second), while the minority of nodes (2 in this case) generate longer messages (100 bits) at a low rate (100 messages per second). The traffic in this case is much denser than protocol 1, as one would envisage, thus the effect on the average delay is more profound. The delay of the DLCN is almost identical to the worst case, while a considerable improvement in the ODM was achieved (over 70% at high utilization).

Again by reversing the traffic pattern so that the minority of nodes generate short messages (30 bits long) at high rates (10,000 messages per second) and the majority generated long messages (100 bits) at low arrival rate (100-900 messages per second); the ODM still proves superiority over the DLCN and the worst case by 81% and 83% respectively as seen from figure 5.12.

If the traffic on the loop is always assigned priority over the traffic from the nodes, it is interesting to see that it would give better average delays over DLCN (nodes have priority) in most cases. Figure 5.13 shows a 6 node loop with a traffic pattern similar to that in figure 5.12.

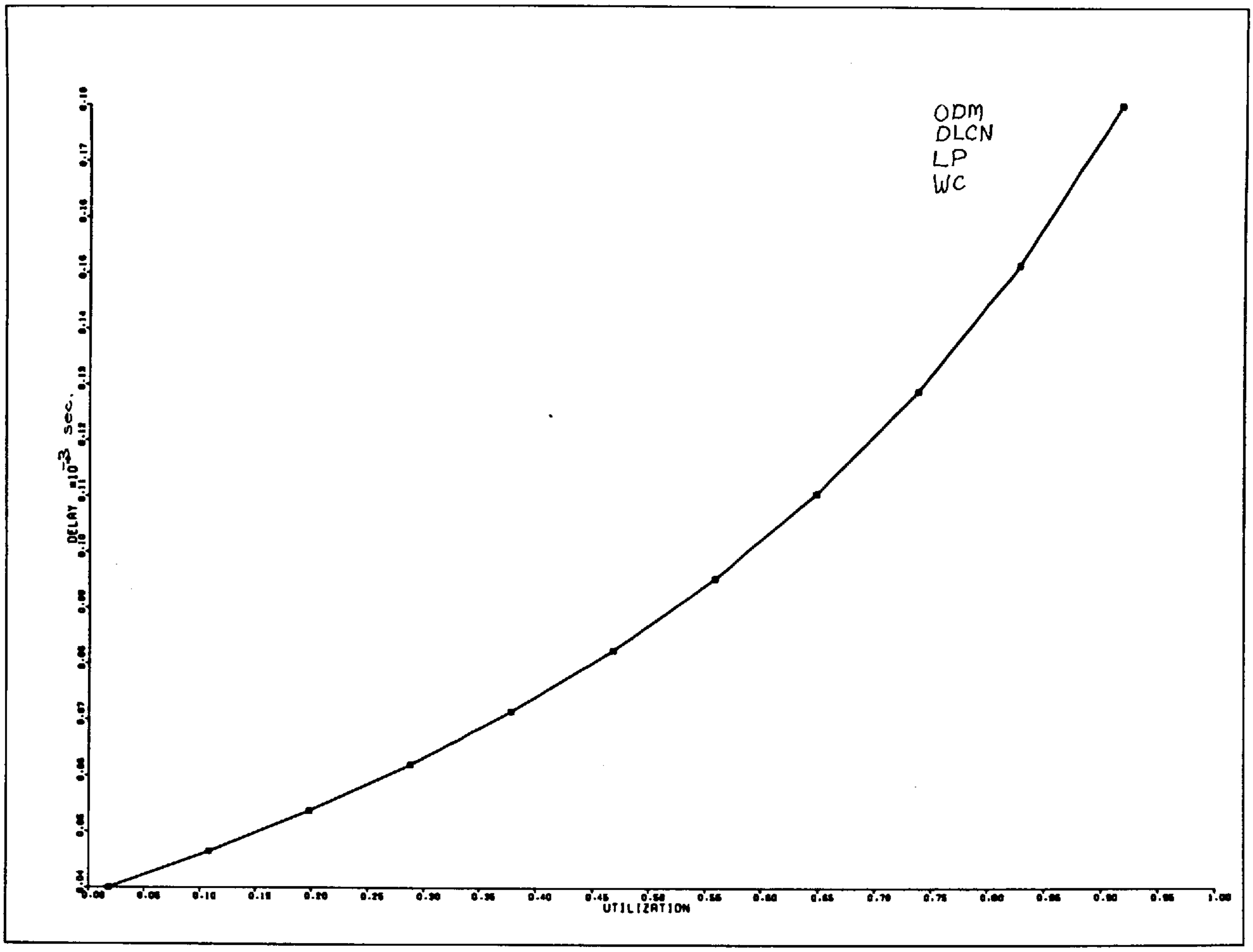


Fig. 5.6 Delay characteristics vs. network loading (or utilization) for a balanced traffic network (Protocol-1).
(All priority schemes behave similarly).

C = 1M bit/sec
$1/\mu$ = 30 bits
λ = 100-5000 messages/sec
N = 6

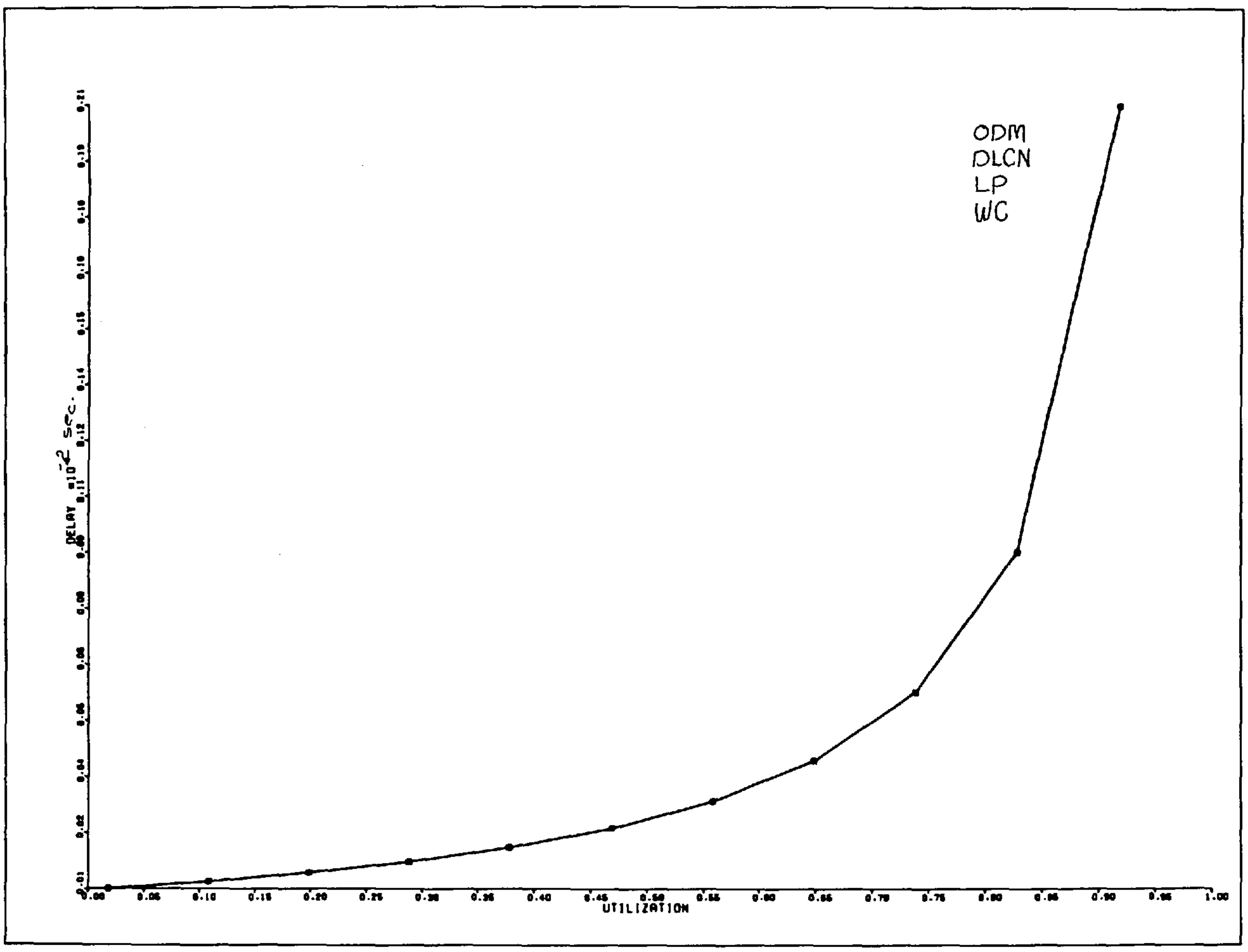


Fig. 5.7 Delay characteristcs vs network utilization for a balanced traffic network (protocol-2). (All priority schemes behave similarly).

c = 1 u bit/sec
$1/\mu$= 30 bits
λ= 100-5000 messages/sec
N = 6

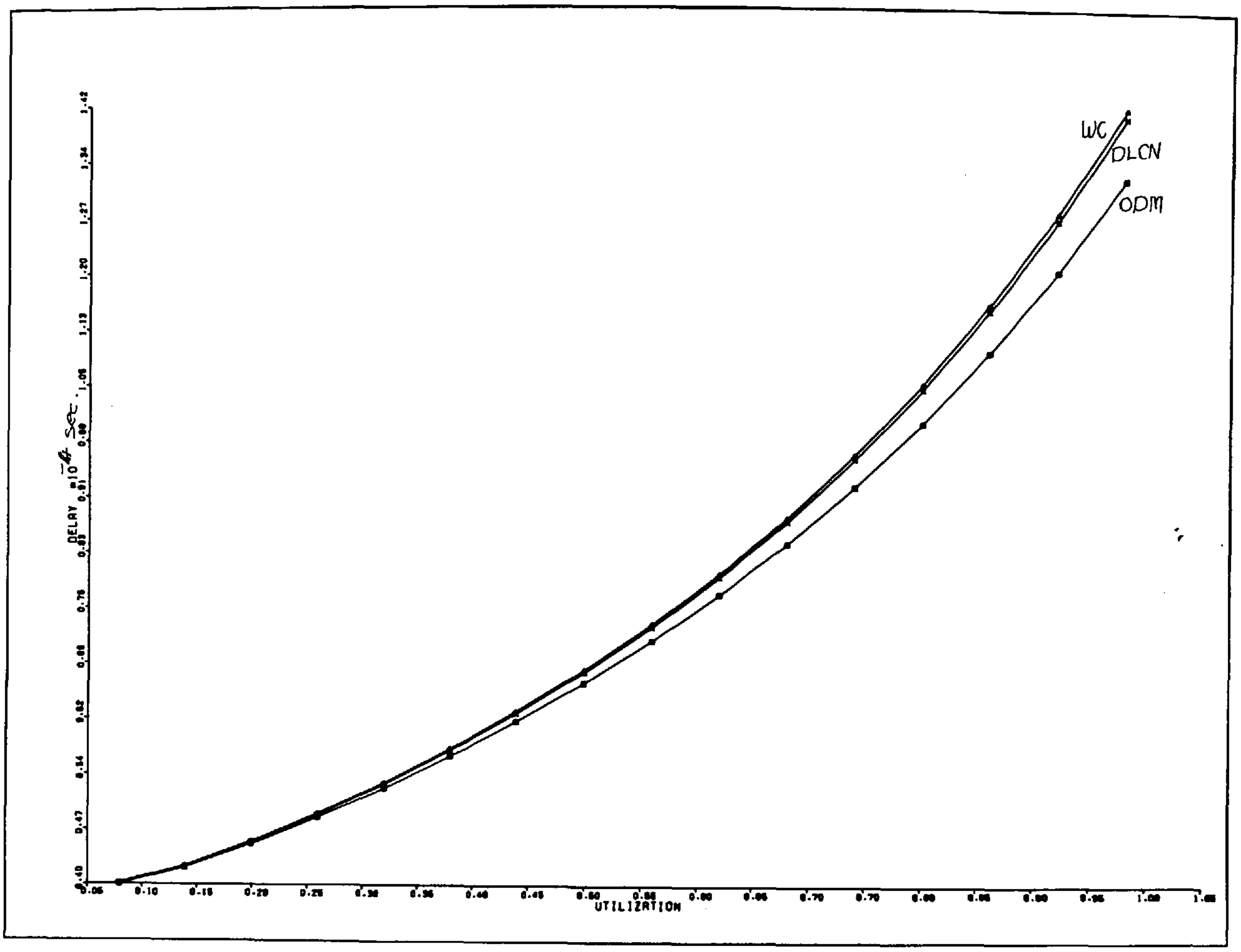


Fig. 5.8 Delay characteristics for asymmetrical traffic (protocol-1). Majority of nodes have high traffic and short messages; Minority have long messages and low traffic.

▲ Worst case delay (WC)
✗ Node priority (DLCN)
¤ Optimum delay method (ODM)

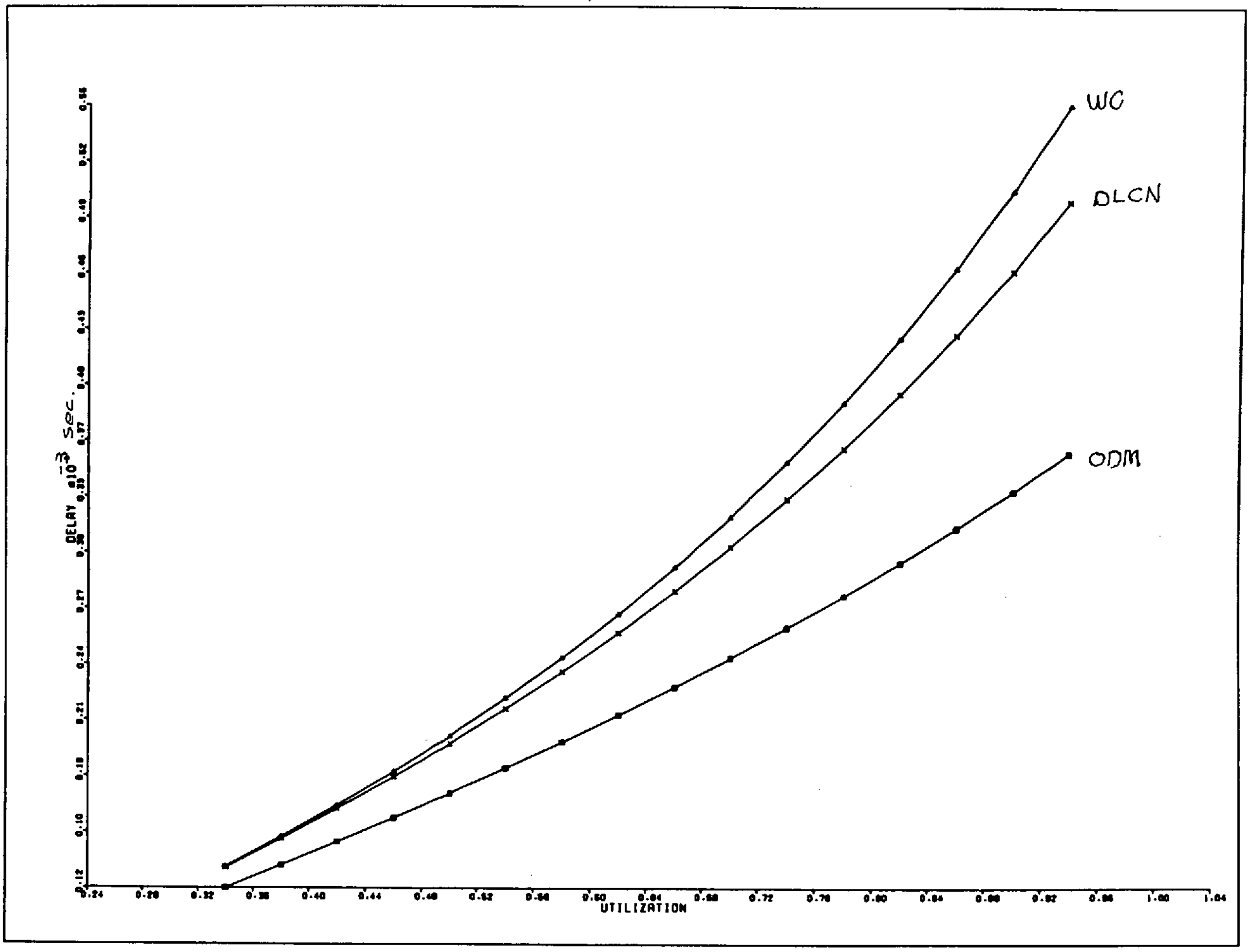


Fig. 5.9 Delay characteristics for asymmetric traffic (Protocol-1). Majority of nodes generate long messages at low traffic rates; Minority generates short messages at high traffic rates.

▲ WC
× DLCN
□ DDM

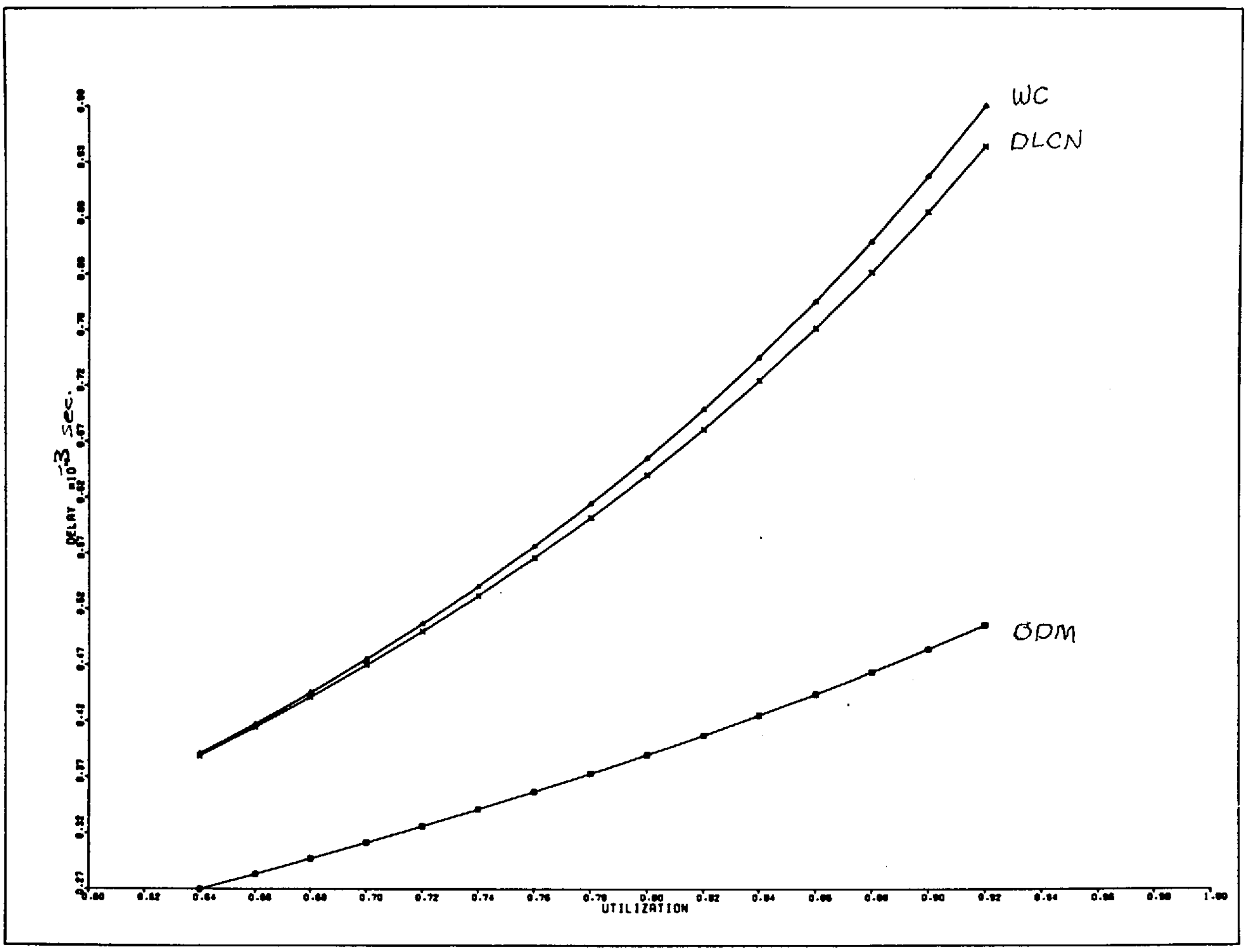


Fig. 5.10 Delay characteristics for asymmetrical traffic (Protocol-1). Same as fig. 5.9 with higher traffic rates generated by minority.

▲ WC
x DLCN
◻ ODM

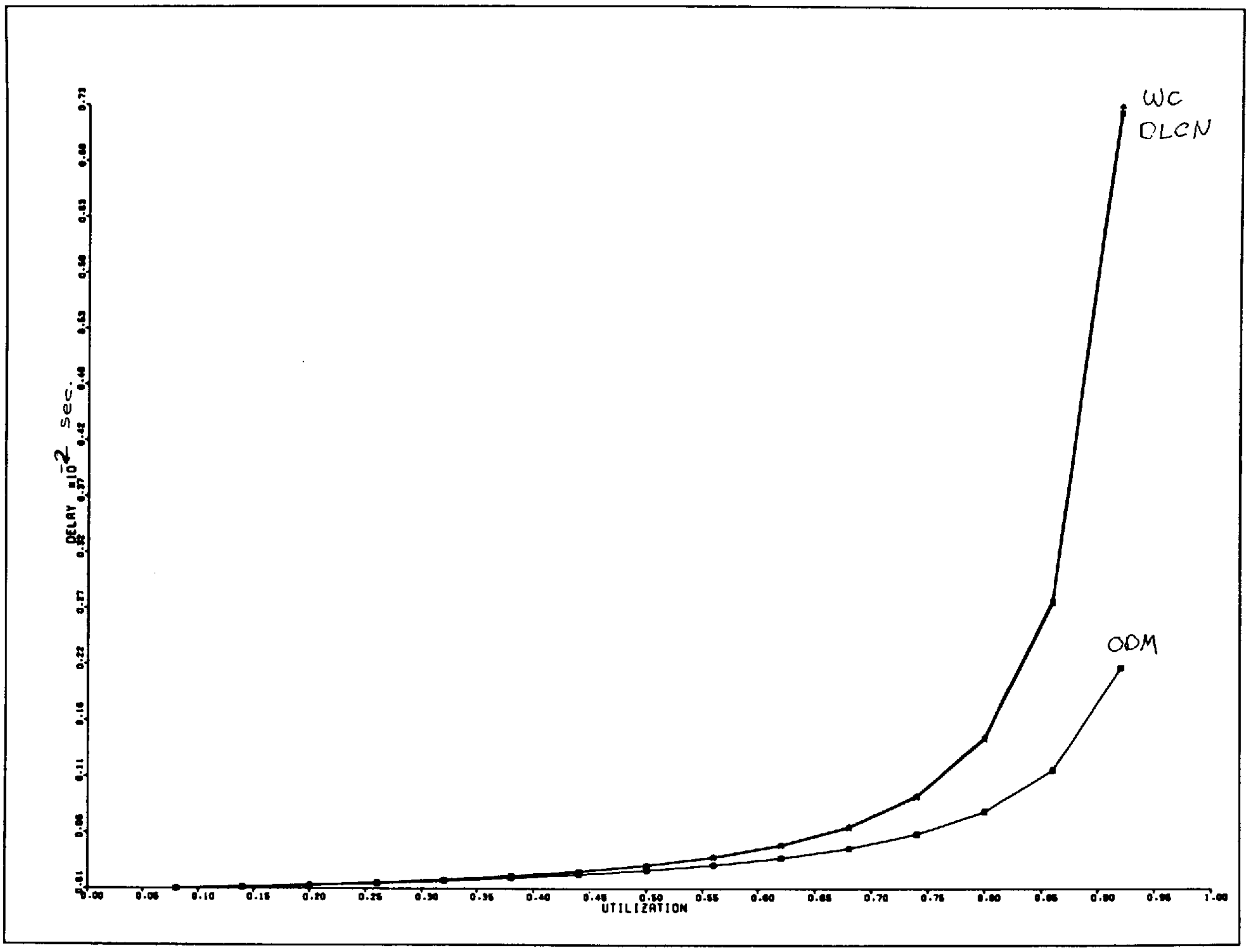


Fig. 5.11 Delay characteristics for asymmetric traffic (Protocol-2). Minority generates long messages at low traffics; Majority generates short messages and high traffic rates.

A WC
x DLCN
□ ODM
(WC and DLCN almost coincide).

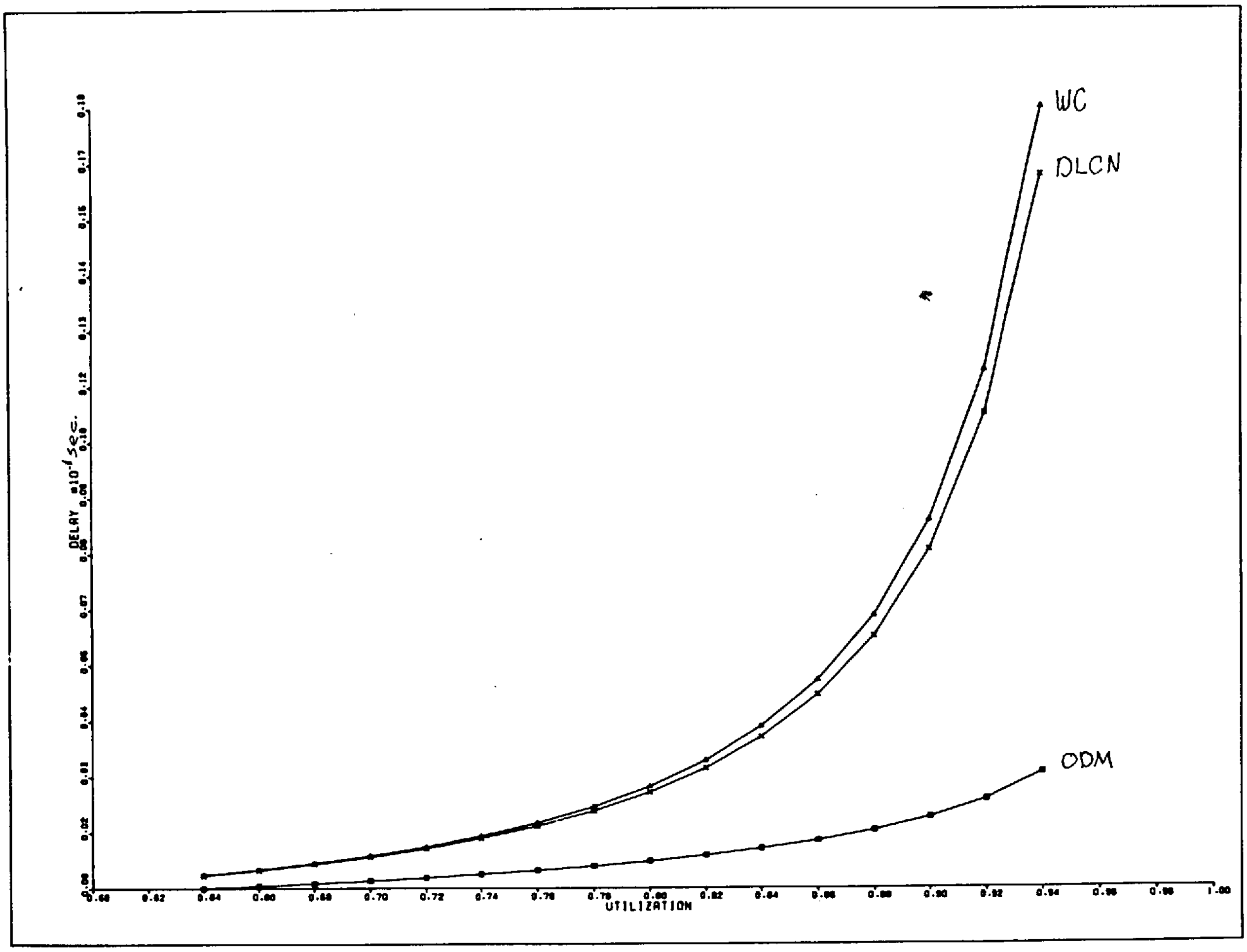


Fig. 5.12 Delay characteristics for asymmetric traffic (Protocol-2). Majority generates long messages with low traffic; Minority generates short messages with high traffic.

▲ WC
x DLCN
□ ODM

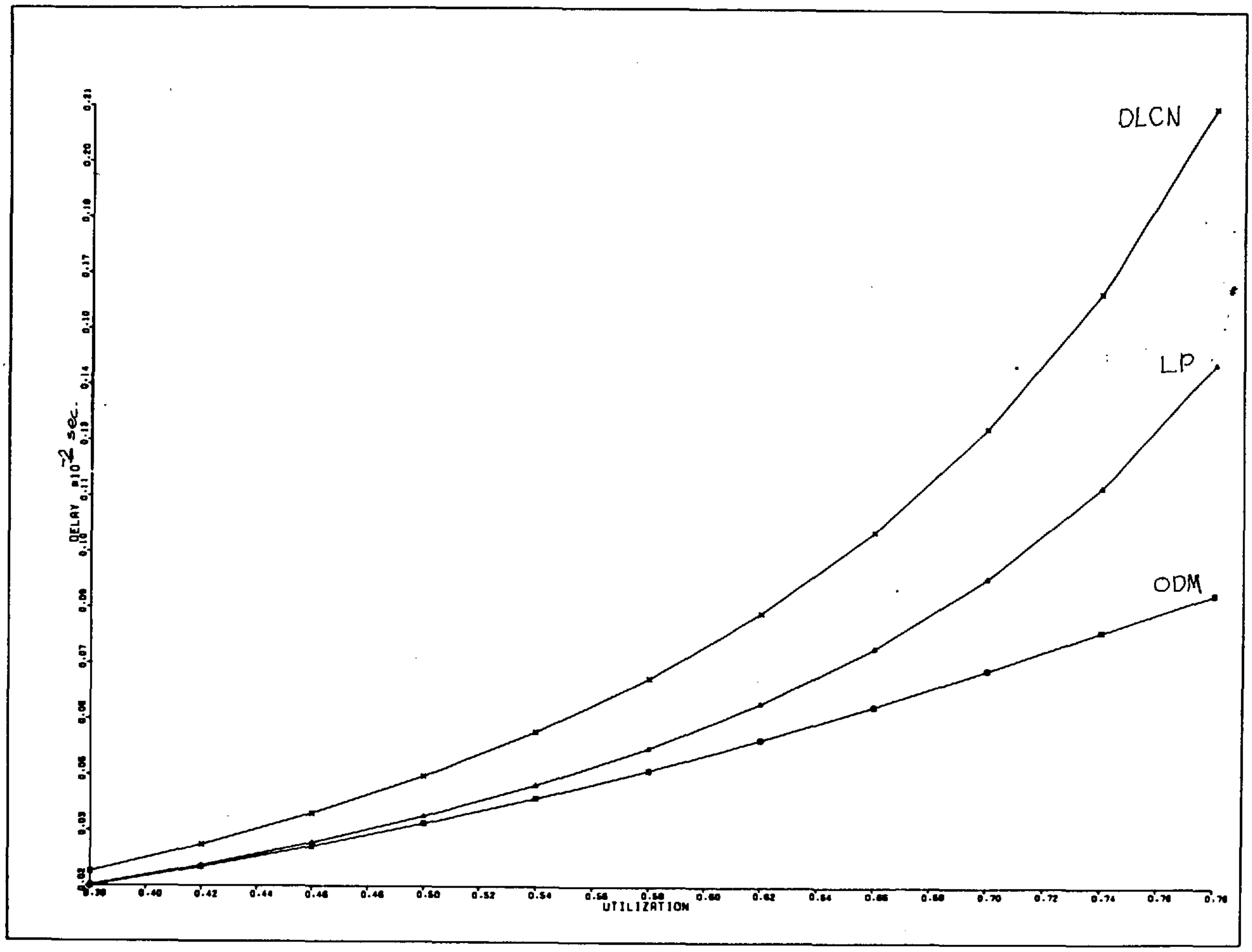


Fig. 5.13 Delay characteristics for an asymmetric network (Protocel-2) comparing the ODM, the DLCN and line priority cases.

x DLCN
△ Line priority
□ ODM

The 'ODM' still produces much lower delays and shows a 40% improvement at network loads of 75% over the 'line priority' case. The delay improvement of the 'line priority' over the DLCN is 29% at the same load.

To conclude our discussion we have seen that the 'ODM' has achieved a superior improvement over the DLCN and the worst case average delay. This improvement was greatest in the case of protocol 2 and for two types of environment; viz. that where the network is dominated by high rate control or alarm messages (which are characterized by short length) with a minority of long messages of slow rate (such as data files or user information) and that characterized by a majority of long messages at low traffic rates dominating the network, with a small number of nodes generating short control messages of a high rate. In the case of protocol 1, ODM still proves superiority although it was much less strong than in the second environment described above.

References

[5.1] Reames, C.C. and Liu, M.T. "A loop network for simultaneous transmission of variable length messages". Proc. 2nd Annual Symposium on Computer Architecture, Houston, Texas (Jan. 1975) 7-12.

[5.2] Kleinrock, L. "A Conservation Law for a wide class of queueing discipline". Naval Research Logistics Quarterly, 12, 181-192 (1965).

[5.3] Kleinrock, L. "Communication Networks: Stochastic Message Flow and Delay", McGraw-Hill (New York) (1964).

[5.4] Kleinrock, L. "Queueing Systems, Vol.II", J. Wiley, (1976), pp. 113-116.

[5.5] Cobham, A. "Priority Assignment in Waiting Line Problems", Operations Research, 2, 70-76 (1954).

[5.6] Kleinrock, L. "Queueing Systems, Vol.II" J. Wiley, (1976). pp. 125-126.

[5.7] Bard, Y. and Schatzaff, M. "Statistical Methods in Computer Performance Analysis", Current Trends in Programming Methodology, Vol.III, Prentice-Hall, (1978).

CHAPTER 6

Message Priority Classes in the Optimum Delay Method for a Loop Computer Network

6.1. Introduction

In control environments, some control processes acquire a minimum delay in receiving their orders, therefore message priority mechanisms are to be thought of. In a typical control application such as an electron probe microanalyser there are at least three categories of message in the order of importance shown below.

1. Alarm messages to initiate emergency action following fault conditions in the plant or computer system.
2. Control messages exchange between processors in the continuous mode of operation.
3. Low priority messages such as user information, printer output, data files, etc.

The implementation of priorities could be catered for by providing a field in the header of each message to indicate the priority. The length of this field depends on the number of priorities required.

Having seen the ODM method as the most efficient in terms of delay, it was interesting to examine the behaviour of the average delay for messages in different groups of priority.

A mathematical model has been developed in order to calculate the average delay for each priority group. Following the same notations used previously, the model assumed infinite A and B buffers. The model led to a new assignment rule for each node in order to give an optimum average delay of a general system G/G/1. However, based on the assignment rule, expressions for average delay were derived based on an M/G/1 model.

6.2. A Mathematical Model for an ODM with Priority Classes

Each node was allowed to generate a set of messages of different priority groups (1,2, ..., P). Stream A therefore received messages of each group with different traffic rates $(\lambda_1, \lambda_2, \lambda_3 \ldots \lambda_p)$ and with different average message durations $(\bar{x}_1, \bar{x}_2, \ldots \bar{x}_p)$. Consequently, stream B received traffic rates $(\gamma_1, \gamma_2, \ldots \gamma_p)$ and message durations $(\bar{y}_1, \bar{y}_2, \ldots \bar{y}_p)$. It was the job of each node then to assign the loop channel to messages of different groups and from different streams (A or B).

Messages from high priority groups had priority over all the groups below, and if it happened that messages from the same group required transmission simultaneously, then an assignment rule was necessary.

In the following sections, an overall assignment rule is established, and it was found to be identical for each priority group as the assignment rule previously established for the 'ODM'. To derive expressions for average delay in each class of priority, it was necessary to modify the general conservation rule for a priority model (see section 6.2.1.)

6.2.1. A modified conservation law for a single server queueing model with priority classes

The definition of unfinished work as described earlier (Section 5.2) considered the average amount of service the system had to go through for all priority classes. However, in order to be able to observe the behaviour of each priority class separately, it was necessary to be able to look at the unfinished work of that selected priority and all those priorities above it. This did not take into account all the priorities below, but considered the residual work which could have been happening to a message from any class of priority. The system was considered to be non-preemptive and work conserving. The unfinished work as seen by the system was referred to by U. We shall refer to the unfinished work (U_p) as the amount of work seen by an observer looking at priorities (p, p+1, ... P) at an instant of time t. Therefore:

$$U_p(t) = x_0 + \sum_{j=p}^{P} \sum_{i=1}^{N_j(t)} x_{ij} \qquad (1)$$

Where x_0 is the residual service of a customer found ahead, (i.e. already being serviced), x_{ij} is the service time of the ith message from priority class j and $N_j(t)$ is the number of messages in group j at time t.

Taking the expectations of both sides of equation (1) we get

$$E[U_p(t)] = [W_{0p}] + \sum_{j=p}^{P} \sum_{n_j=0}^{\infty} P[N_j(t)=n_j] \cdot \sum_{i=0}^{n_j} E[x_{ij}] \qquad (2)$$

We observe that $E[x_{ij}] = \bar{x}_j$ independent of the index i.

Taking t at random (and large) we may write:

$$\bar{U}_p = W_{O_p} + \lim_{t\to\infty} \sum_{j=p}^{P} \sum_{n_p=0}^{\infty} n_p P[N_j(t)=n_p] \cdot \bar{x}_j$$

$$= W_{O_p} + \sum_{j=p}^{P} \bar{x}_j E[N_j] \qquad (3)$$

Using Little's theorem, $E[N_j] = \lambda_j . W_j = \rho_j W_j / \bar{x}_j$. Since this result is valid for individual priorities as well, we conclude that

$$\bar{U}_p = W_{O_p} + \sum_{j=p}^{P} \rho_j W_j \qquad (4)$$

$\bar{U}_p$ is the average unfinished work for priority 'p' and priorities above it, and is still independent of their order.

Equation (4) could still be written in the form:

$$\sum_{j=p}^{P} \rho_j W_j = \bar{U}_p - W_{O_p} = CP_p \qquad (5)$$

where CP_j is a constant depending on the priority level, 'j', above which we are considering all traffic.

6.2.2. Expressions for Average Message Delay and Deduction of the Assignment Rule

We shall now derive a recursive formula for the traffic in each individual priority group, at each node, considering the highest priority group (P) and applying equation (5) to our model which includes messages from streams A and B we can write:

$$\rho_{A_P} \cdot W_{A_P} + \rho_{B_P} \cdot W_{B_P} = CP_P$$

and for the priority group (P-1) we get:

$$\rho_{A(P-1)} \cdot W_{A(P-1)} + \rho_{B(P-1)} \cdot W_{B(P-1)} + CP_{p} = CP_{p-1}$$

Therefore the recurrence formulae for any priority group j would be:

$$\rho_{Aj} \cdot W_{Aj} + \rho_{Bj} W_{Bj} = CP_{j} - CP_{j+1} = K_{j} \qquad (6)$$

Having derived the traffic formula of a node for each priority group based on the modified conservation law, we can now consider the total average delay in the network in terms of the delay in the B buffer.

The following equation gives the A delay (W_A) as a function of the B delay (W_B). From equation (6) we can easily derive for any node that:

$$\lambda_{j} \cdot W_{Aj} = \frac{K_{j}}{\bar{x}_{j}} - \gamma_{j} \cdot \frac{\bar{y}_{j}}{\bar{x}_{j}} \cdot W_{Bj} \qquad (7)$$

where λ_j and γ_j represent the traffic (in messages per second) of priority group j flowing through streams A and B respectively.

To calculate the total average waiting times for each priority group we shall again make use of Little's result. Thus the average waiting time (W_J) for a priority group (J) can be expressed as follows:

$$\left(\sum_{i=1}^{N} \lambda_{Ji}\right) \cdot W_{J} = \sum_{i=1}^{N} \lambda_{Ji} \cdot W_{AJ}^{i} + \sum_{i=1}^{N} \gamma_{Ji} \cdot W_{BJ}^{i} \qquad (8)$$

where the subscript 'i' in all variables indicate the node number.

From equations (7) and (8) we get W_J in terms of the B-channel delay:

Hence:

$$W_J = \frac{1}{\lambda_J}\left[\sum_{i=1}^{N} \frac{k_{Ji}}{\bar{x}_{Ji}} + \sum_{i=1}^{N} \gamma_{Ji}\left(1 - \frac{\bar{y}_{Ji}}{\bar{x}_{Ji}}\right) \cdot W^{i}_{BJ}\right] \qquad (9)$$

where $\lambda_J = \sum_{i=1}^{N} \lambda_{Ji}$

and $\bar{x}_{Ji}$ and $\bar{y}_{Ji}$ are the average message durations of messages in priority group J passing through the A and the B buffers of node i.

This equation leads to the same assignment rule obtained for the ODM without priorities. [section (5.3)] .

The assignment rule therefore becomes:

At any time, if messages in one of the streams (A or B) are of a higher priority group than messages in the other stream (B or A), then priority is strictly assigned to the channel with higher priority messages.

If both channels (A and B) had messages from the same priority group, the choice should be made according to the average message lengths of messages from this particular priority. In other words, if the average message length of messages in the A buffer is greater than those on the B buffer within a time 't', then priority is assigned to B and vice versa. Messages of the same priority group in channels A and B having the same average length will have no effect in producing an optimum delay.

6.2.3. The Total Average Message Delay for Messages in Each Priority Group

To be able to calculate the average message delay we had to assume an M/G/1 traffic model at each node. This assumption was proved to be a good approximation in networks with mixed traffic patterns (6.1)(6.2). The components forming the total delay of a message from source to destination are still the same as was shown in section (5.5). Namely:

1. The waiting time in buffer A of the originating node i.
2. The waiting time at each of the intervening nodes in buffer B, from node i+1 to node k-1, before arriving at the destination node k.
3. Inspection time (T_h) of the header of length H bits, of each message at each intervening node.
4. The propagation delay T_p between origin and destination.

Components (1) and (2) of the total average message delay were represented in equation (9) for a G/G/1 system. However, the values of W_B for each node depended on the assigned status.

We then proceed to recalculate the parameters of equation (9), representing the average delay in the A and B buffers, based on an M/G/1 assumption. This was essential since the constant value of K_{iJ} is undeterminable in the case of a G/G/1 model.

Equation (5) presents a modified conservation law for a G/G/1 model. Assuming that all priorities above any p^{th} priority followed an FCFS rule and had Poisson arrivals, then the average unfinished work is equal to the average waiting time for messages, hence:

$$\bar{U}_p = W_p = \frac{W_0}{1-\rho_p} \tag{10}$$

Equation (10) gives the average waiting time in the queue for messages from priority groups p, p+1, ... P. W_0 is the expected residual time of a message which is being serviced. This residual message could be from any priority group. Equation (11) below gives the value for W_0 at any node i.

$$W_0^i = \sum_{j=1}^{P} \rho_j \cdot \frac{\bar{x}_j^2}{2\bar{x}_j} \tag{11}$$

and $$\rho_p = \sum_{j=p}^{P} \rho_j = \sum_{j=p}^{P} \lambda_j . \bar{x}j$$

Therefore from equations 9, 10 and 11 we can deduce our new conservation law for an M/G/1 system with P priority groups, regarding priorities p to P only:

$$\sum_{j=p}^{P} \rho_j W_j = \frac{\rho_p W_0}{1-\rho_p} = CM_p \tag{12}$$

The recurrence formula for any priority group j (see equation 6) for an M/G/1 model becomes:

$$\rho_{A_j} \cdot W_{A_j} + \rho_{B_j} W_{B_j} = CM_j - CM_{j+1} = M_j \tag{13}$$

So the average delay of messages in a group J for a loop of N nodes with M/G/1 traffic pattern is finally:

$$\lambda_J W_J = \sum_{i=1}^{N} M_J^i/\bar{x}_{Ji} + \sum_{i=1}^{N} \gamma_{Ji}\left(1 - \frac{\bar{y}_{Ji}}{\bar{x}_{Ji}}\right) \cdot W_{BJ}^i \qquad (14)$$

The average delay in a B buffer for a message of priority 'p' is W_{B_p} (for one node). W_{B_p} is a minimum if stream B had priority over stream A, and maximum if stream A had priority.

If stream B had priority over stream A then the average B delay for messages of priority J becomes:

$$W_{BJ} = \frac{W_0}{(1-\delta_J)(1-\delta_{J+1})} \quad , \; J = 1,2, \; \ldots \; , P-1$$

where $\delta_J = \rho_B + \sum_{i=J+1}^{P} (\rho_{A_i} + \rho_{B_i})$

and $\rho_{J+1} = \sum_{i=J+1}^{P} (\rho_{A_i} + \rho_{B_i})$

while if stream A had priority over stream B, then:

$$W_{BJ} = \frac{W_0}{(1-\delta'_J)(1-\delta_{J+1})}$$

where $\delta'_J = \sum_{i=J}^{P} (\rho_{A_i} + \rho_{B_i})$.

Then W_{BJ} varies between the two limits set above:

$$\frac{W_0}{(1-\delta_J)(1-\delta_{J+1})} < W_{BJ} < \frac{W_0}{(1-\delta'_J)(1-\delta_{J+1})} \qquad (15)$$

The average delay (T_{h_J}) due to the message header inspection time again depends on the average path length and the header length (H): so

"Average header delay" = [Average path length of a message in priority group J]. H/C and the "Average path length"

$$= \sum_{i=1}^{N} (\gamma_{J_i} + \lambda_{J_i}) / \sum_{i=1}^{N} \lambda_{J_i}$$

Therefore:

$$T_{h_J} = H \cdot \sum_{i=1}^{N} (\gamma_{Ji} + \lambda_{Ji}) / C \cdot \lambda_J \qquad (16)$$

The propagational delay could be neglected since its value is infinitesimal in localized networks. So, the final expression for the average message delay in a priority group J becomes:

$$T_J = W_J + T_{h_J} \qquad (17)$$

Substituting (14) and (16) in (17) we get:

$$T_J = \frac{1}{\lambda_J} \left[\sum_{i=1}^{N} M_J^i / \bar{x}_{J_i} + \sum_{i=1}^{N} W_{B_J}^i \cdot \gamma_{J_i} \left(1 - \frac{\bar{y}_{J_i}}{\bar{x}_{J_i}}\right) + \frac{H}{C} \sum_{i=1}^{N} (\gamma_{J_i} + \lambda_{J_i}) \right] \qquad (18)$$

6.3. Comparisons and Results

A network comprising 6 nodes and a transmission speed of 1 Mbits/sec. is tested for asymmetrical traffic. Figure 6.1 shows the average message delay for the 'ODM' method with 10 priority classes. Figures 6.2 and 6.3 show the same network for the DLCN method and the 'line priority' method. The figures show that the delay behaviour in the 'ODM' is better than both of the other cases. Figure 6.4 shows a comparison between the three methods with two priority classes only.

It is worth mentioning at this stage, that the implementation of a system with priority classes, requires a series of parallel buffers in both the A and the B channels. Each buffer is assigned a priority class, and messages are sorted out on their entry. On the other hand a memory pool could be used and the buffers in this case become blocks of memory.

References

6.1 Kleinrock, L. "Queueing Systems: Vol.1", John Wiley & Sons - 1976.

6.2 Kleinrock, L. "Queueing Systems: Vol.2", John Wiley & Sons - 1976.

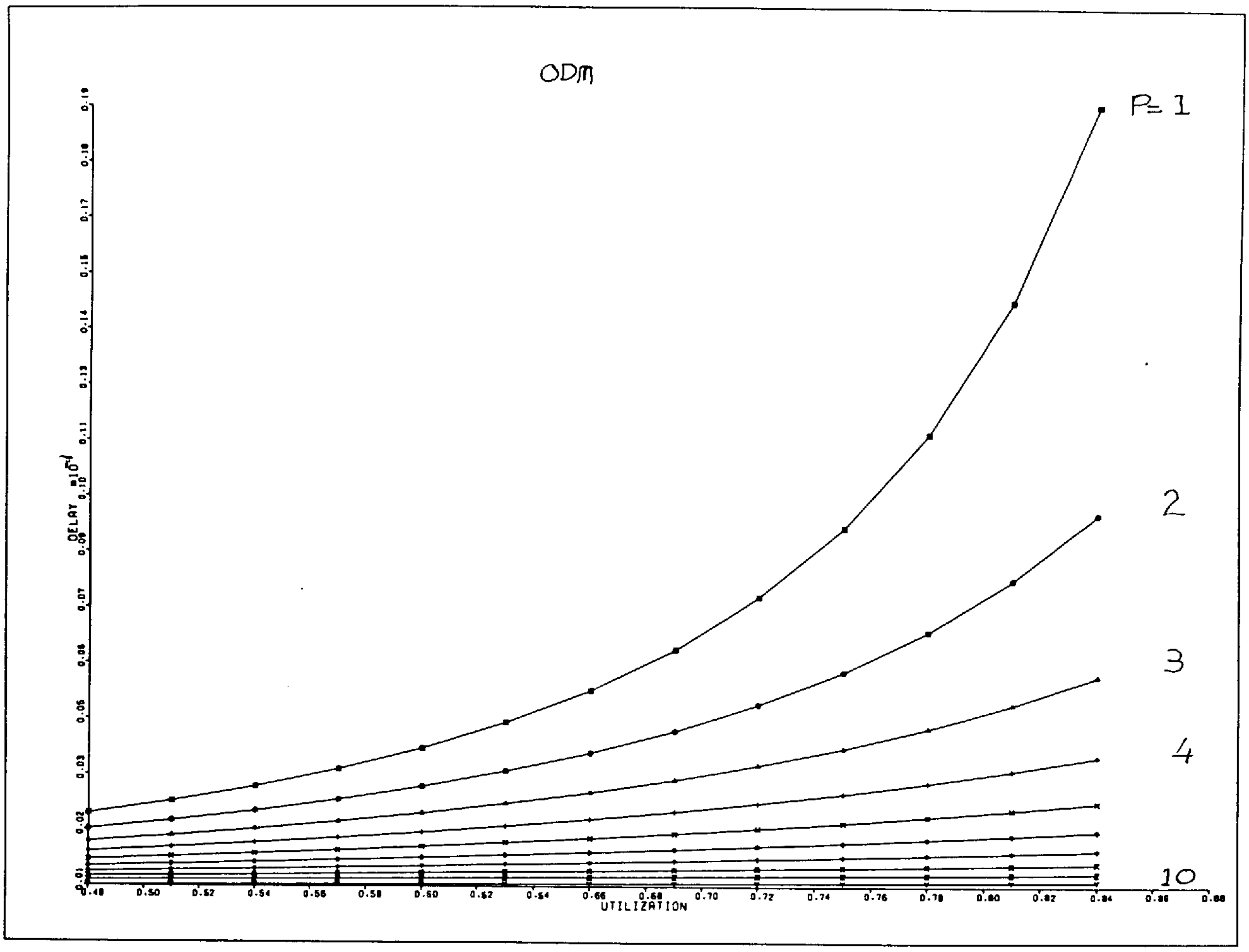

ODM
P=1
2
3
4
10
DELAY *10^-1
UTILIZATION
0.48
0.50
0.52
0.54
0.56
0.58
0.60
0.62
0.64
0.66
0.68
0.70
0.72
0.74
0.76
0.78
0.80
0.82
0.84
0.86
0.88

Fig.6.1

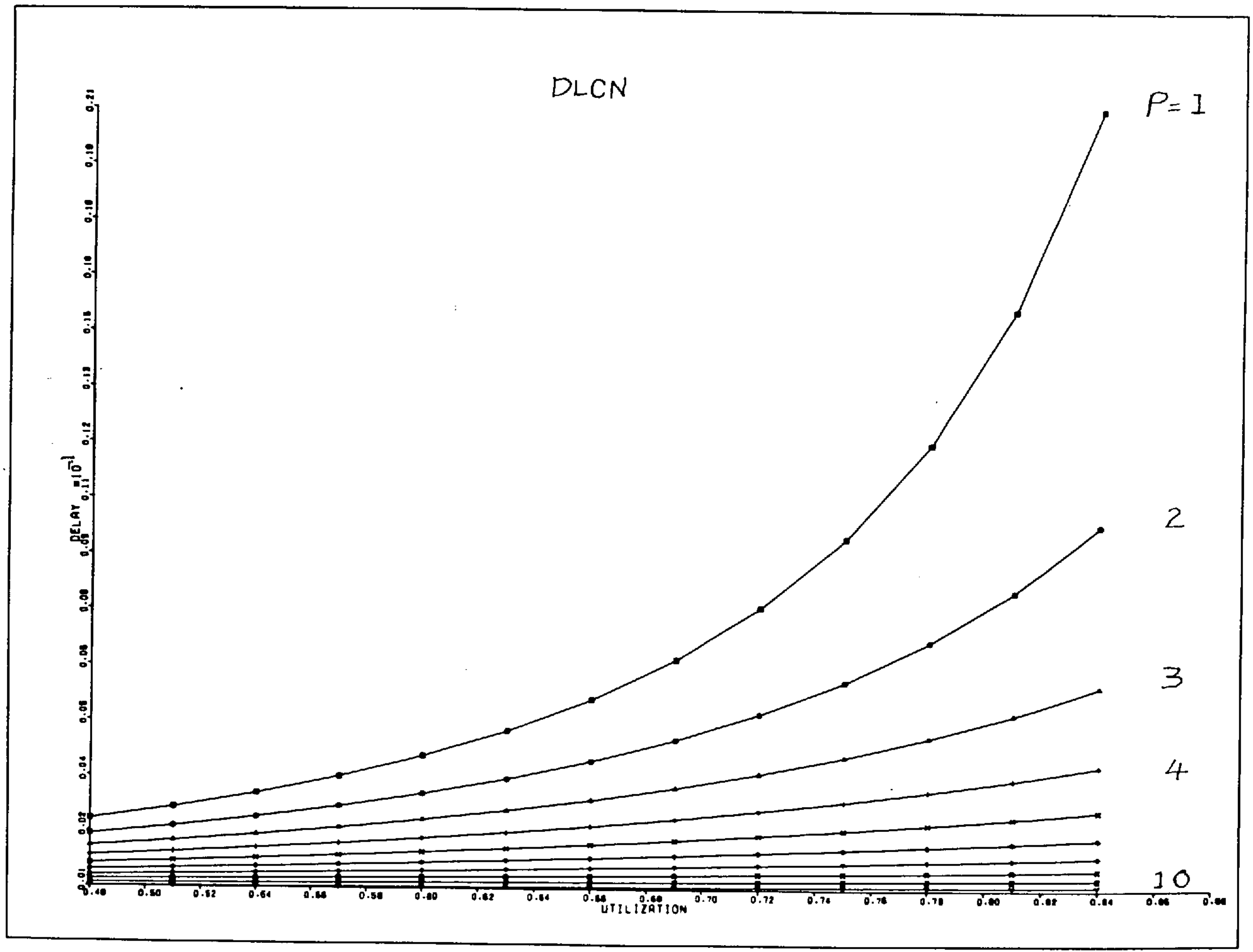

DLCN
P=1
2
3
4
10
DELAY *10^-1
UTILIZATION
0.48
0.50
0.52
0.54
0.56
0.58
0.60
0.62
0.64
0.66
0.68
0.70
0.72
0.74
0.76
0.78
0.80
0.82
0.84
0.86
0.88

Fig.6.2

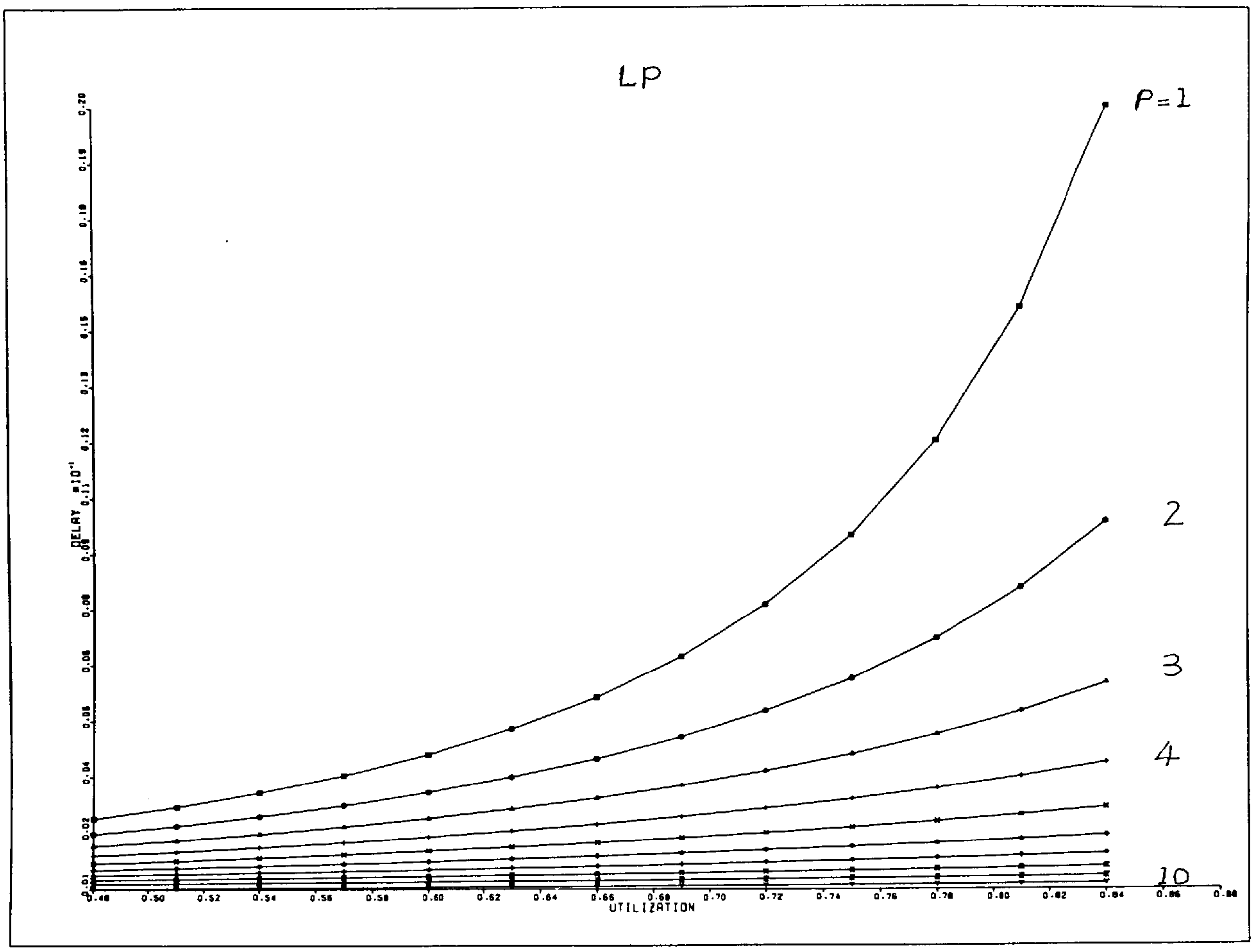

LP
P=1
2
3
4
10
DELAY *10⁻¹
0.20
0.02
UTILIZATION
0.48
0.50
0.52
0.54
0.56
0.58
0.60
0.62
0.64
0.66
0.68
0.70
0.72
0.74
0.76
0.78
0.80
0.82
0.84
0.86
0.88

Fig.6.3

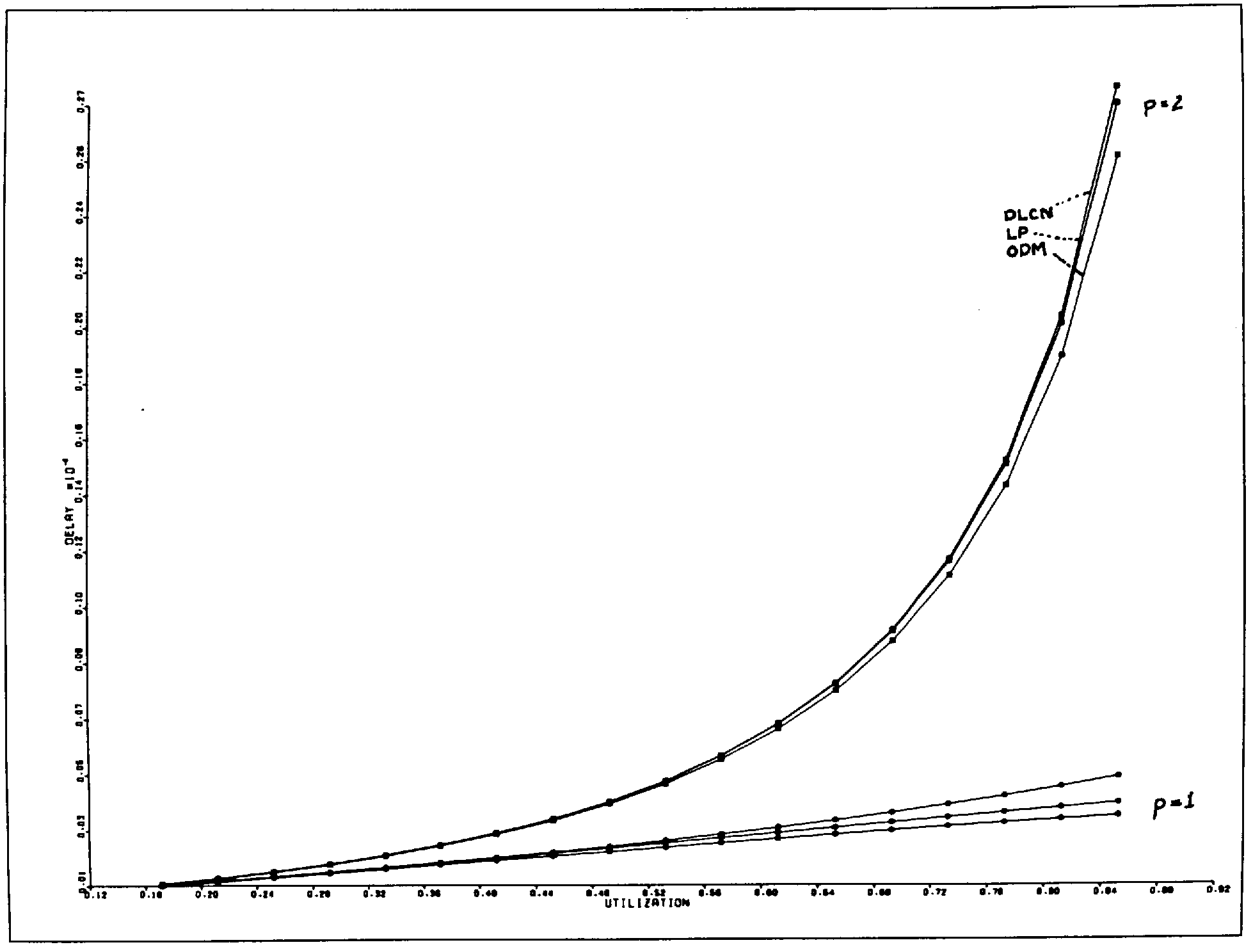

p=2
DLCN
LP
ODM
p=1
DELAY *10⁻¹
0.27
0.01
UTILIZATION
0.12
0.16
0.20
0.24
0.28
0.32
0.36
0.40
0.44
0.48
0.52
0.56
0.60
0.64
0.68
0.72
0.76
0.80
0.84
0.88
0.92

Fig.6.4

CHAPTER 7

Analysis of a 4 node loop network controlling an electron probe microanalyser

The control processes discussed in Chapter 3 were assigned to 4 processors. A loop computer network is designed for the communications. Four microcomputers (DEC's LSI-11) are used for this purpose.

The following sections describe the formats of messages exchanged between the four nodes. Typical traffic rates are deduced from practical specimens. The delay performance of the network based on the 'optimum priority scheduling' method is compared to other delay insertion techniques.

Figure 7.1 shows the various process variables assigned to each node on the loop.

In chapter 3 we have discussed the control processes assigned to each node and the interconnections between them.

The following section gives a description of the message formats, based on the protocol designed for localized control systems (chapter 2). The messages discussed here are those concerned with the steady state operation of the high speed quantitative linear analysis.

7.1. Message Formats (steady state operation)

All messages constitute 3 parts; a header, text and a trailer. The common fields in all messages are shown below.

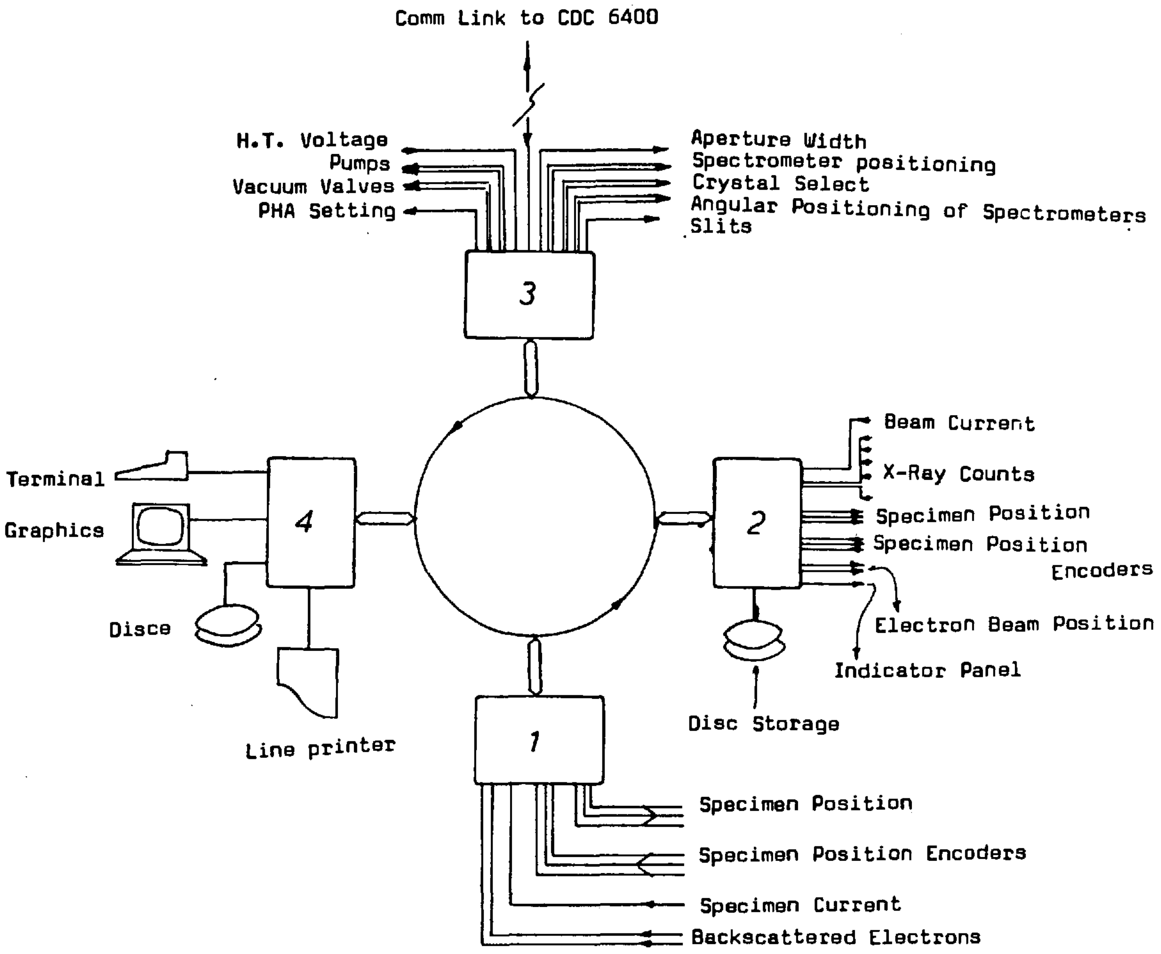


Fig. 7.1 Signals and devices on the 4-node computer network controlling the X-ray micro-analyser.

		Number of bytes
Header	SOH	1
	Mess. Type	1
	Length of Text	2
	Mess No.	1
	Source Code	1
	Destination Code	1
	Text	
Trailer	EOM	1
	Checksum	1
		9 bytes + Text

In section 3.3 a summary of the types of messages exchanged during the steady state operation were presented. The type and size of information carried in the text field of the different messages vary considerably as shown below.

(1) Control messages

a. - Messages exchanged between nodes 1 and 2 to increase or decrease λ and start from a boundary position: the 'text' format in this case will be:

	No. of bytes
Speed	1
X-position	3
Y-position	2
Z-position	2
	8 bytes.

b. - Messages from 1 to 2 for coordination in phase identification and recognizing topographical effects.

(2) Information messages

a. - Messages carrying phase information for remote filing, the text format will take the following form:

	No. of bytes
Type of phase	1
X-position	3
Y-position	2
	6

b. - Status information to the user (up to 20 bytes of text).

(3) User request messages (status enquiries)

Up to 20 bytes of text.

(4) Message carrying filing, line printer and display information

The text in these messages is allowed to achieve 1024 bytes.

(5) Alarm Messages

These messages carry orders to alarm processes informing them of names of components to be isolated, or actions (such as close-down) to be carried out. These messages are of high priority and hopefully occur infrequently.

7.2. Message Rates

1. Type (1) messages:

The messages exchanged between nodes 1 and 2 depend on the distances L1 and L2 (fig.3.1) and the speed of the specimen's movement. Two typical samples are given in Table 7.1, to show the relative average lengths of phase and matrix in various mineral compositions (7.1).

Table 7.1

Average phase lengths of mineral compositions in typical specimens

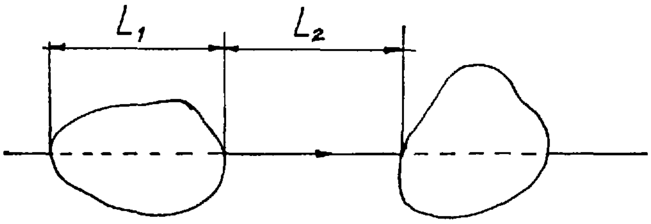
	Mineral name	% mineral in specimen		Mean phase length (μm)
Specimen 1	Pyrite	14.5		24.8
	Bornite	0.4		19.4
	Chalcopyrite	1.0		32.7
	Carrolite	1.0		22.8
	Chalcocite	0.6		27.2
	Mixed Silicates	9.1		27.4
		26.6%	mean =	25.7

Mean inter-phase length when % concentration is considered = 70.μm.

Specimen 2	Cassiterite	0.48		16.3
	Sulphide	8.75		26.4
	Silicates	19.80		33.0
		29.03%	mean =	25.2

Mean inter-phase length = 61.7 μm.

The message generation rates dependent on the specimen are calculated below for specimen 1.



Assuming the slow speed (across L1) is 100 micro-m/sec, and the high speed (across L2) is 1000 micro-m./sec.; the average time spent over a particle (L1) is 257 m.sec., and the average time spent between two particles (L2) is 70.3 m.sec.

Thus, each node would generate a control message (type 1) on the average every 257 + 70.3 = 327.3 m.sec, i.e. a message generation rate of 3.05 messages/sec.

2. Type (2) messages

Messages carrying phase information transmitted by node 2 could be generated at high rates if highly repetitive short phases are encountered. Message generation rates could achieve ranges of 500 to 700 messages/sec.

3. Types (3) and (4) messages

User request (status information) messages type (3) are broadcast from node 4 to other nodes at fairly slow rates. A rate of 2 - 3 requests/minute are justified by experienced machine operators (0.033 messages/sec).

Table 7.2 gives typical ranges of message rates for all types of messages classified by origin/destination pairs.

The message lengths are given in 'bits' and the message rates in 'message/sec'.

The loop is assumed to have synchronous transmission (19.2 k bit/sec) and 8 bits are added to each message for synchronisation.

Table 7.2

Average Message rates in the 4 node loop.

O/D	Message Type	Av.Message Length	Message rate (av).
1-2	1a	136	1-4
1-2	1b	136	1-700
1-4	2b	232	0.033
2-1	1a	136	1-4
2-3&4	2a	112	1-700
2-4	2b	232	0.033
3-4	4	1072	0.29 (in bursts)
3-4	2b	232	0.033
4-1	3	232	0.033
4-2	3	232	0.033
4-3	3	232	0.033
4-3	4	1072	0.29 (in bursts)

Alarm messages of type (5) are not included in Table 7.2 since their occurrence is scarce and highly improbable.

7.3. Performance of the four node loop using the ODM method

The mathematical model developed in Chapter 5 for 'optimum priority scheduling', is used in testing the delay behaviour of the 4-node network controlling the electron probe micro-analyser.

Traffic patterns based on those in table 7.2 were used and a channel speed of 19.2 k bits/sec. was assumed (this is the transmission speed of DEC's synchronous communications interface, the DUV-11).

It must be noted that the assumptions made in calculating the delay were based on Markovian arrivals. This assumption does not affect the validity of the superiority of the 'optimum priority scheduling' method (which is based on general arrival patterns) and in fact it is not far from reality due to the random nature of the messages generated.

Table 7.3 gives the average message delays on the loop for the 'ODM' and compares them to a case when priority is always assigned to the nodes (DLCN) and the case when the traffic on the loop has priority over the nodes traffic, at different network loadings. Figure 7.2 shows a graph comparing 'ODM', DLCN and 'worstcase'. The average delay is shown for low traffic loads (utilization, $\sum_i \lambda_i \bar{t}_i$), up to 0.3 at which the improvements of the 'ODM' over DLCN is 13%. At higher loads, however, improvements over DLCN and the worst case delays exceed 85% as shown in figure 7.3.

The reason for the great improvement in average delay can be explained as follows:

If short control and information messages are sent from node 2 to node 1 via nodes 3 and 4, and node 3 was sending long file messages to node 4, then:

- In the 'ODM' case, priority is dynamically assigned to the channel with shorter messages (based on sample averages). Therefore short messages will cut their way through the loop much more quickly than long messages.

In fact, the average delay experienced by short messages from origin to destination is much less than the total average delay if the 'O.D.M.' method as seen in figures 7.2 and 7.3 and in table 7.3.

Table 7.3.

Load or utilization	Average Delay (m.sec)		
$\sum_i \lambda_i \cdot \bar{x}_i$	Optimum priority scheduling 'ODM'	Priority given to line	Priority given to nodes (DLCN)
0.03	1.781	1.858	1.789
0.16	2.87	2.89	3.12
0.29	4.2	4.3	5.14
0.42	6.2	6.32	8.67
0.55	9.44	9.55	15.7
0.68	15.1	15.25	32.9
0.81	28.53	28.72	96.5
0.94	97.0	98.7	935.0

The 'ODM' method does not show a significant improvement in average delay over the case when the line has priority over the nodes, yet a considerable improvement in the delay performance over the 'DLCN' type loop occurs at very low loads.

However, the 'ODM' did not show improvement over the case of the 'line priority', fig.7.4, because of the line traffic characteristics dealt with in this particular case. The addition of other nodes creating short messages will immediately start showing the superiority of 'ODM', as seen in fig. 5.13 (chapter 5).

7.4. Performance of the 4-node loop using the ODM with priority classes

The messages defined in section 7.1 could be further divided into four categories of priority classes:

1. Alarm messages (type 5).
2. Control messages (types 1a and 1b)
3. Information messages (types 2a, 2b and 3)
4. Filing messages (type 4)

Table 7.4 shows the average message rates and lengths for each message type at each priority group. The expected origin destination (O/D) pairs are also specified. Alarm messages are excluded since their occurrence is unpredictable.

Table 7.4

Average Message rates for Priority Classes

Priority	Mess.type	O/D	$1/\mu$(av)	λ(av)
2	1a	1-2	136	1-4
		2-1	136	1-4
	1b	1-2	136	1-700
3	2a	2-3&4	112	1-700
	2b	1-4	232	0.033
		2-4	232	0.033
		3-4	232	0.033
	3	4-1,2&3	232	0.033
4	4	3-4	1072	0.29
		4-3	1072	0.29

Figure 7.5 shows the delay characteristics of the three major priority groups (excluding alarm messages). At high loads the average delay of control messages is in the order of 4 m.secs. and the information messages have an average delay of 7.5 m.secs. The values of the average delay given in this analysis are not exact due to the assumptions and the approximations used in the model. However the ratio between the average delays at each priority group should give a good indication of the delay behaviour; this ratio is found to be 5 : 8 : 36 at higher loads.

7.5 Conclusions

The implementation of the loop network for controlling the electron probe micro-analyser is based on four DEC-LSI-11s equipped with synchronous communications interfaces (DUV 11) for loop communications. The communication level protocol will be tested on the basis of the 'ODM' method for minimal delay. The higher levels of the protocol will be based on the principles set in chapter 2 (section 2.4). Priority groups will need structuring of a block in the memory so as to act as separate buffers accounting for each priority and for the node's traffic and loop's traffic. Practical experimentation to verify the 'ODM' method are being carried out on a 4 node network in the Department of Computing at Imperial College.

References

7.1 Jones, M.P. "Automatic Mineralogical Measurements in Mineral Processing". To be presented at the XIII International Mineral Processing Congress - Warszawa 1979.

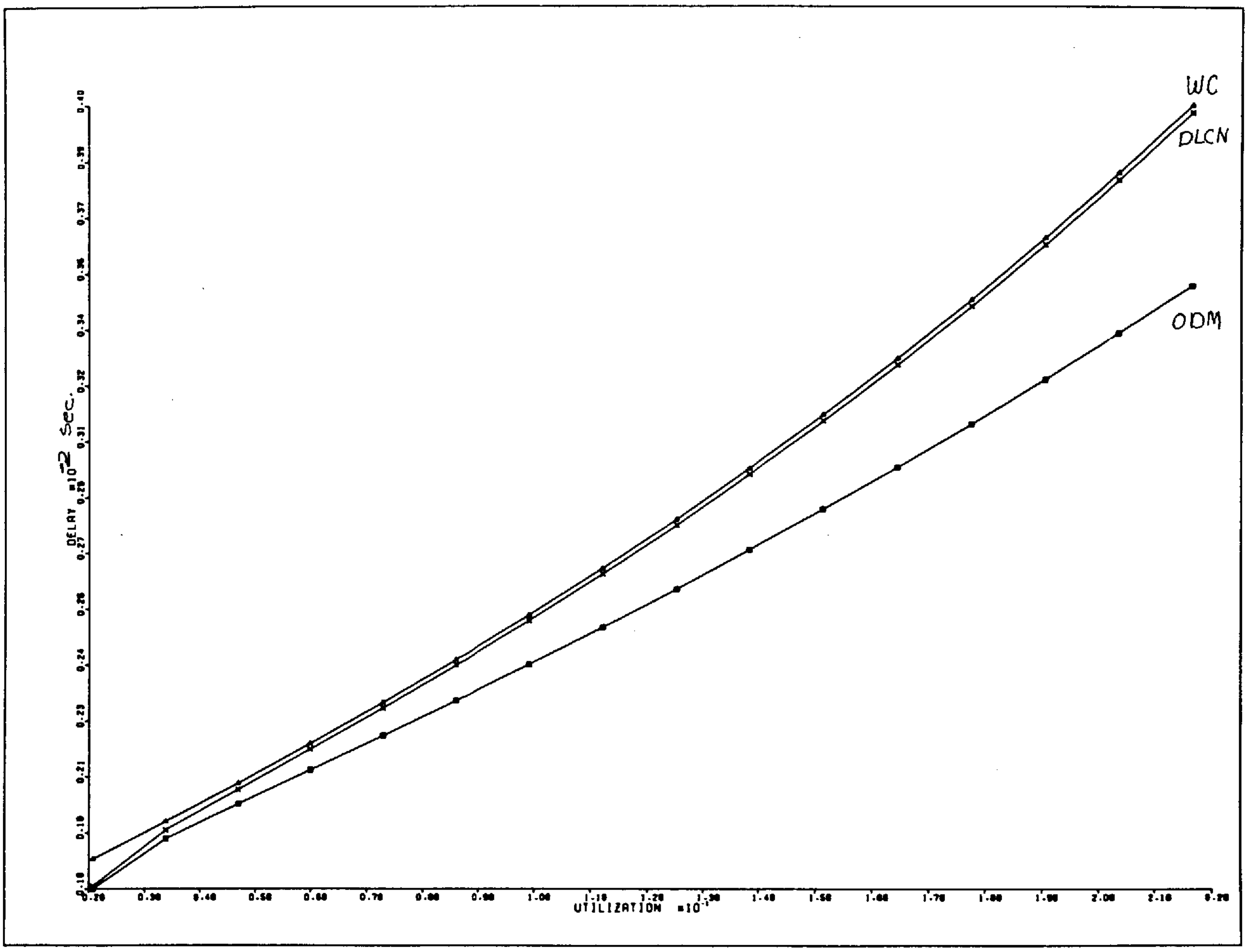


Fig. 7.2 Delay characteristics of the loop network controlling the X-ray micro analyser at low traffic loads.

c = 92 K bits/sec
N = 4
▵ WC
✗ DLCN
□ ODM

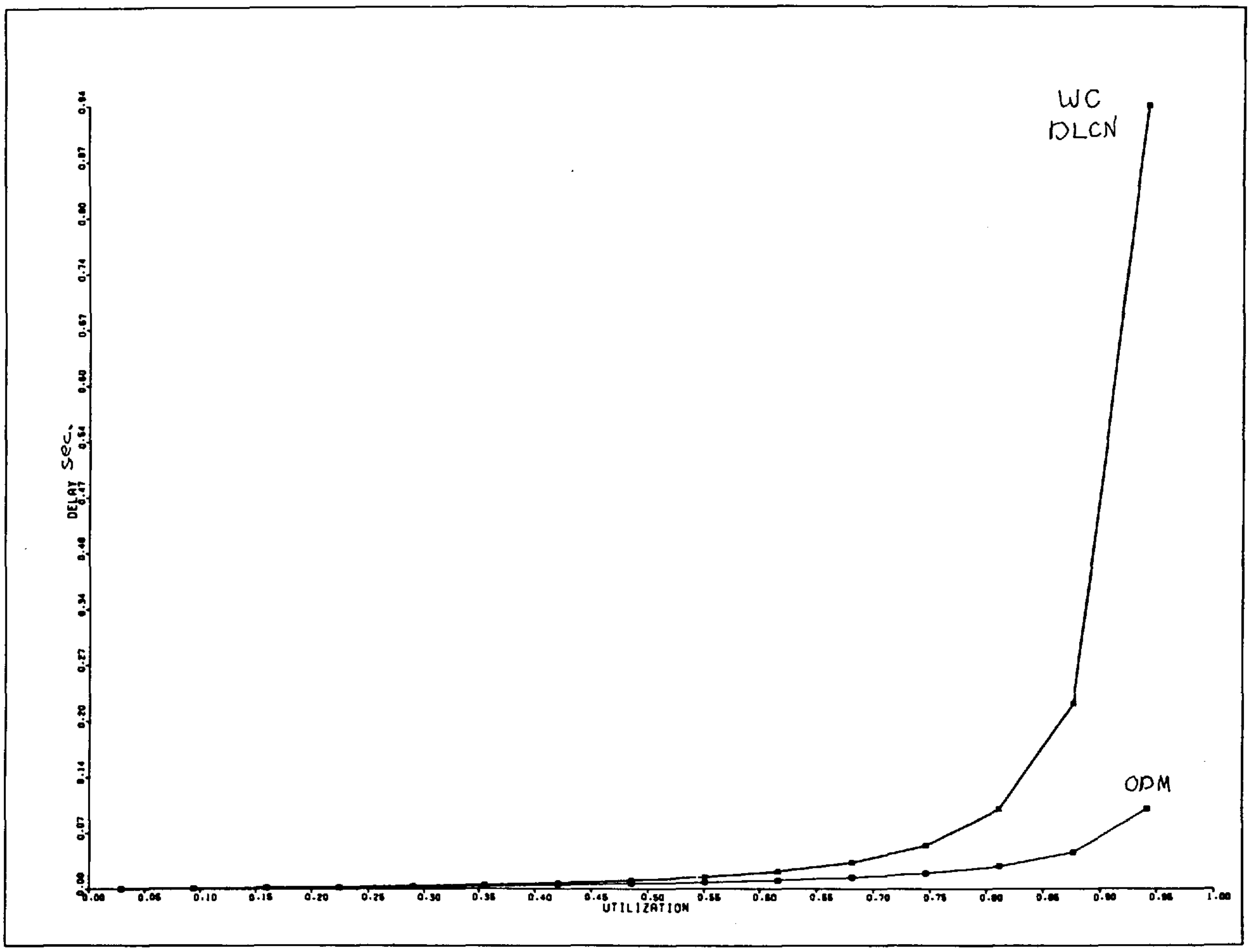
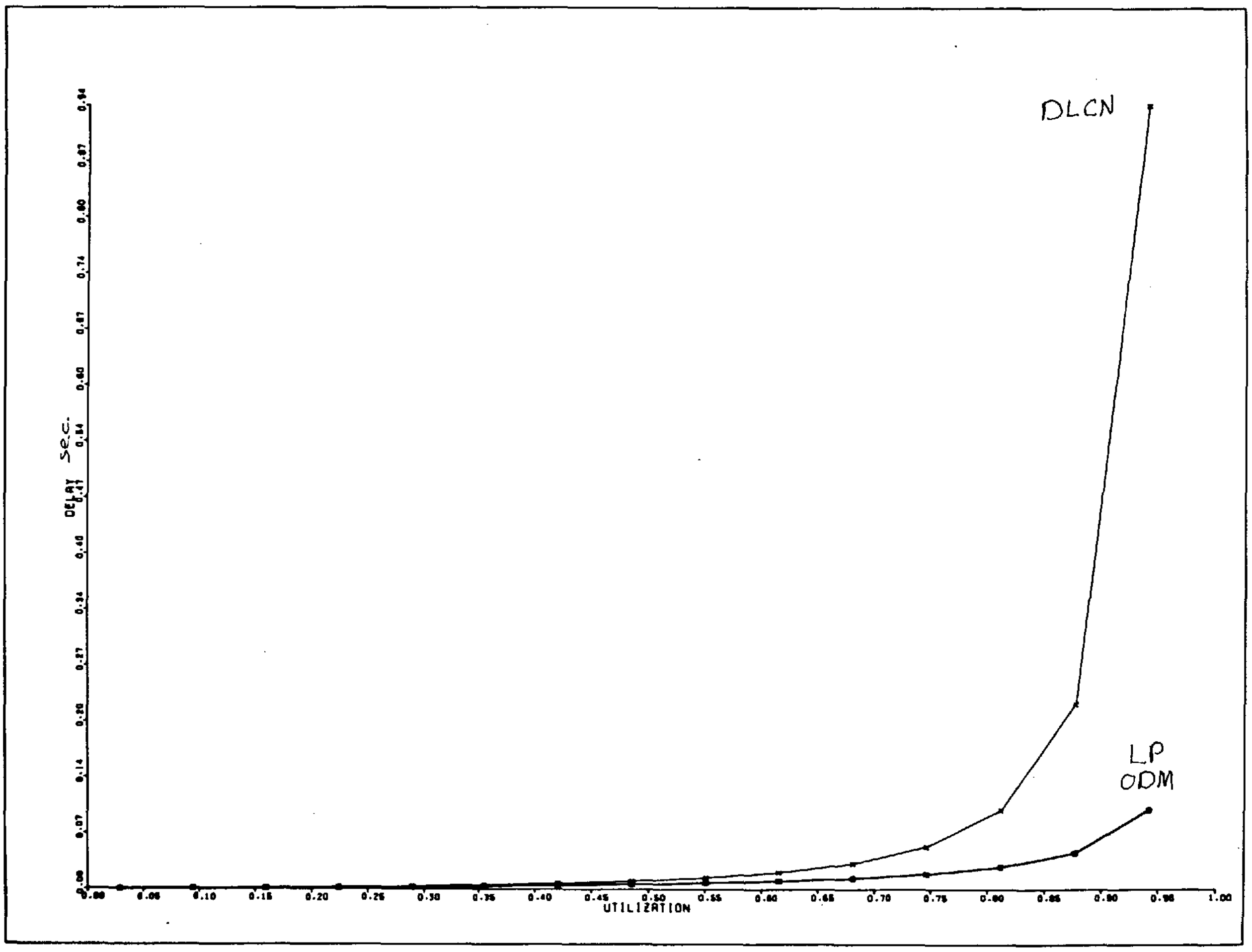


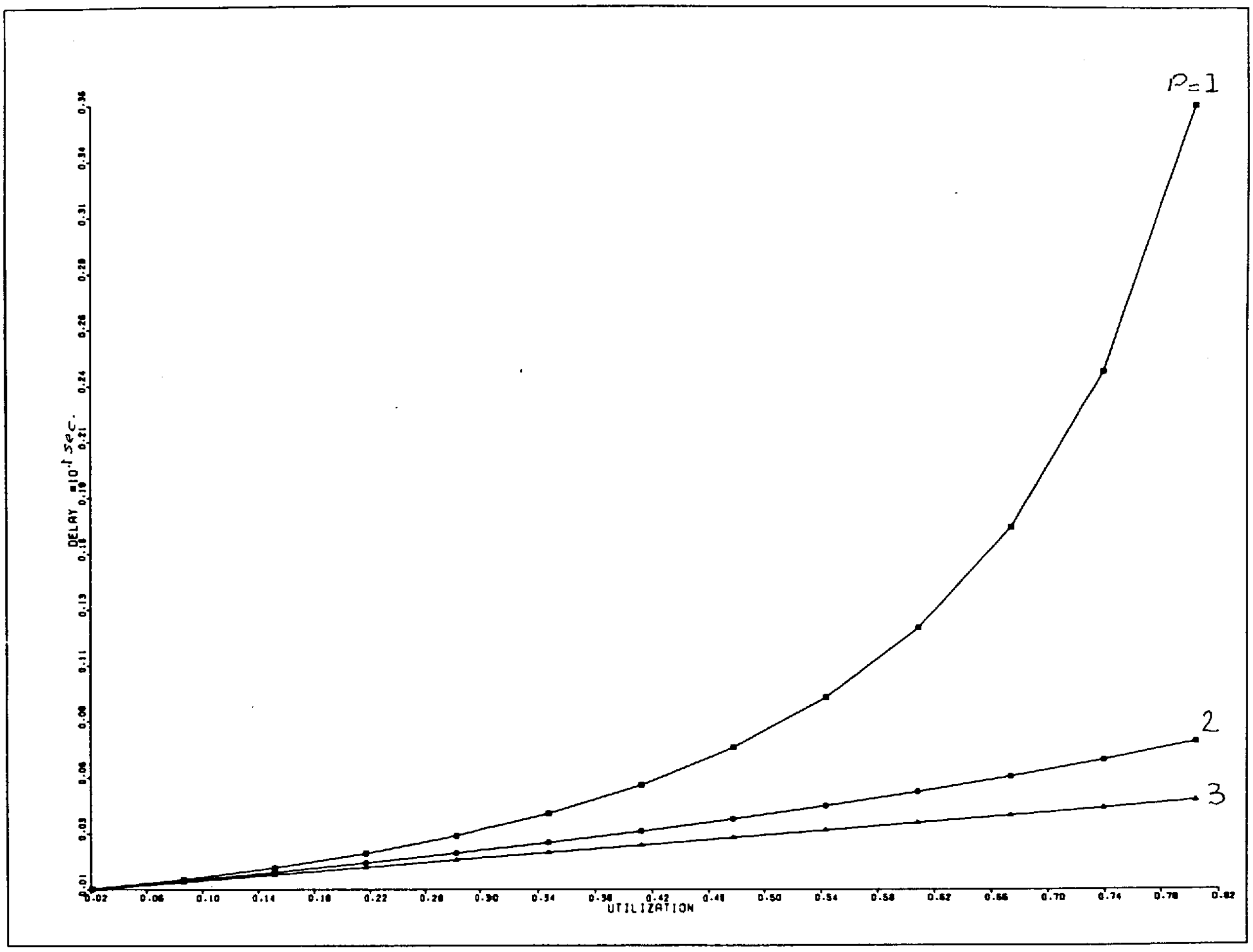
Fig. 7.3 Delay characteristics of the loop network controlling the X-ray micro analyser at high traffic loads.

c = 92 K bits/sec
N = 4
▲ WC
✕ DLCN
□ ODM
(WC and DLCN almost coincide



<u>Fig. 7.4</u> Delay characteristics of the loop control network comparing the ODM, DLCN and line priority.

X DLCN
Δ line priority
☐ ODM
(line priority and ODM almost coincide)



<u>Fig. 7.5</u> Delay characteristics of the loop control network comparing the three main priority classes of messages for the ODM technique.
(Highest priority exhibit least delay).