1

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

PRECISIONSM RESOLUTION TARGETS

PIONEERS IN METHYLENE BLUE TESTING SINCE 1974

MICROD

10800 COUNTY ROAD 8, BURNINGWELL, WIS 53507 USA
TEL: 612 426 1887 FAX: 612 426 1887 TLA 8100000000

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

The Effectiveness of a Contemporary Workstation Architecture in Servicing a Broadband Integrated Services Digital Network Workload

by

**James David Nicholas Myronyk
B.Comm., M.B.A.,**

**A thesis submitted to the Faculty of Graduate
Studies and Research in partial fulfilment
of the requirements of the degree of
Master of Science in Information and Systems Science.**

**Department of Systems and Computer Engineering
Faculty of Engineering
Carleton University
Ottawa, Ontario
September 1991**

**© Copyright
1991, James David Nicholas Myronyk.**

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-75995-X

Canada

The undersigned recommend to the Faculty of Graduate Studies
and Research acceptance of the thesis
**"THE EFFECTIVENESS OF A CONTEMPORARY WORKSTATION
ARCHITECTURE IN SERVICING A BROADBAND
INTEGRATED SERVICES DIGITAL NETWORK WORKLOAD"**

submitted by
James David Nicholas Myronyk, B.Comm., M.B.A.,
in partial fulfilment of the requirements for
the degree of Master of Science in Information and Systems Science.

Thesis Supervisor

Chair, Department of Systems and Computer Engineering

Carleton University
December 1991.

ABSTRACT

A Broadband Integrated Services Digital Network (BISDN) is expected to operate at data rates of 150 to 600 Mbps. This thesis investigates the effectiveness of a personal computer, or workstation, with a conventional or contemporary architecture to service a BISDN workload at these data rates. The thesis goal is to demonstrate that the architecture of today's contemporary personal computer architecture is not suitable for the real time requirements of future BISDN video and voice applications.

Due to the characteristics of the multiple service centre, multiple queue workstation model and the multiple job class, multiple priority, multiple job routing workload model, theoretical analysis was an unsuitable solution method. Consequently a ten thousand line SimscriptTM simulation program was developed to test the hypothesis.

Results were obtained from computer runs simulating eight hours of workload processing. The results indicated that, as expected, the system bus was the performance bottleneck. Additionally, the performance of time sensitive jobs — for example, voice and low resolution video jobs — were shown to deteriorate beyond practical limits at sub-BISDN data rates.

ACKNOWLEDGMENTS

I thank Dr. D.C. Coll for his supervision, suggestions, discussions and general assistance; also for his patience, genuine interest and concern.

I acknowledge financial support received from the Department of Systems and Computer Engineering and TRIO. I also acknowledge and thank P and C, plus M for their financial loans.

To my friends, both new and old, thanks for your words of encouragement and support. To J, who “enjoyed” a parallel experience, thanks for your friendship and for “adopting a student”. To T, a warm hug and deep thanks for your support, encouragement, friendship and understanding.

Lastly, to my family, a special thanks for your support, kindness and caring during this personally rewarding, albeit stressful, time.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	iv
LIST OF FIGURES	vii
1 INTRODUCTION	1
2 BACKGROUND	3
2.1 Computer System Architecture	3
2.2 Computer Communications	12
2.2.1 BISDN Services	13
2.2.1.1 Interactive Services	13
2.2.1.2 Distributive Services	14
2.2.2 BISDN Architecture	15
2.2.2.1 SONET	15
2.2.2.2 Asynchronous Transfer Mode	19
3 WORKSTATION AND WORKLOAD MODELS	25
3.1 The Workstation Model	25
3.2 Workload Model	32
3.2.1 Facsimile Job Class	32
3.2.2 Data Transfer Job Class	36
3.2.3 Interactive Job Class	39
3.2.4 Voice Job Class	42
3.2.5 Low Resolution Video Job Class	46
3.2.6 High Resolution Video Job Class	49
3.3 Resource Parameters	52
4 APPROACHES TO COMPUTER SYSTEM MODEL PERFORMANCE EVALUATION	55
4.1 Historical Perspective	57
4.2 Analytical Modelling	62
4.2.1 Basic Concepts	62
4.2.2 Queuing Networks	65
4.2.2.1 Separable Queuing Networks	65

4.3	Analytical Evaluation of Queuing Networks	66
4.3.1	Queuing Network Model Inputs	66
4.3.2	Queuing Network Model Outputs	67
4.3.3	Bounding Analysis	68
4.3.4	Estimation Techniques	69
4.3.4.1	Multiple Job Class Models	69
4.4	Simulation Modelling	75
4.4.1	Model Implementation	76
4.5	Performance Targets	77
4.5.1	Workload Requirements	78
4.5.1.1	High Resolution Video	81
4.5.1.2	Low Resolution Video	81
4.5.1.3	Voice	82
4.5.1.4	Interactive	83
4.5.1.5	Data	84
4.5.1.6	Facsimile	85
4.5.2	Workstation Requirements	85
4.5.2.1	Time	85
4.5.2.2	Space	87
5	METHODOLOGY	94
6	RESULTS	96
7	DISCUSSION	110
8	CONCLUSIONS AND FUTURE RESEARCH . .	113
	BIBLIOGRAPHY	115

LIST OF FIGURES

1	Instruction and data streams in a simple computer.	4
2	General computer system structure with $n \geq 1$ processors and $m \geq 0$ memory units.	5
3	Expanded structure of the Princeton Institute for Advanced Studies (IAS) Computer.	6
4	Hardware architecture of the "Personal Multimedia Teleconferencing Terminal"	8
5	Trade-off between frame rate, frame size and compression rate over a 64 kbps ISDN transmission facility.	9
6	Komiya, et al, multifunction graphic terminal for the ISDN environment	10
7	Heathrow fiber access system	11
8	Block diagram of BISDN User-Network Interface .	12
9	The STS-1 Synchronous Payload Envelope	16
10	The STS-N frame structure	17
11	SONET digital interface rates	18
12	ATM cell structure.	20
13	ATM within SONET (150.520 Mbits/s)	21
14	ATM adaptation layer model.	22
15	ATM adaptation service classes.	23
16	Job flow through a service centre.	26
17	A computer system modelled as a network of service centres.	27
18	An interim resource model with an interrupt queue.	29
19	Resource model with separate queues for waiting and interrupted jobs for each job class priority.	30
20	The workstation modelled as a network of queues.	31
21	Facsimile job parameters.	33
22	Facsimile job's routing probabilities.	34
23	Conditional probabilities for the routing of facsimile jobs.	35
24	Resource visitation probabilities for a facsimile job.	36

25	Data transfer parameters.	37
26	Data job routing parameters.	38
27	Conditional probabilities for data jobs.	38
28	Resource routing probabilities for a data job.	39
29	Interactive job parameters.	40
30	Interactive job routing probabilities.	41
31	An interactive job's conditional probability routing.	41
32	Resource visitation probabilities for an interactive job.	42
33	Voice job parameters.	44
34	Voice job routing probabilities.	44
35	Conditional probability of voice route selection.	45
36	Resource visitation probabilities associated with a voice job.	46
37	Low resolution video job parameters and routing probabilities.	47
38	Low resolution video job routing probabilities.	48
39	Resource visitation probabilities for low resolution video jobs.	49
40	High resolution job parameters and routing information	50
41	Conditional probability low resolution video job routing.	51
42	Resource visitation probabilities for high resolution video jobs.	52
43	A single server queue system.	58
44	The machine repair model of a time sharing system. Source Fig.1 [LAVE89].	58
45	Central server model of a multiprogramming system. Source: Fig.2 [LAVE89].	59
46	System arrivals and completions	63
47	Fundamental Laws.	64
48	Multiple class model outputs	68
49	One version of BISDN service requirements	79
50	A second version of BISDN traffic	80

51	The effect of a workload increase on user response time.	86
52	Service centres visited by each job class.	88
53	Job size and activation rate.	89
54	Calculation of minimum buffer space requirements for the bus service centre	89
55	Buffer space requirements for system resources .	91
56	Maximum cycle times.	99
57	Average cycle times.	100
58	Average cycle and wait times for a facsimile job.	101
59	Average cycle and wait times for a data job. . .	102
60	Average cycle and wait times for an interactive job.	103
61	Average cycle and wait times for a voice job. . .	104
62	Average cycle and wait times for a low resolution video job.	105
63	Average bus server utilization	106
64	Average and maximum number of jobs waiting in the bus interrupt queue	107
65	Average and maximum number of jobs waiting in the bus wait queue	108

CHAPTER 1

INTRODUCTION

The thesis examines the effectiveness of a personal computer, or workstation, with a conventional architecture to service a Broadband Integrated Services Digital Network (BISDN) workload. The thesis goal is to demonstrate that the architecture of today's contemporary personal computer architecture is not suitable for the real time requirements of future BISDN video and voice applications.

This work is theoretically significant because it identifies that a change in the strategic direction of workstation design may be necessary. This change may be necessary to support the real-time aspects of some envisioned BISDN applications. In a broader context, it implies that if a broadband high resolution video job (for example, a multimedia application) is a crucial ingredient in the widespread use of BISDN services, then the rate of BISDN acceptance and utilization could be adversely affected.

There are eight chapters in this document. Chapter one is this *INTRODUCTION*. It states the thesis goal and significance; additionally this chapter identifies and provides an overview of the remaining chapters.

Chapter two, *BACKGROUND*, describes a conventional or contemporary computer architecture and provides information concerning multimedia workstation research. This chapter also describes BISDN communication services, and SONET and ATM protocols.

Chapter three, *WORKSTATION and WORKLOAD MODELS*, describes a multipurpose, integrated services, BISDN personal workstation and a multiple class BISDN workload. The developed workstation model represents a computer system as a network of queues. Each system resource is modelled as a service centre with an interrupt and a waiting queue for each of the six job classes. The job classes have fixed priorities. The priorities, from worst or lowest, to best or highest, are: data, interactive, voice, low resolution video and high resolution. Jobs are serviced by a resource based on their priority, and their possible status in either the waiting or interrupt queues.

Chapter four, *APPROACHES to COMPUTER SYSTEM PERFORMANCE*, provides a theoretical discussion of analytical modelling and evaluation techniques. It describes why these techniques are ill-suited to the evaluation of the

developed models and explains why simulation is well-suited for this evaluation. This chapter concludes with a development of workstation and workload requirements.

Chapter five, *METHODOLOGY*, provides sufficient information concerning the experimental approach and SimscriptTM run time parameters to permit a motivated researcher to repeat the experiment and conclude with the same, or nearly identical, results.

Chapter six, *RESULTS*, reports on the performance measurements obtained from approximately twenty-five simulation runs. The results are presented as plots of average and maximum job cycle times, job wait times, and bus server, interrupt and wait times. Only the bus is reported as it is the apparent system bottleneck.

Chapter seven, *DISCUSSION*, observes that the level of model detail may have provided marginal returns — that a simpler model and simulation program may have resulted in similar findings. It also estimates that workstation technology would need to increase by approximately 20 to 25 times its current capacity to allow the high resolution video job class to be effectively serviced on modelled workstation. It lastly introduces the idea that a multiple, parallel bus architecture could be more effective and compatible with the effective servicing of integrated, broad bandwidth BISDN applications.

Chapter eight, *CONCLUSIONS and FUTURE CONSIDERATIONS*, restates the conclusion that the results of the simulation evaluation of the developed workstation and workload models indicate that the thesis goal was obtained. Namely, to demonstrate that a personal workstation with a contemporary architecture would not effectively service a BISDN workload — especially a workload with a high resolution video component. Future work could be directed at different components of the BISDN workstation and workload puzzle — alternate analysis techniques to collaborate these findings, parallel architecture modelling, BISDN application and operating system specification and development, and eventually, developing a BISDN workstation prototype.

CHAPTER 2

BACKGROUND

This chapter provides background information relevant to explaining the thesis topic environment from the perspectives of *COMPUTER SYSTEMS ARCHITECTURE* and *COMPUTER COMMUNICATIONS*.

Section one, *COMPUTER SYSTEMS ARCHITECTURE*, describes the conventional computer architecture developed in the mid 1940s and identifies its influence on today's computer systems. Section two, *COMPUTER COMMUNICATIONS*, describes the BISDN environment and identifies requirements of some broadband services that may be ineffectively addressed by a conventional personal workstation.

Stallings states that the rapid rate of social and economic change experienced by the more-developed nations in the post-industrial era was fuelled by technological change. He further identifies *computer and communication* systems as the two key fundamental ingredients driving this technological change. He observes that computer and communication systems have developed traditionally along separate paths. Their future development, however may be about to merge with the introduction of Integrated Services Digital Network (ISDN) systems, and by inference, BISDN [35].

Section 1 Computer System Architecture

The majority of computers utilize key organizational and behavioral concepts that remain from the design of first generation computers dating back to the mid 1940s. In itself, this is not necessarily a good or bad situation. Instead each situation must be examined on its own merit to determine if original design concepts remain sound and responsive to users' requirements into the future. For example, the world continues to wear a well beaten path to the standard spring-loaded mouse trap. On the other hand, there is also evidence of originally sound ideas falling by the wayside as they fall out of step with users' current requirements. New black and white televisions, or engineering slide-rules are conspicuous — to turn a phrase — by their presence. Determining if a conventional computer system belongs to the former or later category requires a clear understanding of what is meant by a computer architecture.

Hayes describes “computer architecture” as the structure and behaviour of a computer system. A computer architecture can be characterized by the number of instructions that can be processed simultaneously, the internal organization of the processors, the interprocessor connection structure, and the methods used to control the flow of data and instructions through the system [13].

A popular classification architecture classification system was developed by Flynn [13]. This system classified computer systems according to the number of their simultaneous instruction and data streams. The steps associated with transferring instructions from memory defined the *instruction stream* while the steps concerned with transferring operands to and from memory defined the *data stream*. Flynn further defined a typical instruction cycle as having the following major steps:

1. generate the next instruction address;
2. fetch the instruction;
3. decode the instruction;
4. generate the operand address;
5. fetch the operands;
6. execute the instruction;
7. store the results.

This arrangement is depicted in Figure 1.

Figure 1 Instruction and data streams in a simple computer.

Flynn’s system allowed for four categories of computer systems: single instruction stream, single data stream (SISD); single instruction stream, multiple data stream (SIMD); multiple instruction stream, single data stream (MISD); and multiple instruction stream, multiple data stream (MIMD). At one end of this classification spectrum is a sequential computer — a SISD system — while at the other end of this spectrum is a parallel computer — a MIMD system. While

there exists many computer systems of each classification type, the most common general purpose systems are SISD computers. The SISD classification is also referred to as an IAS computer or a von Neumann computer. The IAS computer is the prototype of all subsequent general purpose computers [35].

The Flynn classification has been criticized for being behavioural in nature and ignoring overall computer structure or interconnect topology, especially in the context of parallel systems. Hayes proposes a classification system that addresses these criticisms [13]

Hayes views a computer system as a set of $n \geq 1$ processors $P_1, P_2, \dots, P_n$ and $m \geq 0$ shared (main) memory units or modules $M_1, M_2, \dots, M_m$ communicating through an interconnect network N . For simplicity, IO processors or peripheral devices are ignored from the classification viewpoint. This general structure is shown in Figure 2.

Figure 2 General computer system structure with $n \geq 1$ processors and $m \geq 0$ memory units.

In the traditional case, the number of processors is one, the number of memories is one and the interconnection network is a single shared system bus over which all processor-memory communications occur. An example of such a system is the von Neumann or IAS computer shown in Figure 3.

Figure 3 Expanded structure of the Princeton Institute for Advanced Studies (IAS) Computer.

While the Flynn classification scheme is a quarter of a century old, the von Neumann reference is associated with the latter's role in the development of the first generation computer EDVAC, in 1945, and Princeton's Institute for Advanced Studies (IAS) computer, in 1946. One major characteristic of the IAS computer was its sequential architecture — a single central processor interfaced to memory and input/output equipment. Subsequent generations of computer

systems expanded on this notion and eventually off-loaded processing and control functions to subsystem controllers. With few exceptions however, the underlying sequential nature of task processing remains unchanged even today, as does the utilization of a single system bus for processor to subsystem communication. Most computer systems, including nearly all personal computers, can be described as having an IAS architecture.

One of the reasons why the fundamental IAS or von Neumann design has been able to survive through the decades has been its ability to offset the demands of new requirements with technological advances. This apparent strategic reliance on “faster, bigger, cheaper” computer components has been reinforced by the computer industries pattern of doubling system capabilities every two to two and one half years [13].

However in spite of this rate of change, the basic architecture is sometimes found wanting. For example, pipeline and multiprocessor parallel architectures were developed in order to process applications having inherently non-sequential and/or real-time characteristics. In addition, third party vendors have developed and marketed add-on subsystem components in order to provide a new capability or boost performance. This behaviour is especially commonplace in the personal computer marketplace.

There is not a lot of information available regarding BISDN workstations. The available material tends to approach this matter with a perspective usually easily identifiable with specific industries or research areas. These approaches will be referred to as communication-centric, computer-centric and application-centric.

One vision of a future workstation design is presented by Nakamura, et al [26]. They describe a prototypical personal multimedia teleconferencing terminal that integrates voice, video, bi-level graphics, color still pictures, handwritten images and electronic pointers. As Nakamura et al are employed by Hitachi Research Laboratory, Hitachi Ltd., their approach will be classified as computer-centric. This approach, illustrated as Figure 4, involves augmenting a standard personal computer with special purpose hardware and software in order to deliver and manage a specific integrated application.

This approach typifies the computer industry's approach to addressing ISDN application requirements by modifying an existing pre-ISDN approach. Specifically, the researchers applied software and hardware modifications to an IAS, or conventional, personal computer to execute an ISDN application. This is a viable and economical approach provided the workstation can effectively service the application's requirements.

However to demonstrate the workstation could service the multimedia teleconferencing workload, the application workload's characteristics were interpreted to reflect the maximum deliverable capability of the workstation and network environment, rather than the application's true service requirements.

Figure 5 Trade-off between frame rate, frame size and compression rate over a 64 kbps ISDN transmission facility.

This "augmented-PC" approach, that is common from computer researchers, is also championed by Komiya et al [18] in their description of a multifunctional graphic terminal. It is illustrated as Figure 6.

Figure 6 Komiya, et al, multifunction graphic terminal for the ISDN environment

In contrast, BNR's Heathrow (Florida) experiment with providing fiber to the door, is representative of a communications-centric vision of the ISDN workstation [5]. From this perspective, the workstation is really seen as multiple user interfaces individually connecting to the network via a household or neighbourhood node. The Heathrow situation is shown in Figure 7.

Figure 7 Heathrow fiber access system

Application-centric researchers have been silent regarding their perceptions of workstation requirements for ISDN applications. Implicitly it appears that they have assumed that the all pervasive conventional personal computer architecture will be the chosen instrument for application delivery. This is a rational approach that has been repeatedly verified by leapfrog technological advances that have made successively more powerful applications generally available. Witness the growth in power of engineering workstations and graphical user interfaces within the previous half decade.

Stallings' [36] perspective is representative of a communication-centric vision; it is depicted in Figure 8. In this view, BISDN services are delivered to a customer's premise where they are managed by multiple, largely autonomous, end-user equipment. Implicitly, the user has more than one workstation to operate and is responsible for integrating the various information sources.

Figure 8 Block diagram of BISDN User-Network Interface

Stallings' communication-centric view is supported by Alcatel's Treves whose work considered imaging requirements in tele-medicine, tele-education, electronic publishing, and industrial design and manufacturing applications. Treves describes the typical broadband terminal as a personal computer augmented with data/image devices (e.g. keyboard, scanner, TV camera, mouse, light-pen), image output devices (e.g. laser/graphics printers, video display units) and data/picture storage devices (e.g. optical disks) [38].

Section 2 Computer Communications

The CCITT has described BISDN as "a service requiring transmission channels capable of supporting rates greater than the (ISDN) primary rate" [36]. This is analogous to saying: 'IBM makes computers' and 'the Titanic had a difficult voyage'. Though accurate statements, metaphorically all three barely touch the tip of the iceberg.

In contrast to ISDN's 64 Kbps basic and 1.544 Mbps primary rates, BISDN is anticipated to provide approximately 150 to 600 Mbps bandwidth [36]. Those

additional magnitudes of bandwidth are being targeted towards providing video services that are currently infeasible to transmit using ISDN facilities. For example, imaging, entertainment video or broadband multimedia information retrieval services.

BISDN Services

The CCITT classified potential BISDN services as interactive or distributive. The former is characterized by a bidirectional flow of information between subscribers or between a subscriber and a service provider. Conversational, messaging and retrieval services are subclassifications of interactive services.

In contrast, distributive services are primarily characterized by a unidirectional information flow from a service provider to a BISDN subscriber. This flow may represent broadcast or cyclical services.

Interactive Services

Conversational This service provides bidirectional, real-time end-to-end information transfer between two users or between a user and a service provider host. The exchange of this information is conceptually within a private connection and not available to the general public.

Conversational interactive services can be further organized into video, data and document categories. Video telephony and video conferencing are examples of the video category. Video telephony refers to the inclusion of a video transmit and receive capability in the end user's telephone equipment. The conversing parties could then choose to have a virtual face-to-face conversation, or to display an object, for example a chart or the latest *Calvin and Hobbes*® cartoon, to augment or animate the information content of the conversation.

Video conferencing refers to a typical (board) meeting where the participants are geographically separated and require the ability to exchange, in real time, information visually. The service represents a shared virtual conference room — conference participants see each other and interact in a way analogous to being physically located in the same room.

Data conversational services refers to high speed data transfer, such as inter-LAN or computer-to-computer connections, high volume file transfers and CAD/CAM program transfers. Document conversational services refers to very high resolution facsimiles, and multimedia data transfers.

Messaging Messaging refers to inter-user communication of electronic mail (email). The basic service is similar to typical email programs, like the SunOSTM offering *mail*. This program provides facilities to write, send and receive text messages. Broadband versions of this service could include video or mixed media (e.g. voice, graphics, etc.) components to augment the textual message.

Retrieval Retrieval services provide the user with the capability to retrieve information from public databases. Videotex applications would be an example of this service in a narrowband environment. Broadband videotex would enhance standard videotex offerings of text and simple graphics with sound, video and high resolution images. A second version of this service type could be for renting movies from a "virtual" video rental outlet or library.

Distributive Services Stallings identifies two types of distributive services as broadcast and cyclical [36]. The former permits a subscriber to tap into a flow of information without being able to control the content or scheduling of the information's delivery. The latter permits user control over the content and scheduling of an reoccurring cycle of information. These services are now briefly described.

Broadcast A broadcast service provides a continuous flow of information that is distributed from a central source to an unlimited number of authorized users. Each user can access the information flow but the central source establishes the transmission time and sequence. An example application of this type of service is high definition television (HDTV).

A second example of this type of service would be the broadcast of an electronic newspaper [36]. This facility can be imagined to be a transmission of facsimile images of newspaper pages to subscribers of the service.

Cyclical A cyclical service is a type of information distribution characterized by a cyclical repetition of information frames. An example of a BISDN cyclical services application is cabletext [36]. Cabletext could provide users with frequently requested information having text, audio, video and image components. This service could be installed in a large hotel as a guest-controlled virtual concierge.

In comparison with a similar ISDN application supporting a few hundred pages and a few tens of seconds cycle time, a BISDN cyclical service could support 10,000 pages.

BISDN Architecture

The bandwidth requirements of high resolution video, approximately 150 Mbps, will drive the overall BISDN data rate requirement [36]. The simultaneous support of multiple interactive and distributive services may require a 600 Mbps data rate. The only appropriate technology for widespread support of such data rates is optical fibre [36]. The CCITT has identified Synchronous Optical Network (SONET) as the optical network physical interface standard.

In contrast to the high bandwidth requirements of high resolution video services, other BISDN services may have narrowband requirements. For example, some file transfers, "plain-old-telephone-service" (POTS), facsimile transfers and videotelephony transmissions have default requirements of 64Kbps.

There is also wide variability in the degree of burstiness — the ratio between the time that data is sent versus the time that the channel is occupied — for the various communications services. For example messaging and retrieval services are more bursty than distributive video services.

These two factors, bandwidth and burstiness variability, lead to the development of a fast packet switching technique, Asynchronous Transfer Mode (ATM), as the BISDN user payload interface.

SONET The SONET approach was originally proposed by Bellcore (Bell Communications Research) as the standard BISDN optical interface.

The SONET basic building block is the synchronous transport signal level 1 (STS-1) frame. The STS-1 frame has a bit rate of 51.84 Mbps and is repeated every 125 μ seconds. This 125 μ second period supports digital voice signal transport at 64 kbps.

The STS-1 frame structure is shown in Figure 9. It represents nine rows and ninety columns (9 x 90 octets or bytes), and is transmitted row by row, from left to right. The STS-1 frame is divided into the transport overhead and the Synchronous Payload Envelope (SPE). The transport overhead carries overhead information while the SPE carries the SONET payload. Within the SPE there is one column (nine bytes) of path overhead the path corresponds to a logical connection between source and destination.

Figure 9 The STS-1 Synchronous Payload Envelope [36].

The transport overhead carries information necessary for secure transmission of the SPE. It consists of nine rows by three columns (twenty seven bytes) and carries overhead bits for connections at the section level (connection between regenerators) and connections at the line level (connection between light-wave terminating equipment). The section overhead is processed at each regenerator. The line overhead is passed transparently through the regenerators and is processed by light-wave terminating equipment.

The SPE carries the SONET payload. The payload allocates one column for path overhead and twenty-six columns for the service or signal being transported. The path overhead is passed transparently from the point where the STS-1 payload

is composed to the point where it is decomposed. The important functions of the path overhead are:

1. end-to-end payload error monitoring;
2. identification of the type of payload being carried;
3. path status indication;
4. a trace function which allows a user to trace a signal through the network as it goes through different elements.

Higher rate SONET signals (STS-N) are obtained through the synchronous byte multiplexing of STS-1 frames. The STS-N frame structure is illustrated in Figure 10.

Figure 10 The STS-N frame structure [36].

This frame structure consists of nine rows and N times ninety columns. The transport overhead consists of N x three columns, and there are N x one columns of

path overhead. The aggregated bit rate of an STS-N signal is exactly N times the basic rate of 51.84 Mbits per second. Thus an STS-3 carries three byte-interleaved STS-1 signals in an aggregated bit rate of 155.52 Mbits per second. Currently the only values of N allowed are 1, 3, 9, 12, 18, 24, 36 and 48. The optical signals can be obtained by passing the electrical STS-N signal through an electro-optic conversion. The optical signal of STS-N is called an Optical Carrier Level N (OC-N). The OC-N will have a line rate exactly the same as the corresponding STS-N. The digital interface rates for SONET are given in Figure 11.

Figure 11 SONET digital interface rates

OC LEVEL	LINE RATE
OC-1	51.840
OC-3	155.520
OC-9	466.560
OC-12	622.080
OC-18	933.120
OC-24	1244.160
OC-36	1866.240
OC-48	2488.320

If all of the STS-1 signals in the STS-N have the same destination, then they can be concatenated. This concatenated STS-N signal is referred to as an STS-Nc signal. In this format, the payload is treated as a single unit, and thus only one column of path overhead is needed. This is not the case of the unconcatenated STS-N signals. However N copies of the section and line overheads are still required in the STS-Nc signal since any intervening transmission equipment expects to see the individual section and line overhead.

A key feature of SONET is the virtual tributary (VT) payload structure. VTs operate as separate containers within the STS-1 frame structure and are used to carry a variety of lower rate signals such as DS1, DS1C and DS2 within an STS-1. To efficiently accommodate the North American and European digital hierarchy,

these containers come in four sizes: VT1.5, VT2, VT3, and VT6. A VT1.5 can be used to carry a North American DS1 signal (1.544 Mbits per second), a VT2 for a European CEPT-1 signal (2.048 Mbits/s), a VT3 for a DS1C signal (3.152 Mbits/s), and a VT6 for a DS2 signal (6.312 Mbits/s). A DS3 signal (44.736 Mbits/s) is carried in an STS-1 SPE.

SONET uses payload pointers to allow easy access to the payload. As previously noted, the STS-1 frame is divided into the transport overhead and STS-1 SPE. The payload is easily accessed since the STS-1 payload pointer, contained in the transport overhead, indicates the starting byte location of the STS-1 SPE within the STS-1 frame. The STS-1 payload pointer also avoids the need for the slip buffers and eliminates associated payload corruption and delay. In conventional methods, such as fixed location mapping, slip buffers are needed at the multiplexing equipment interfaces to take care of frequency differences by either repeating or deleting a frame of information. This increases delay and may cause signal impairment due to slipping. The STS-1 payload pointer avoids the need for slip buffers since small frequency variations of the STS-1 payload can be accommodated by adjusting the pointer value.

SONET can also have VT pointers. The VT pointers follow the same principle as the STS-1 payload pointers, except at the VT level. The VT pointer indicates the position of the starting byte of the VT SPE within the VT payload structure. The pointer adjustment rules are analogous to that of the STS-1 pointer, and the VT pointer has the same advantages as the STS-1 pointer, the VT pointer has the same advantages as the STS-1 pointer, (e.g. dynamic alignment between the STS-1 SPE and the VT SPE, minimal delay).

Asynchronous Transfer Mode CCITT Recommendation 1.121 identifies ATM as the final transfer mode for BISDN. ATM is a fast packet switching technique based on virtual connections using fixed-size packets called cells. An ATM cell stream consists of a continuous sequence of cells, either designated as being empty or dynamically allocated to variable or fixed rate data sources.

The cells consist of a header and information field. Cells routing is performed based on the Virtual Circuit Identifier (VCI) contained in the cell header. The cell transmission time is equal to a slot length, and slots are allocated in an asynchronous, demand-based manner.

Figure 12 ATM cell structure.

Header Fields::

- generic flow control field (4 bits)
- virtual path field (8 bits)
- virtual channel field (16 bits)
- payload type (2 bits)
- loss priority field (1 bit)
- reserved (1 bit)
- header check sequence (8 bits)

ATM is efficient and flexible [36]. Its slots are allocated to services on demand and consequently ATM can easily handle variable bit rate services. Additionally, no bandwidth is consumed unless information is actually being transmitted. Further bandwidth efficiency can be gained through statistical multiplexing bursty traffic sources. As bursty traffic does not require continuous allocation of the bandwidth at its peak rate, several bursty traffic sources can share the bandwidth. ATM can also support circuit-oriented and continuous bit-rate services by allocating bandwidth based on the peak rate, given that sufficient resources are available.

The ATM cell consists of a 5 octet header and a 48 octet information field. As illustrated in Figure 13, the cells are carried within the payload area of a SONET frame.

Figure 13 ATM within SONET (150.520 Mbits/s)

ATM networks provide huge bandwidth with low error rates using optical fiber. In such a high-speed network environment, processing time becomes a bottleneck. The conventional OSI 7 layer protocol architecture may involve too much processing and thus a new protocol architecture is needed [36].

Figure 14 illustrates the CCITT Study Group XVIII recommended BISDN ATM protocol model. The protocol hierarchy consists of the physical (i.e. physical-medium-dependent (PMD)) layer, the ATM layer, the adaptation layer, and the higher service layer.

Figure 14 ATM adaptation layer model.

Class	1	2	3	4	
Service Layer	CBR	VBR-T	VBR CONS (PVP)	VBR CLNS	SIG
Adaptation Layer	Adapt	Adapt	CONS CONV	CLNS CONV	SIG CONV
			SAR		
	ATM				
	Physical				

CBR: - constant bit rate (e.g. PCM voice, DS1, etc)

CONV: - convergence sublayer

PVP: - packetized voice protocol

SAR: - segmentation and reassembly

VBR CLNS: - variable bit rate connectionless network service (e.g. SMDS)

VBR CONS: - variable bit rate connection-oriented network service (e.g. X.25)

VBR-T: - variable bit rate service with end-to-end timing (e.g. compressed video)

The functional layering in the BISDN protocol model does not follow the ISO model [36]. The physical-medium-dependent, or transmission, layer is responsible for the proper bit transmission and for performing functions that are necessary to insert and extract the cell flow into and out of a transmission frame. This layer is also responsible for electrical-optical conversion for BISDN utilization of optical fibre.

The ATM layer contains all the details of the ATM technique, and is common to all services. This layer is physical medium independent, and thus it is independent of the underlying PMD layer. The data unit at this layer is the ATM cell. The ATM layer performs the cell header functions, such as cell routing based on VCI/VPI and error detection on the header based on HEC. This layer also performs cell-based multiplexing and de-multiplexing and cell delineation. The information field of an ATM cell is passed transparently through the ATM

layer, and no processing, including error control, is performed on the information field at the ATM layer.

The adaptation layer and service layers of the ATM protocol model are service-dependent. The boundary between the ATM layer and the adaptation layer corresponds to the differences between the functions applied to the cell header and functions applied to the information field [36]. The adaptation layer provides the higher service level with the necessary functions which are not provided by the ATM layer; for example, preserving timing, data frame boundaries and source clock. The adaptation layer is divided into two sublayers, the Segmentation and Reassembly (SAR) Sublayer and the Convergence (CONS) Sublayer. These two sublayers provide different functions for each of the four service classes.

Four service classes are defined at the adaptation layer. Class 1 represents constant bit rate (CBR) services; CBR audio and video are examples of this service class. Class 2 represents variable bit rate (VBR), connection-oriented, such as variable bit rate audio and video. For these two classes of service, timing between the source and destination needs to be related. Class 3 also represents VBR connection-oriented services but the timing between source and destination need not be related. Connection-oriented data and signalling data are examples of Class 3 services. Class 4 services correspond to VBR connectionless services, and connectionless data belong to this class. These class characteristics are summarized in Figure 15.

Figure 15 ATM adaptation service classes.

	Class 1	Class 2	Class 3	Class 4
Timing between source and destination	related		not related	
Bit rate	constant	variable		
Connection mode	connection-oriented			connection-less

In summary, this chapter identified that there are approximately 110 million computers in use in the world today. Of this number approximately 21 million

are personal computers [28]. The great majority of these computers have a IAS or conventional architecture — an architecture that can be traced back to the mid-1940s.

This chapter also identified that communication systems were originally developed to haul voice traffic. However the above noted proliferation of computer systems have contributed to the development of integrated digital environments. The pinnacle of this development is currently envisioned as BISDN. SONET and ATM have been specified as the protocols to delivery these broadband data services.

The remaining chapters will show that, while computer systems have doubled their processing capabilities approximately every two years [28], the capability of a personal computer to effectively address the asynchronous and synchronous requirements of interactive and distributive BISDN services is questionable. The process for testing this hypothesis begins with the development of workstation and workload models. This development is the focus of the next chapter.

CHAPTER 3

WORKSTATION AND WORKLOAD MODELS

This chapter describes the models developed by the author to represent a personal computer workstation with a conventional architecture and its potential BISDN workload. The models are intended to provide a realistic abstraction of a BISDN user environment for the purpose of evaluating the workstation's performance in servicing a potential BISDN workload.

The development of these models requires a balance of real-world system details with abstraction's simplification and focus on essential characteristics. Consequently, the appropriate level of workstation and workload modelling detail was determined to be consistent with a "systems" perspective, rather than a "detailed design" perspective.

That is, there is sufficient detail to provide a description of each system resource as a "black box" characterized by multiple waiting and interrupt queues of infinite size, and a throughput capacity. Similarly, the workload model describes multiple job classes, each with a priority, activation rate, job size and chain of resources to be visited. Thus the intended level of detail is sufficient for an analyst's requirements and insufficient for a designer's needs. This is considered to be an appropriate level of detail as the goal of this thesis is to "analyze", rather than "design", an architecture at a gross systems level.

Section 1 The Workstation Model

It is assumed, for the purpose of developing this model, that the essence of a typical computer system can be reduced to *a set of resources, a resource interconnect topology or architecture, and an arbitration scheme*. These three elements are described in the remainder of this section.

Using standard analytical techniques, a resource can be described as a combination of a queue and a service centre. As illustrated in Figure 16, a job arrives, waits — possibly zero seconds — for service, is serviced, then departs.

Figure 16 Job flow through a service centre.

The multiple subsystem design of a personal computer workstation can be expressed as a simple extension of the single server example. The conventional architecture of the workstation is reflected in the specifications that no resource may be duplicated and that the passage way between resources is a single bus. A network of service centres representing this type of personal workstation is illustrated in Figure 17.

Figure 17 A computer system modelled as a network of service centres.

Figure 17 actually represents a model of a closed system — the system starts with a specified load and new jobs neither arrive nor depart; as such it doesn't quite accurately model the workstation system being considered. It is sufficient though, to help explain the general arbitration scheme and workload flow through the system.

The simplified model, Figure 17, illustrates that there are several identifiable subsystem components being modelled. The bus resource service centre is drawn offset from the other resources in order to highlight that all other resources intercommunicate via the bus. Thus an arbitrary job task requiring a transfer of data from the disk to program memory requires service first by the disk, then the bus and finally the program memory resource service centres.

Although a real-world computer system would be expected to simultaneously "hold" multiple system resources, the modelled system does not. Instead, in the workstation model a job only "holds" a resource while it is being serviced by that resource. On service completion, the job travels instantly to the next resource described in its routing chain and waits for service in an infinity sized, input queue.

The main reasons for "sequentially" holding, rather than "simultaneously" holding, the workstation resources are to more easily identify the location and cause of system bottlenecks, and to provide more optimistic results from the servicing of a multiple job class workload. The former point is helpful in analyzing architectural constraints while the latter is useful in estimating "best-case" performance measurements.

To manage conflicting resource service requests in a multiple class workload environment, a distributed preemptive arbitration scheduling scheme was developed and adopted at each resource service centre. In this scheme, a resource is allocated to a particular job according to the job's priority class, its interruption status and the length of time it has been enqueued.

Specifically, a job arrives at a resource service centre and notifies the resource's service arbitrator that it requires service. If the resource is idle then the arbitrator simply grants the job's request. However, if the resource is busy, then the arbitrator compares the priority class of the arriving job with the priority class of the job currently being serviced. If the arriving job has a higher priority than the in-service job then the arbitrator will interrupt the in-service job, assign it an interrupted status and place it in a special interrupt queue. The resource

arbitrator then grants service to the arriving job. In contrast an arriving job, that has a priority class that is either equal to or less than the current in-service job, is enqueued until it becomes the head of the waiting queue.

After a job's servicing requirement is satisfied, it informs the resource arbitrator that it is completed and is leaving the resource. The completed job leaves the resource and travels to the next resource in its chain or job route. The resource arbitrator marks the resource as idle and examines the resource's regular and special interrupt queues for waiting jobs. If any exist, the arbitrator selects the job with the highest class priority. If both queues have equal priority jobs waiting then the job in the interrupt queue is given precedence. Otherwise, simply the highest priority , longest waiting job is serviced.

This cycle of arriving, enqueueing, being serviced, possibly being interrupted, then departing is repeated for each job as it journeys through the model according to a job-influenced path. The cycle continues until the simulation run time has expired and the resource is idle, the regular queue is empty and the interrupt queue is empty. An illustration of a generic interim model representing a resource service centre with a regular and interrupt queue is provided as Figure 18. Each modelled resource has an identical resource service arbitrator.

Figure 18 An interim resource model with an interrupt queue.

To address the multiple class aspect of the workload, the interim model shown in Figure 18 is extended to provide separate waiting and interrupt queues for each job class. This model is illustrated in Figure 19.

Figure 19 Resource model with separate queues for waiting and interrupted jobs for each job class priority.

As shown in Figure 20, the resource model structure illustrated in Figure 19 represents the fundamental building block used in forming the workstation model. This structure can be considered a series or network of queues.

Figure 20 The workstation modelled as a network of queues.

Section 2 Workload Model

The workload to be applied to the model described in the previous section is the focus of this section. Six job classes are defined and described below. From the lowest to highest priority, these job classes are:

1. Facsimile;
2. Data Transfer;
3. Interactive;
4. Voice;
5. Low Resolution Video; and
6. High Resolution Video.

Facsimile Job Class

The facsimile job class was included in the workload mix because of the recent emergence of facsimile transfers as a common form of communications. A fax job was defined to be either the transmission or receipt of a multi-page facsimile document according to CCITT Facsimile Group IV specifications and consistent with literature references [36]. This application is envisioned to operate in a background mode, much like a current electronic mail application would.

Upon activation, a facsimile job is designated as a being “received” or “transmitted”. A fax job being received is processed in either batch or interactive mode. A batch job is automatically received by the workstation and directed to the printer without user intervention. An interactive job arrives at the workstation then, under user control, is read and either discarded, saved to disk or printed.

A fax job being transmitted implies that a user will cause a facsimile document to be transmitted to a single destination. The source of a facsimile job being transmitted is assumed to be either a digitizer attached to the workstation, a file editor or an existing disk file.

Facsimile job parameters are provided in Figure 21.

Figure 21 Facsimile job parameters

Class Name: Facsimile

Class Priority: 1 (1 is lowest, 6 is highest)

Job Activation Rate: Exponential(200) seconds.

Job Size: Each job consists of Uniform(10 to 100) pages; each page is 64 kbits.

General Description: Jobs move through the model in 64 kbit page blocks.

Job Routes Modelled:

Route 1: An arrival with batch processing:

Case a) Fax Board to Bus to Graphics Memory.

Case b) Fax Board to Bus to Graphics Memory to Bus to Disk.

Case c) Fax Board to Bus to Graphics Memory to Bus to IOPORT (printer).

Route 2: An arrival with interactive processing:

Case a) Fax Board to Bus to IOPORT (printer).

Route 3: A transmitted job originating with a digitizer:

Case a) Digitizer to Bus to Fax Board.

Route 4: A transmitted job originating with the digitizer, with preview and file options.

Case a) Digitizer to Bus to Graphics Memory to Bus to Fax Board.

Case b) Digitizer to Bus to Graphics Memory to Bus to Fax Board to Bus to Disk.

Route 5: A transmitted job created with the graphics editor.

Case a) Graphics Memory to Bus to Fax Board.

Case b) Graphics Memory to Bus to Fax Board to Bus to IOPORT (printer).

Case c) Graphics Memory to Bus to Fax Board to Bus to Disk.

Route 6: A transmitted job originating from an existing disk file.

Case a) Disk to Bus to Graphics Memory to Bus to Fax Board.

The probabilities associated with Figure 21 facsimile job routing options are identified in Figure 22.

Figure 22 Facsimile job's routing probabilities.

EVENT	PROBABILITY	COMMENT
OVERVIEW: AN ARRIVAL A DEPARTURE	$P(A) = .5$ $P(D) = .5$	A facsimile job can be designated either an arriving or departing job.
ARRIVALS: ROUTE 1: CASE A: CASE B: CASE C:	$P(R1) = .5$ $P(R1A) = .4$ $P(R1B) = .4$ $P(R1C) = .2$	Arrivals follow only routes 1 or 2. $P(\text{Route 1}) = P(\text{Route 2})$ Conditional probabilities
ROUTE 2: CASE A:	$P(R2) = .5$ $P(R2A) = 1.0$	
DEPARTURES: ROUTE 3: CASE A:	$P(R3) = .25$ $P(R3A) = 1.0$	Departures follow only routes 3, 4, 5 or 6. Routes 3,4,5 and 6 are equally likely.
ROUTE 4: CASE A: CASE B:	$P(R4) = .25$ $P(R4A) = .5$ $P(R4B) = .5$	
ROUTE 5: CASE A: CASE B: CASE C:	$P(R5) = .25$ $P(R5A) = .4$ $P(R5B) = .2$ $P(R5C) = .4$	
ROUTE 6: CASE A:	$P(R6) = .25$ $P(R6A) = 1.0$	

The information from Figure 22 is provided as conditional probabilities in Figure 23.

Figure 23 Conditional probabilities for the routing of facsimile jobs.

The routing and conditional probability information from these two previous figures is used in establishing the state diagram in Figure 24. This diagram shows the probability of a facsimile job's inter-resource movement at any given instance in time. This information can be used to affirm the reasonableness of the model's simulation results. It also hints at the possibility of the system bus being a potential bottleneck.

Figure 24 Resource visitation probabilities for a facsimile job.

Data Transfer Job Class

For illustration purposes, assume that you want to make an investment in a few high technology companies listed on the Toronto Stock Exchange. A strategy followed by some investors is to apply financial measures to companies within an industry sector and identify undervalued, but financially sound companies. While the necessary information exists, a current copy is something most users do not have stored away on their personal computers. Instead, the information is typically “downloaded” to the user’s system for a fee.

Though I am not endorsing this or any other technique for making investments, the underlying point is this: *in an information age, no machine is an island*. It is for this reason, and its common occurrence in real life, every day situations that the data transfer job class is included in the workload mix.

A data transfer job is defined to be a relatively low priority, background task concerned with either transferring or receiving a file. Jobs in this class have been defined to range in size from a few hundred kilobytes to a few hundred megabytes [36].

A data transfer job is designated as “incoming” or “outgoing”. An incoming job may be automatically filed on a disk or be displayed on the screen; an outgoing data transfer job simply retrieves the appropriate file and transfers it to the intended destination.

Representative parameters for data transfer class jobs are provided in Figure 25.

Figure 25 Data transfer parameters.

Class Name: Data Transfer

Class Priority: 2 (1 is lowest, 6 is highest)

Job Activation Rate: Exponential(5400) seconds

Job Size: Each job requires Uniform(30 to 2700) seconds

General Description: Jobs move through the model in 64 kbit packets

Job Routes Modelled:

Route 1: An incoming job

Case a) IOPORT to Bus to Program Memory to Bus to Disk

Case b) IOPORT to Bus to Program Memory to Bus to CPU
to Bus to Program Memory to Bus to Disk

Route 2: An outgoing job

Case a) Disk to Bus to IOPORT

The probabilities associated with the routing options of Figure 25 are identified in Figure 26.

Figure 26 Data job routing parameters.

EVENT	PROBABILITY	COMMENT
OVERVIEW:		
AN INCOMING JOB	$P(I) = .5$	A data transfer job can be designated only incoming or outgoing
AN OUTGOING JOB	$P(O) = .5$	
INCOMING:		
ROUTE 1:		
CASE A:	$P(R1A) = .8$	Conditional probabilities
CASE B:	$P(R1B) = .2$	
OUTGOING:		
ROUTE 2:		
CASE A:	$P(R2A) = 1.0$	

The conditional probabilities of Figure 25 are broken down on a routing subcase basis and illustrated in Figure 27.

Figure 27 Conditional probabilities for data jobs.

The routing and conditional probability information from the two previous figures is used to create the inter-resource state probabilities for a data job shown in Figure 28. As was the case in the discussion of the facsimile job class, the system bus resource is shown to be a relatively heavily used resource in the data job's flow through the system.

Figure 28 Resource routing probabilities for a data job.

Interactive Job Class

The interactive job class is a common component of most personal computer sessions and has been a long standing ingredient in benchmarks of system performance. It typically represents a user session that is characterized by the foreground execution of predominantly simple commands which are expected to be completed within a short time span.

Traditionally, a mixture of simple editing, compiling and executing a program is used to represent an interactive workload. This mixture, within the thesis, is

modelled as a cycle of actions that load and manipulate a program, then direct the output to a disk, printer or graphical device.

Literature, and the experience of most users, leads to the conclusion that user productivity rapidly decreases as response time creeps towards and beyond a one second plateau. The increase in response time may result from increased contention for system resources along a job's execution path [36].

From this perspective, provided that interactive jobs fairly compete for system resources, then interactive response time can be a simple and immediate, albeit gross, indicator of overall system performance. Furthermore interactive response time can be considered a dependent variable against which the effects of an independent variable — the offered workload — can be charted.

Representative parameters and routing probabilities are provided in Figures 29 and 30 respectively.

Figure 29 Interactive job parameters.

Class Name: Interactive

Class Priority: 3 (1 is lowest, 6 is highest)

Job Activation Rate: One per second

Job Size: 64 kbits

General Description: Jobs move through the system in 64 kbit packets; this value is a valid

Memory Management Unit pagesize parameter for SUN 3 workstations.

Job Routes Modelled:

Route 1:

Case a) Disk to Bus to Program Memory to Bus to CPU to Bus to Program Memory to Bus to Disk.

Case b) Disk to Bus to Program Memory to Bus to CPU to Bus to Program Memory to Bus to IOPORT.

Case c) Disk to Bus to Program Memory to Bus to CPU to Bus to Program Memory to Bus to Graphics Memory.

Figure 30 Interactive job routing probabilities.

EVENT	PROBABILITY	COMMENT
OVERVIEW:		An interactive job is constantly cycling through the system.
ROUTE 1:	$P(R1) = 1.0$	
CASE A:	$P(R1A) = .6$	There is only one major route with three options for how the session terminates.
CASE B:	$P(R1B) = .2$	
CASE C:	$P(R1C) = .2$	

The information from Figures 29 and 30 is condensed into the conditional probabilities depicted in Figure 31.

Figure 31 An interactive job's conditional probability routing.

The probability of an interactive job's transmission amongst system resources is illustrated in Figure 32. The potential for the system bus to be a potential bottleneck is apparent in this figure.

Figure 32 Resource visitation probabilities for an interactive job.

Voice Job Class

The voice job class represents a sampling of current and envisioned telephony services. It is included in the model's workload as future telephony services are assumed to be primarily available through an integrated workstation that handles voice packets as digitized objects that can be manipulated. A standard telephone may coexist near the workstation, but would be seldom used; e.g. during power outages.

The rationale for integrating telephony services in a multipurpose workstation is based on the idea that the modelled workstation is intended to provide a single interface to the user for all digital services. This concept is consistent with, but an extension of, the idea of using "Your Workstation as a Super Phone" as presented at a Feb 19, 1991 IEEE seminar by a representative of Bell Northern Research's (BNR) Computer Research Laboratory (CSL).

The scope of telephony applications has been restricted to voice jobs performing conversational and messaging functions. The need to retrieve voice messages from a disk storage device are excluded from the model because it is assumed they would have minimal impact on the system workload and hence performance results. This is based on the information given in Figure 34 regarding the relatively low probability of leaving a message (i.e. 10 per cent of incoming calls) and the assumption that a message playback would follow a very short path (i.e. from the disk to the bus to an external speaker or hands set).

Voice job parameters are described in Figure 33, while routing probabilities are identified in Figure 34.

Figure 33 Voice job parameters.

Class Name: Voice

Class Priority: 4 (1 is lowest, 6 is highest)

Job Activation Rate: One every 64th of a second

Job Size: 1 kbits

General Description: A default 64 kbit voice job is decomposed for smoother flow through the system into 64 times 1 kbit packets.

Job Routes Modelled:

Route 1: An incoming voice job:

Case a) Voice Board to Bus (to a speaker or handset).

Case b) Voice Board to Bus to Disk.

Route 2: An outgoing voice job:

Case a) (microphone or handset to) Bus to Voice Board.

Figure 34 Voice job routing probabilities.

EVENT	PROBABILITY	COMMENT
OVERVIEW:		
INCOMING	$P(I) = .5$	A voice job can be designated only one of two equally likely events..
OUTGOING	$P(O) = .5$	
INCOMING:		
ROUTE 1:	$P(R1) = 1.0$	Conditional probabilities
CASE A:	$P(R1A) = .9$	
CASE B:	$P(R1B) = .1$	
OUTGOING:		
ROUTE 2:	$P(R2) = 1.0$	Only one option for an outgoing call.
CASE A:	$P(R2A) = 1.0$	

The conditional probabilities for a voice job executing a particular case of a route is shown in Figure 35.

Figure 35 Conditional probability of voice route selection.

The probabilities associated with a voice job's inter-resource transitions are presented in Figure 36. Again the central importance of the system bus is noted.

Figure 36 Resource visitation probabilities associated with a voice job.

Low Resolution Video Job Class

A video job is defined to be a logical stream of digital packets having a near real time processing requirement. In particular, a low resolution video job represents all video applications having a relatively narrow band requirement. For example, a low scan rate, compressed, video conference call over an ISDN 64 kbps B channel would be at the low end this classification. Conversely, an uncompressed 640 x 512 pixel, 24 bit per pixel, 30 frame per second video output from a BISDN application would be near the high end.

A low resolution video job can originate with, or have as its destination, the modelled workstation. All jobs originating at the workstation capture the image with a video camera — which is integrated into the workstation — and transmit it to the decoder and modem. A job arriving at the workstation follows a reverse route that ends with the image appearing on the workstation monitor.

Low resolution video job parameters and routing probabilities are described in Figure 37.

Figure 37 Low resolution video job parameters and routing probabilities.

Class Name: Low Resolution Video

Class Priority: 5 (1 is lowest, 6 is highest)

Job Activation Rate: One every 30th of a second

Job Size: 2,621.44 kbits per low resolution video frame

General Description: The default video packet size is defined as $640 \times 512 \times 8 = 2,621.44$ kbits every 1/30th of a second. The low resolution video board must therefore be capable of processing $2,621.44 \times 30 = 78,643.2$ kbits per second.

Job Routes Modelled: Routes 1 and 2 are equally likely.

Route 1: An incoming low resolution video job:

Case a) Modem to Bus to Low Resolution Video Board to Bus to Graphics Memory

Route 2: An outgoing low resolution video job:

Case a) (camera to) Low Resolution Video Board to Bus to Modem

The conditional probabilities associated with a low resolution video job are shown in Figure 38.

Figure 38 Low resolution video job routing probabilities.

Integrating the information from the two previous figures — regarding conditional sub-route execution and resource visitation — results in the low resolution video job's inter-resource visitation probabilities illustrated in Figure 39.

Figure 39 Resource visitation probabilities for low resolution video jobs.

High Resolution Video Job Class

As defined earlier, a video job is defined to be a logical stream of digital packets having a near real time processing requirement. In contrast to low resolution jobs, high resolution video jobs represent all video applications having a broadband requirement. For example, high definition television (HDTV), entertainment video and sophisticated video aspects of multimedia applications.

High resolution will apply to all requirements beyond those previously scoped as low resolution. For example, a high resolution job could have a 1600 x 1280 pixel, 24 bit per pixel, 60 frame per second stream of packets associated with it.

A high resolution video job can originate with, or have as its destination, the modelled workstation. All jobs originating at the workstation capture the image with a video camera — which is integrated into the workstation — and transmit it to a high resolution decoder and very high speed modem. A job arriving at the workstation follows a reverse route that ends with the image appearing on the workstation monitor.

High resolution video job parameters and routing probabilities are described in Figure 40.

Figure 40 High resolution job parameters and routing information

Class Name: High Resolution Video

Class Priority: 6 (1 is lowest, 6 is highest)

Job Activation Rate: One every 60th of a second

Job Size: 49,152 kbits per video frame

General Description: The default highres video packet size is defined as $1600 \times 1280 \times 24$
= 49,152 kbits per frame every 1/60th of a second. The high resolution
video board must therefore be capable of processing 2,949,120 kbits per second.

Job Routes Modelled: Routes 1 and 2 are equally likely.

Route 1: An incoming high resolution video job:

Case a) High Speed Modem to Bus to High Resolution Video Board to
Bus to Graphics Memory.

Route 2: An outgoing high resolution video job:

Case a) (camera to) High Resolution Video Board to Bus to Modem

Expressed conditionally, the sub-routing information for high resolution video jobs appears in Figure 41.

Figure 41 Conditional probability low resolution video job routing.

The previously presented conditional probability and routing information results in the inter-resource visitation diagram for high resolution video jobs shown in Figure 42.

Figure 42 Resource visitation probabilities for high resolution video jobs.

Section 3 Resource Parameters

The workstation model's resource parameters, expressed in kilobits per second, and the source of these parameters are listed below. Where appropriate, calculation of values are described.

1. Bus:
 - a. Value: 320,000;
 - b. Source of Information: Chipset Reference Manual — 32 bits at 40 MHz;

2. CPU:

- a. Value: 128,000;
- b. Source of Information: Chipset Reference Manual
- c. Calculation of Value: 5 MIPS (max) for a 80386. The average number of bytes per instruction is given as 3.2. Then the average number of bytes, based on the maximum MIPS rating, is $3.2 \times 5 \text{ M} = 16 \text{ MB/sec}$ or 128,000 kbits/sec;

3. Digitizer:

- a. Value: 125;
- b. Source of Information: Based on a simple test of the Department's Corel digitizer located in room 326. A 50% dense image was scanned in one minute resulting in a 1,062,856 byte (uncompressed) file. Thus the digitizer processed 125,047 or approximately 1,062,856 byte (uncompressed) file. Thus the digitizer processed 125,047 or approximately 125 kilobits per second;

4. Disk:

- a. Value: 16,000;
- b. Source of Information: BYTE Nov 89 pp299 -305: Are 32 bits enough;
- c. Calculation of Value: SCSI disk controller $\leq 2\text{MB/sec} = 16,000 \text{ kbps}$;

5. Fax Board:

- a. Value: 64;
- b. Source of Information:[36];

6. Graphics Memory:

- a. Value: Assumed to be the same as program memory;

7. High Resolution Video Board:

- a. Value: 1,887,436.800;
- b. Source of Information: —Sun SPARCstation 330 VX/MVX: 1280 x 1024 screen with 24 bit colour; — assumed : 60 frames per second.
- c. Calculation of Value: $1280 \times 1024 \times 24 \text{ bit colour} \times 60 \text{ frames/sec} = 1,887,436,800 \text{ bits/sec}$, or approximately, 1.89 Gbps;

8. High Speed Modem:

- a. Value: 7,865;
- b. Source of Information: Assumed value; must match the throughput of the low resolution video board;

9. IOPORT:

- a. Value: 19,200;
- b. Source of Information: Intel Handbook for peripheral controller;
- c. Calculation of Value: 2.4 MBytes/sec;

10. Low Resolution Video Board:

- a. Value: 7,865;
- b. Source of Information: Low resolution video conference = 7,864,320 bits/sec or approximately 7,865 kbps [18];
- c. Calculation of Value: $256 \times 256 \times 8 \text{ bit colour} \times 15 \text{ frames/sec} = 7,864,320 \text{ bits/sec}$ or approximately 7,865 kbps;

11. Program Memory:

- a. Value: 457,143;
- b. Source of Information: Sun Microsystems Inc.: SPARCstation 2 Series; 10/90; DE425-0/200K
- c. Calculation of Value: — 70 ns access time $\times$ 32 bit data path = 457,142,857 bits per second.

12. Very High Speed Modem:

- a. Value: 1,887,436.8;
- b. Source of Information: Assumed;
- c. Calculation of Value: Must match the throughput of the high resolution video board;

13. Voice Board:

- a. Value: 64;
- b. Source of Information: [24]; also values for 32 and 16 are given.

CHAPTER 4

APPROACHES TO COMPUTER SYSTEM MODEL PERFORMANCE EVALUATION

Computer system performance can be evaluated using direct measurement, analytical modelling or simulation modelling techniques [19]. If the system under examination already exists, is operational and tuned, then direct measurements can be obtained and the system's performance evaluated. However if the system, or a comparable one, isn't available, then a model of the desired system must be developed and solved using either analytical or simulation techniques.

As described later in this chapter, analytical techniques have long been associated with the solution of computer systems. Although they can be validly applied in many situations, the complexity of some models makes these techniques invalid and/or impractical. The developed models, with systems that have mixed (i.e. open and closed) workload birth-death processes, multiple queue types, multiple job classes, numerous system states and multiple job chains or routes, qualify as being sufficiently complex to render the use of analytical techniques impractical.

In contrast, performance results obtained from simulation analysis can accommodate any level of model detail. For example, assumptions that might otherwise be violated in the application of analytical techniques can be incorporated in the design of the simulation program. Similarly, complex workstation and workload model interactions can be recorded through "direct" measurement of the simulated activity. These capabilities however, are not without a price. As the level of model detail, the number of measurements required at each simulation step, and the length of the simulated period increases, obtaining results from a simulation program can also require a substantial amount of time.

This chapter is organized into five sections. This information provides the reader with an abridged analytical perspective of computer system evaluation techniques to support the author's conclusion that these methods must be discarded and that simulation must be adopted for solving the developed models. It also establishes *a priori* targets to use as yardsticks for evaluating the simulation results.

The five sections of this chapter are entitled: *Historical Perspectives; Analytical Modelling of Computer Systems; Analytical Evaluation of Separable Queuing Networks; Simulation and Performance Targets*.

Section one, *Historical Perspectives*, highlights research into analytical techniques applied to computer system modelling and performance evaluation using queuing theory. This section describes Kleinrock's use of classic queuing theory to initially model a computer system as a single server queue representing the processor [17]. More complex models of computer systems and their workloads were subsequently researched. This led to the representation of complex computer systems as queuing network models in general, and separable queuing network models in particular [17]. The developed workstation model is an example of a separable queuing network.

Section two, *Analytical Modelling of a Computer System*, stipulates that a separable queuing network requires system resources to be represented as a network of queues — a collection of service centres — and users or transactions as customers. Thus rather than modelling a complex system as a single service centre with potentially complex characteristics, queuing networks model systems as a collection of service centres with relatively simple characteristics. As this approach is restricted to computer systems, researchers are able to select an appropriate subset of networks of queues and develop evaluation algorithms with meaningful performance measures that exhibit an appropriate balance between accuracy and efficiency. Similarly these algorithms could be expressed in computer system terms rather than queuing semantics and notation [20].

Queuing network models attain a good balance between computational complexity and output accuracy. The cost of attaining additional accuracy is significantly higher than the incremental benefit for a wide variety of applications. For example, the solution of queuing network models has been shown to be accurate to about 5 to 10% for utilizations and throughputs, and to within 10 to 30% for response times. This is suitably accurate for many design and analysis applications and is consistent with the accuracy of other component parameterization — for example, workload characterization [20].

Section three, *Analytical Evaluation of Separable Queuing Networks* briefly describes two analytical evaluation techniques — bounding analysis and estimation analysis. The former defines optimistic and pessimistic bounds on performance, while the latter defines average performance point values. Bounding anal-

ysis techniques are typically less accurate and less computationally complex than estimation analysis techniques. Subject to the goal of a particular performance study, either or both of these techniques can be utilized to evaluate computer system performance [20].

Other analytical performance evaluation techniques of computer system models are outside the scope of this discussion. Some, such as Markov Chains, are inapplicable because of finite state conditions — for example, they require finite queue lengths and a reasonably countable number of valid system states — while others are too detailed for the purpose of this study. An example of the latter approach is hierarchical modelling; here a system is decomposed into submodels and then recombined into flow equivalent service centres (FESCs). Information concerning the former topic is available in [25], and the latter in [20].

Section four describes that in modelling situations where analytical evaluation techniques are inapplicable or impractical, *Simulation* modelling and analysis is applicable. This could be the case if analytical modelling assumptions are violated or multiple class, multiple resource systems make the solution prohibitively computationally expensive for complex multiclass models [20]. The developed workstation model could be so described. Simulation modelling can accommodate any level of model detail but can also require considerable development and evaluation time.

Section five, *Performance Targets*, or service objectives, are developed for the workload and workstation model components. These objectives are expressed in terms of time and space. For the workload model component, these parameters can be interpreted as service time and buffer space requirements. For the workstation service centres, time and space can be interpreted as resource utilization and queue space requirements.

The ideas introduced above are developed in the following material.

Section 1 Historical Perspective

Computer system performance is strongly affected by queuing delays that may result from multiple users requesting and receiving service from the system's resources. As such, performance models were initially based on queuing models [20].

Kleinrock, for example, modelled and evaluated a time sharing multiprogramming computer system as a single server queue where the single server queue represented the system processor and other system resources were dismissed [17].

A single server queue system is illustrated in Figure 43. It indicates that a job or customer arrives at a system, queues for service, is dispatched to the server according to some discipline, is serviced and then departs the system.

Figure 43 A single server queue system.

A subsequent approach, the machine repair model, viewed the entire computer system as a single resource [19]. This model, as illustrated in Figure 44, is a closed system with a finite number of job source terminals.

Figure 44 The machine repair model of a time sharing system. Source Fig.1 [LAVE89].

The repair model assumed that one job existed in the system for each user terminal. In this model it was assumed that a cycle of the model represented a user thinking, entering terminal input, a job queuing for and then receiving service,

followed by the user receiving output at the terminal as a result of the servicing. The servicing and thinking times were assumed to be exponentially distributed and the queuing discipline was assumed to be First Come, First Served (FCFS) [19].

Neither of these two initial approaches accounted for more realistic and complex characteristics of computer systems — for example, multiple job classes may compete for multiple resources. Early representation of a computer system as a network of queues was subsequently investigated by [10] and several others [12], [7], [21]. These researchers assumed all jobs were probabilistically identical.

Gaver considered a closed network with two queues. The first queue was a single server queue which represented the processor, while the second was a multi-server queue representing multiple identical I/O devices. A semi-Markov process was used to analytically evaluate the model but was found to be very limiting because the model typically permits only one arbitrary service time and is practically intractable for networks with more than two queues.

Buzen's central server model, a special case of the more generalized work performed by Jackson [15] and Gordon and Newell [12], more realistically has a queue for each I/O device [19]. The central server model is illustrated in Figure 45.

Figure 45 Central server model of a multiprogramming system. Source: Fig.2 [LAVE89].

Collectively, these researchers modelled a computer system as a closed queuing network with a fixed number of jobs and a single job class. The steady state performance solution for this type of model could be determined as the

product of simple terms, one term for each queue, divided by a normalization factor. Buzen additionally presented a relatively efficient algorithm for recursively determining this normalization factor. However neither the algorithm nor the product form equations are presented as more comprehensive queuing network solution techniques were subsequently developed. The significance of this body of research, and Buzen's work in particular, is that it laid the foundation for computer performance evaluation using product form queuing networks [19].

Subsequent research established the applicability of product form queuing network models to progressively more realistic computer systems. Open queuing networks with varying multiprogramming workloads were investigated by Avi-Itzhak and Heyman [1]. Baskett et al [4] introduced multiple job class workloads with probabilistic routing, or Markov chaining, to open, closed and mixed queuing networks.

Reiser and Kobayashi [30] establish a more efficient, convolution method for computing the normalization factor of the product form solution. Their approach reduces the number of arithmetic operations required to calculate this factor from Buzen's $\prod_{k=1}^K \binom{N_k + J - 1}{J - 1}$ to $J \prod_{k=1}^K \frac{(N_k + 2)(N_k + 1)}{2}$ where K refers to the number of job classes, N_K the number of jobs in class K , and J the number of queues (service centres).

Reiser and Sauer [29] further advanced the study of queuing network modelling by developing the Mean Value Analysis (MVA) algorithm for calculating throughputs, mean queue lengths and mean response times without determining the normalization constants.

The convolution and MVA approaches can be computationally prohibitive for networks that have many chains, or few chains with many jobs per chain. Schweitzer [31] addressed this problem by developing an approximation solution technique. Subsequently more accurate approximation solution techniques were developed by several researchers (e.g. [6], [36]).

To apply these more efficient solution techniques, certain assumptions are applied to queuing networks resulting in a restricted model known as separable queuing networks. A model with priority job class scheduling — for example, the thesis' workstation model — would violate these assumptions [20].

Hierarchical decomposition is a variation of separable queuing network modelling for evaluating computer system performance when queuing network assumptions are violated [20]. It is a method for modelling a computer system

as multiple submodels that are solved individually then recombined to provide a system solution.

The hierarchical approach defines aggregate and complement subsystems. The aggregate, which may be an arbitrarily defined subsystem, interacts with the complement or complementary network, which represents all other subsystems. The aggregate is replaced by a single service centre that approximates the aggregate's average residence time and the aggregate's departure rate and distribution. The replacement service centre is a flow equivalent service centre (FESC), which is virtually a 'black box' to the complementary network. That is, the aggregate subsystem's definitional parameters are required by the complementary network but the particular structure of the aggregate subsystem is unimportant.

Hierarchical decomposition can be applied repeatedly to a computer system model to result in multiple levels of FESCs. Typically the decomposition is repeated with a 'top-down' perspective. The most general level, *zero* is at the pinnacle of the hierarchy while the most detailed level, *L*, is at the bottom. The solution of hierarchical networks is performed bottom-up.

Global balance is the general analytical technique for evaluating closed, non separable queuing networks. This technique involves creating and solving linear equations that describe the behaviour of these models. Global balance requires one equation per state of the network, where a state can be considered the placement of customers at service centres. A system model — relatively simple in comparison to the workstation and workload models developed in this thesis — with six service centres, five job classes and five customers in each job class has more than 10^{12} states. Accordingly, this technique is practical for only small models as the growth in size of the state space is given by $\prod_{c=1}^C \left[\begin{matrix} N_c + K - 1 \\ K - 1 \end{matrix} \right]$ where $\left[\begin{matrix} n \\ p \end{matrix} \right]$ indicates the number of ways of choosing p objects from a population of n [20]. The complexity of the developed workstation and workload models makes this approach impractical.

An alternate approach to solving complex non separable queuing networks is simulation. Though simulation modelling and evaluation can also require significant time and skill to produce meaningful evaluation results, this approach is sufficiently flexible to represent the level of detail in the workstation and workload models developed in the previous chapter.

In summary, the modelling and evaluation of computer systems was initially based on classical queuing theory. From this perspective, either a single computer

resource or the entire system was modelled as a single server queue. Subsequent research lead to the development of queuing network models which represented the computer system as a collection of service centres and the workload as customers or jobs, possibly of multiple classes. Separable queuing networks represent a subset of queuing networks that are more efficiently solvable. There are several exact and approximate techniques for solving separable queuing networks. Systems that violate the separable queuing network assumptions, for example priority scheduling, can be solved instead using hierarchical decomposition with FESCs. However this technique is only practically feasible for small networks with few job classes because of the rapid growth global balance state space. Simulations however, can be used to evaluate the performance of computer systems with any level of complexity.

Section 2 Analytical Modelling

Basic Concepts

Fundamental performance measures are based on three parameters [20]:

1. T , the length of time a system is observed;
2. A , the number of arrivals observed during this period; and
3. C , the number of completions observed.

The above measures can be used to define the arrival rate — the rate at which jobs arrive at the system ($\lambda = A / T$), and the departure rate, or throughput — the rate at which completed jobs leave the system ($X = C / T$).

In a single resource system, the amount of time the resource is busy, β , can additionally be measured. This quantity can be used to help define the resource's utilization, ($U = \beta / T$) and the average service requirement per request, ($S = \beta / C$).

The relationship between a resource's utilization, the departure rate and the average service time per request is known as the *Utilization Law*. It states that a resource's utilization, U , is equal to the product of the departure rate and the service requirement per request, $X S$. This equation can be simply proven by manipulating the formulae in the previous paragraph.

Little's Law is the second important fundamental relationship. It defines the number of jobs in a system, N , to be the product of system throughput, X , and the

length of time each job stays in the system, R . The average system residence time, R , per request is defined as accumulated waiting time, W , divided by the number of completions, C . Accumulated waiting time is represented as the shaded region in Figure 46.

Figure 46 System arrivals and completions

The relatively thin line in Figure 46 represents job arrivals, while the thicker line, job completions or departures. The vertical distance between these two lines represents the number of jobs in the system, N . This quantity, N , is defined as the product of the accumulated time in the system, W , divided by total observation time, T . Algebraic manipulation results in Little's Law, $N = XR$.

Little's Law is of primary importance as the theoretical basis for algorithms that evaluate queuing networks. It is widely applicable to computer system performance evaluations because two of its three parameters are usually known, and it has weak assumptions that permit its use to verify measurement data.

Little's Law is applicable at all levels of a system's abstraction. Thus it can inspect a resource, a resource and its queue, a subsystem of resources, or the entire system. This ability to scope its focus at different levels is especially appropriate for the developed model as individual and aggregate resource and workload measurements are required.

The *Forced Flow Law* states that the throughputs at a resource must be proportional to the throughputs of the system. Restated, if a single system transaction visits a resource ν times, then the resource will experience ν times more completions than the system.

A natural derivative of the Forced Flow Law specifies that a job's service demand at a particular resource, D_k , equals the product of the job's service request at the resource, S_k , and the job's visitation rate, V_k . Both, resource throughputs and service demands, are of interest in the model's evaluation.

The *Flow Balance Assumption* equates the number of arrivals and the number of completions (i.e. $A = C$, therefore $\lambda = X$). It applies to systems that can handle the offered workload without having any of its resources utilized more than one hundred percent. The Forced Flow Law, Little's Law and the Flow Balance Assumption can collectively determine a resource's utilization based on system arrival rates. Further, the forced flow law indicates that a resource's composite utilization can be determined by aggregating individual job class demands. This result is directly applicable to the evaluation of the thesis model as resource utilization, by single and aggregate job classes, is of significant interest.

Figure 47 summarizes the fundamental laws that were previously described.

Figure 47 Fundamental Laws.

The Utilization Law:	$U_k = X_k S_k = X D_k$
Little's Law:	$N = X R$
The Forced Flow Law:	$X_k = V_k X$

Queuing Networks

As noted previously, computer system modelling and analytical performance evaluation is rooted in classical queuing theory. Early work by Kleinrock utilized queuing theory to model a computer system as a single server queue. Subsequent variations of the classical queuing approach attempted to more accurately reflect the architecture and behaviour of a computer system and its workload. This search for a 'good fit' between theory and practice lead to the development of separable queuing network models of computer systems.

A separable queuing network is an important technique for modelling and evaluating computer systems. It represents a computer system as a network of queues or, equivalently, a collection of service centres. The model is developed by representing each system resource as a service centre and each job in the offered workload as a customer.

Jobs within the workload flow through the network according to a predetermined route, or chain. All service centres identified in the chain are visited according to a predetermined sequence. At each visited service centre, the job initially waits for service then receives service and, upon service completion at this centre, continues along its route until all intended centres are visited. The job is then terminated; this is indicated by the job leaving an open system model, or recycling within a closed system model.

Separable Queuing Networks A separable queuing network is a subset of general networks of queues consisting of separable queuing networks extended where necessary to accurately represent particular computer system characteristics.

This subset can be efficiently evaluated, which is necessary in current computer system models that may consist of hundreds of resources, dozens of workload components, each possibly consisting of many users or jobs.

Scoping the discussion to separable queuing networks implies certain assumptions to the computer system under study. These assumptions are seldom strictly satisfied but inaccuracies from violated assumptions are typically no worse than those associated with other sources — for example, inadequate performance measures [20].

General networks of queues, which violate many of these assumptions, can still be evaluated analytically, but the solution's time and space requirements grow prohibitively quickly as the size of the system increases.

Though a separable queuing network model can accurately represent a computer system's structure, it has limitations in directly describing certain system characteristics. These characteristics are:

1. Simultaneous possession of multiple resources by a single job;
2. Constraint on the number of jobs that may be memory resident at one time;
3. Job arrival blocking;
4. Adaptive job routing;
5. Forking, or spawning of sub-processes, during a jobs execution;
6. Synchronization of multiple jobs;
7. High service time variability impact on degraded system performance; and
8. Priority service scheduling disciplines.

While these limitations can be indirectly circumvented, all but the last of these is of little relevance to the thesis model. The priority scheduling discipline limitation, however, *is contrary to the modelled system*.

Section 3 Analytical Evaluation of Queuing Networks

The analytical evaluation of separable queuing network models is the focus of this section. The section begins with a discussion of basic concepts and fundamental formulae. This is followed with a description of a models's inputs and outputs. These two topics, in essence, set the background for the third topic — a detailed description of analytical techniques, formulae and algorithms for efficiently solving separable queuing network models.

Queuing Network Model Inputs

A multiple class workload model may consist of C customer classes. Each class has its own workload characteristics, but jobs within each class type are homogeneous. Job classes represent transaction, batch or terminal workloads.

A transaction is characterized by its arrival rate, a population that varies over time and jobs that leave the system upon completion. A batch workload is characterized by an average number of active jobs, a fixed population size and jobs that circulate continuously around the system. A terminal workload is described by the number of active terminals and the average user think time. The behaviour of terminal jobs with zero think time is similar to the behaviour

of batch jobs; conversely, with nonzero think time, terminal jobs are similar to system transactions that exclude the terminal subsystem.

Models consisting of only transaction workloads are considered “open” models, those with batch or terminal workloads are considered “closed” models. Mixed models have workload components that are open and other workload components that are closed.

The service demand of a job at resource k , D_k , represents the amount of time the job requires in service at this resource. It is determined by dividing the measured busy time of resource B_k by the measured number of system completions, C . Generalizing this formula to a multiple server, multiple class workload model, results in a job's service demand at all resources to be the summation of its individual resource service requirements, $D_c = \sum_{k=1}^K D_{c,k}$

Queuing Network Model Outputs

The model's outputs represent aggregate and individual class measures of system and resource centre performance. Measures of system outputs are average system response time, system throughput and average number of jobs in the system. Resource centre measures are utilization, average residence time, throughput and average queue length.

Aggregate utilization, queue length and throughput performance measures are determined by summing the respective individual per-class performance measures. The aggregate residence and system response time performance measures are determined by a weighted summation of proportionate class throughputs. Hence average system response time is defined as $R = \sum_{c=1}^C \frac{R_c X_c}{X}$ and average residence time is defined as $R_k = \sum_{c=1}^C \frac{R_{c,k} X_c}{X}$.

The basic multiple class model outputs are summarized in Figure 48.

Figure 48 Multiple class model outputs [20].

System Measures	Aggregate	<p>R Average system response time</p> <p>X System throughput</p> <p>Q Average number in system</p>
	Individual Class	<p>R_c Average class c system response time</p> <p>X_c Class c system throughput</p> <p>Q_c Average class c number in system</p>
Centre Measures	Aggregate	<p>U_k Utilization of centre k</p> <p>R_k Average residence time at centre k</p> <p>X_k Throughput at centre k</p> <p>Q_k Average queue length at centre k</p>
	Individual Class	<p>$U_{c,k}$ Class c utilization of centre k</p> <p>$R_{c,k}$ Average class c residence time at centre k</p> <p>$X_{c,k}$ Class c throughput at centre k</p> <p>$Q_{c,k}$ Average class c queue length at centre k</p>

Bounding Analysis

The simplest useful approach to computer system analysis using separable queuing networks is bounding analysis [36].

This approach requires relatively little calculation — a ‘back of an envelope’ method — that produces upper and lower bounds on system throughput and response time based on the workload’s arrival rate. Bounding analysis can be used to evaluate situations requiring limited detailed accuracy or having imprecise

knowledge of the system and its parameters. For example, in identifying system bottlenecks, evaluating configuration alternatives and long range planning.

Bounding analysis can be applied to single class queuing networks where extreme system load conditions — both light and heavy — are investigated. Application of bounding requires the assumption that a job's service demand at any centre is independent both of the number of job's in the system and the particular server that the job is currently located. As the developed model represents a multiple class queuing network, this approach has little value. A reader interested in bounding analysis is directed to [36] for details on this topic.

While bounding techniques are computationally inexpensive and useful for gross analysis, performance estimates are able to provide more accurate performance estimates.

Estimation Techniques

Estimation techniques produce point, rather than range, data as performance measure outputs. In exchange for more accuracy than bounding techniques, estimation approaches are more computationally complex. Within the context of queuing network analysis, there are single and multiple job class models for open and closed systems.

The results from single job class models are more accurate than those produced by bounding techniques, but less accurate and realistic than those produced from multiple job class models. Multiple job class models are not necessary in all situations but are appropriate when there are multiple distinct workloads, class dependent model inputs and class dependent model outputs. This is the case with the workload model developed in the thesis.

Multiple Job Class Models

Open Models

Inputs If there are C job classes in the model, then each class c has an arrival rate of λ_c and the arrival rate for all C classes is:

$$\Lambda = (\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_c)$$

Outputs The multiple class models evaluation formulae are listed according to output measure.

Processing Capacity A system's processing capacity refers to a workload level that can be sustained over a long period of time without saturating some service centre. This quantity is defined as:

$$\max \left\{ \sum_{c=1}^C \lambda_c D_{c,k} \right\} < 1$$

Throughput The forced flow law defines throughput of class c jobs at service centre k , as a function of the system workload, to be:

$$X_{c,k}(\Lambda) = \lambda_c V_{c,k}$$

Utilization The utilization law results in:

$$U_{c,k}(\Lambda) = X_{c,k}(\Lambda) S_{c,k} = \lambda_c D_{c,k}$$

Residence Time The appropriate formulae for determining residence time depends on whether queuing or delay service centre being examined. For a delay service centre, the equation is:

$$R_{c,k}(\Lambda) = D_{c,k}$$

The residence time equation for a queuing service centre is similar to the delay service centre's equation above, except for a factor representing the average number of jobs in the queue. This equation is:

$$R_{c,k}(\Lambda) = D_{c,k} \left[1 + A_{c,k}(\Lambda) \right]$$

The term $A_{c,k}(\Lambda)$ in the queuing service centre residence time equation represents the average number of jobs that an arriving class c job sees at service centre k .

$$R_{c,k}(\Lambda) = D_{c,k} \left[1 + A_{c,k}(\Lambda) \right]$$

Queue Length Application of Little's law to the residence time equation results in the following definition of the queue length of class c jobs at service centre k :

$$Q_{c,k}(\Lambda) = \lambda_c R_{c,k}(\Lambda)$$

For delay service centres, this equation can be expressed as:

$$Q_{c,k}(\Lambda) = U_{c,k}(\Lambda)$$

and for queuing service centres, as:

$$Q_{c,k}(\Lambda) = \frac{U_{c,k}(\Lambda)}{1 - \sum_{j=1}^C U_{j,k}(\Lambda)}$$

System Response Time The response time for a class c job is simply the summation of its residence times at all devices.

$$R_c(\Lambda) = \sum_{k=1}^K R_{c,k}(\Lambda)$$

Average Number in System The average number of class c jobs in the system can be determined by summing the class c queue lengths at all service centres, or by applying Little's law. These two forms are expressed in the following equation:

$$Q_c(\Lambda) = \lambda_c R_c(\Lambda) = \sum_{k=1}^K Q_{c,k}(\Lambda)$$

Closed Models A closed, multiple class model consists of C classes each having a fixed population. This workload intensity can be represented by:

$$N = (\nu_1, \nu_2, \nu_3, \dots, \nu_c)$$

where the term ν_c represents the class c population size.

Since the throughputs of closed models are not provided as inputs, obtaining a solution for closed models is more complicated than open model systems [36]. Closed systems can be solved using exact or approximate mean value analysis (MVA) techniques. MVA techniques rely on three key equations:

1. For each class, Little's law applied to the queuing network as a whole:

$$X_c(N) = \frac{N_c}{Z_c + \sum_{k=1}^K R_{c,k}(N)}$$

2. For each class, Little's law applied to the service centres individually:

$$Q_{c,k}(N) = X_c(N)R_{c,k}(N)$$

and the total queue length at centre k is:

$$Q_k(N) = \sum_{c=1}^C Q_{c,k}(N)$$

3. For each class, the service centre residence time equations for:

- a. Delay centres:

$$R_{c,k}(N) = D_{c,k}$$

- b. Queuing centres:

$$R_{c,k}(N) = D_{c,k} \left[1 + A_{c,k}(N) \right]$$

where $A_{c,k}(N)$ is the arrival instant queue length at centre k seen by an arriving class c customer.

The calculation of the term $A_{c,k}(N)$ in the above equation can be determined 'exactly' or 'approximately'. These terms refer to how the solution relates to the model and not how the solution of the model relates to the system being modelled. These two techniques differ in how the arrival instant queue lengths are calculated and are examined next.

Exact Solution Technique This solution technique of a closed model requires the *exact* determination of the $A_{c,k}(N)$ values. Once this term is evaluated then the MVA equations can be determined.

The queue length seen upon arrival to a centre is equal to the time averaged queue length at the centre with the arriving customer removed from the network. This can be expressed as: $A_{c,k}(N) = Q_k(N - 1_c)$.

Starting with the trivial solution of the network with the empty population for all centres k , an iterative approach can be applied to eventually derive the desired performance measures at a particular population size.

Unfortunately though the complexity of this approach is prohibitively high for all but the simplest of cases. For example a closed system with one job class, ten service centres and fifty customers/jobs requires ten storage locations and five hundred arithmetic operations. A model with ten service centres and five job classes, each with ten jobs, results in approximately eight million arithmetic operations and about one hundred forty five thousand storage locations. This growth in arithmetic operations is defined to be:

$$CK \prod_{c=1}^C (N_c + 1)$$

Similarly, space complexity is defined as:

$$K \prod_{c=1}^C (N_c + 1)$$

, where c does not equal $c_{\max}$, the index with the class having the largest population. This time and space complexity makes it apparent that solving a network with many classes is not practical. This realization leads to the specification of the *approximate* solution technique.

Approximate Solution Technique The solution of multiple class models can be very complex for all but the simplest of situations. An alternate approach, the approximate solution technique, addresses this issue while producing accurate performance measures [36].

The approximate solution technique is essentially similar to the exact solution technique except for the method of calculating a service centre's average queue length. As previously described, the exact solution approach calculates this value by progressing from a zero population multiclass workload to the full target population. In contrast, the approximate solution technique uses an initial guess of full population average queue length and progresses through iterations of an approximation function until successive estimates of time averaged queue lengths are successively close.

The advantage of the approximate solution technique over the exact solution technique is that relatively accurate performance measures result with substantially less time and space calculation complexity. For example, throughputs and utilizations are within a few percent, and residence time and queue lengths are within ten percent of the respective exact solutions [36].

Mixed Model Mixed queuing network models are used in cases where some job classes are closed and some are open. This type of model could represent, for example, a system with a mixture of batch and transaction workloads.

The workload intensity vector of the entire model is represented by: $I \equiv (N_1 \text{ or } \lambda_1, N_2 \text{ or } \lambda_2, \dots, N_C \text{ or } \lambda_C)$

In the evaluation algorithm for the mixed model solution technique appearing below, let $\{O\}$ be the set of open classes and $\{C\}$ the set of closed classes:

1. For each centre k , obtain its utilization by each open class:

$$U_{c,k}(I) = \lambda_c D_{c,k} \quad c \in \{O\}$$

and its total utilization by all open classes:

$$U_{\{O\},k}(I) = \sum_{c \in \{O\}} \lambda_c D_{c,k}$$

This latter equation represents the application of the forced flow and utilization laws to each open class.

2. Solve the closed model consisting of the K service centres and the closed customer classes, but no open classes. The service demand of each closed class job at each centre k in the closed model is set to:

$$D_{c,k}^* = \frac{D_{c,k}}{1 - U_{\{O\},k}(I)} \quad c \in \{C\}$$

The term $D_{c,k}$ is the service demand of class c at centre k in the original mixed model. The throughputs, queue lengths, and residence times obtained from the solution of this model are performance measures for the corresponding closed classes in the mixed model. Utilizations can be determined by applying the utilization law to the original set of service demands $D_{c,k}$

3. Residence times and queue lengths for the open classes can be computed using the performance measures of the closed classes:

$$R_{c,k}(I) = \frac{D_{c,k} [1 + Q_{\{C\},k}(I)]}{1 - U_{\{O\},k}(I)} \quad c \in \{O\}$$

$$Q_{c,k}(I) = \lambda_c R_{c,k}(I) \quad c \in \{O\}$$

where $Q_{\{C\},k}(I)$ is the total queue length of all closed classes at centre k obtained from the solution of the closed model in Step 2.

The second step of the mixed model algorithm calculates the performance measures of the closed classes of the mixed model by eliminating all open job classes and creating a model of only closed classes. The effect of the open classes on the closed class performance measures is represented by inflating the service demands of the closed classes at all devices. The inflation factor $1 - U_{\{O\},k}$

is the percentage of time that the processor is not in use by the open classes. In essence, this factor represents the effective capacity of the service centre as seen by the closed class of jobs, as some portion of its capacity is allocated to the other, open, job classes. This technique is known as *load concealment* and is a popular way for reducing the complexity of mixed class models by eliminating job classes yet accounting for their impact on the performance.

Section 4 Simulation Modelling

Simulation modelling, within the context of this thesis, refers to the software representation and performance evaluation of a computer system. The type of simulation model most useful for evaluating computer system performance is a discrete, event driven model. This class of model requires that states of the system be driven by the occurrence of events that occur at discrete, non identical, times [36].

Simulation's strength and weakness is its flexibility. From a positive perspective, any system characteristic or level of detail can be modelled. Conversely, the modelling of additional detail results in more program development effort and longer run times [36].

For example, the simulation program the author developed for modelling a computer system with approximately a dozen resources and having a multiple job class workload, a priority preemptive resume job dispatching schedule, separate waiting and interrupt queues for each job class at each resource and plentiful performance measures, required approximately ten thousand lines of simulation language code. However, reducing the level of detail in the model or adopting built-in simulation facilities — for example, assuming that each resource has a single First Come First Served queue — can reduce the amount of simulation code the analyst-programmer must develop by approximately one half.

Though simulation modelling can require extensive design and implementation effort, this technique can also provide performance results when analytical techniques are undesirable or infeasible.

Model Implementation

The workstation and workload models described in chapter three were implemented in a general-purpose, discrete-event simulation programming language, Simscript™.

At a very general level, the author's program consists mainly of simulated activities, resources and an arbitration algorithm. Simulation activities represent the actions associated with the workload model's job classes. Activities require an expenditure of time and cause a state change in the simulated system's environment. Simulation resources represent the workstation model's resources and are visited by jobs or tasks. Lastly, the simulation arbitration algorithm specifies the rules the workstation model's resources adhere to in servicing the workload model's jobs.

At a more detailed level, the program used Simscript™ "processes" to define schedulers for each of the six workload model job classes. This use of scheduler processes allowed for the run-time specification of job class activation and parameterization.

The program also used Simscript™ "entities" to identify individual jobs. Each job entity has a data set with the following self-explanatory, job-related attributes: job priority, job size, job path, job creation time and job arrival time at a resource.

Simscript™ "sets" were used to represent the queues of each resource. Sets were necessary to support the complexity of the workstation model's queues; the standard language facilities could not address these needs.

Standard Simscript™ language facilities were used to gather resource and job related statistics during the simulation run. The total number, average, standard deviation, minimum and maximum number of jobs that visited each resource and each type of resource queue were recorded. Similarly, job related statistics were gathered for birth-to-death job cycle times and job cycle times through each visited resource.

The program also included a Simscript™ "process routine" for each job class. These routines established a job's size, duration and route or path, and also specified the frequency at which job schedulers were activated. Each job class also had another "process routine" that specified the order of the resources to be visited and initiated tasks that caused job and resource statistics to be gathered.

The final major SimscriptTM facility used in describing the workstation model is a resource "routine". This type of routine, or procedure, is used to implement the arbitration scheme at each resource.

Other standard SimscriptTM facilities, such as the provided simulation clock, were used where they were consistent with the requirements of the developed models and appropriate.

Section 5 Performance Targets

The Collins concise dictionary defines a *target* as 'a fixed goal or objective'. It subsequently defines a *goal* as 'the aim or object towards which an endeavour is directed' and, cyclically, an *objective* as 'the object of one's endeavours; goal; aim'.

If a goal implicitly provides a qualitative description of a target, then an objective implicitly provides a quantitative target description. For example, the goal version of a performance target could be: *The workstation must be able to effectively service the prescribed BISDN workload.* In comparison, the objective version could state: *Peak and average utilization of each workstation resource must not exceed 85% and 75 %, respectively, of its capacity, between 8:00 a.m. and 6:00 p.m. local time.* To facilitate unbiased comparison with performance measures, performance targets will be expressed as quantitative objectives.

Objectives can be expressed in terms of *Critical Success Factors* (CSFs). As the name may suggest, CSFs represent necessary and sufficient conditions for successful attainment of the targeted objective. This approach metaphorically poses the question: 'What must occur for objective X to result ?'.

In addition to identifying the conditions for success, CSFs metamorphically focus on the 'heart of the matter'. As such, it is philosophically compatible with the abstraction process previously undertaken to model the BISDN workload and workstation. Namely, identifying important concepts and objects while ignoring minute detail. Consequently the CSF approach is well suited, and will be used, to establish workload and workstation performance target objectives.

Performance targets, as quantitative objectives, should be established for individual and aggregate workload and workstation activities. This resulting mixture of micro and macro level performance measures will facilitate interpretation and analysis of results in the next chapter. As a guide, results equal to, or less than,

targeted performance targets indicate acceptable performance; conversely, results that exceed targeted values indicate unacceptable performance.

Time and space parameters are used to represent workload and workstation CSFs. These two attributes of workload and workstation performance are appropriate CSFs for at least three reasons. First, collectively, time and space are universally considered fundamental parameters of most phenomena. Second, individually, time is an obvious performance measure for jobs that are sensitive to implicit or explicit time deadlines. For example, video or voice services are perceptibly degraded when frames or packets are delayed beyond specific deadlines. Third, individually, space is an appropriate design requirement for jobs that require buffering — e.g. jobs that are packetized or queued. As there is a major discrepancy between the multi-megabyte video frame and BISDN's ATM half kilobyte (approximately) transport cell size, there will likely be a workstation requirement for buffer space. Consequently, space is an important consideration.

The interpretation of time and space requirements will be different for workload and workstation performance objectives. For workload performance targets, time and space will relate to processing deadlines and memory requirements. For workstation performance targets, time and space will relate to percentage resource utilization and queue buffering space requirements. The next two topics discuss these ideas.

Workload Requirements

Two versions of BISDN service requirements are provided in Figures 49, [36] and 50 [36]. Figure 49 presents channel data rate and session duration information for a variety of data, audio and video job types. Figure 50 presents similar information plus job's burstiness data. This additional characteristic describes the degree of episodic, rather than continuous, activity during some interval of interest. Collectively, the Figures provide some insight into job characteristics and workload requirements.

The data in Figure 49 represents job characteristics that range from telemetry's low channel rate and short session duration, to broadband information retrieval's high channel rate and mid-scale session duration. The audio jobs in this Figure overlap longer data job sessions and middle channel data rate requirements. Video jobs have high channel data rate requirements for long sessions.

The model's high resolution job class can be associated with the Figure's entertainment video and broadband information retrieval services. Similarly the model's low resolution job class is approximated by the Figure's video teleconferencing service. Continuing in this manner results in the association of the model's voice, interactive, data and facsimile job classes with the Figure's voice, transaction and timesharing, voiceband data and facsimile services, respectively.

Figure 49 One version of BISDN service requirements [36].

The second Figure organizes several BISDN traffic characteristics into communicative (multiple direction) and distributive (one way) services. As was the case with Figure 49, there are more services identified in this Figure than there are job classes defined in the model. However careful inspection of this Figure permits an association between its traffic characterizations and some of the model's job class specifications.

The model's high resolution video job class bit rate is consistent with this Figure's high definition TV service. Similarly, the model's low resolution video job is consistent with the Figure's TV service. Completing this process results in association between the model's voice and facsimile job classes and the Figure's telephony and facsimile (Group 4) services, respectively. Figure 50 does not have services that can be clearly associated with the model's interactive and data job classes.

Figure 50 A second version of BISDN traffic [36].

SERVICES:	Bit Rate (kbit/s)		Call Duration (secs)		Burstiness	
	Min	Max	Min	Max	Min	Max
COMMUNICATIVE SERVICES:						
Dialog						
Telephony		64		100	0.3	1
Videotelephony	64	70,100		100	0.5	1
Telemetry	0.01	10	1	10	0.01	0.1
Telex		10		100		1
Facsimile (Group 4)		64	10	100		1
Videosurveillance	10	30,100	1,000	10,000	0.1	1
Videoconference	1,000	70,100	1,000	10,000	0.5	1
Retrieval						
Videotex	1	64	100	1,000		0.1
Broadband videotex	1,000	70,100	10	1,000	0.1	1
Messaging						
Voice mail	16	64	10	100	0.3	0.5
Video mail	1,000	70,100	10	1,000	0.1	0.5
Electronic mail		10		100		0.5
DISTRIBUTIVE SERVICES						
HiFi sound		768	1,000	10,000		1
TV	30,100	70,100	1,000	10,000		1
High definition TV	140,100	565,100	1,000	10,000		1

Combining the information from these two Figures with the descriptions of job classes from the previous chapter results in the following interpretation of job class characterizations and performance objectives.

High Resolution Video A high resolution video job is represented by a stream of video frames. The default high resolution video frame size consists of megabits of uncompressed data. Applying standard video compression techniques will reduce the amount of data being transmitted by at least one third. The workstation must service a high resolution video frame in no more than one sixtieth of a second. Failing to do so may lead to traffic congestion , data loss or visual discontinuity.

A high resolution video job is assumed to be active for the entire time interval under investigation. That is, a job is “activated” at the start of the evaluation period and “destroyed” at the finish of the period. This assumption was made to ensure that the continuous characteristic of video jobs was fully accounted for in the evaluation of peak load workstation performance. Since this job class is assigned the highest preemptive service priority, this assumption has a potentially interesting effect on the service times of all other jobs.

Specifically, the amount of service time available at each resource along the high resolution video job’s route is reduced for the entire evaluation period by the amount of time required to process this high priority job. For example, the visit of one high resolution video job to the system bus will leave one second less the high resolution video job’s service time per every second of the evaluation period, for other jobs wanting service at the bus. The significance of this is it will tend to magnify any congestion problems that may arise within the model at resources that are visited by multiple job classes.

Summarizing these high resolution video job characteristics in the form of performance objectives results in the following:

1. Source to sink workstation service time per frame of data to be less than or equal to one sixtieth of a second; and
2. A minimum buffering size of one frame per active job for each workstation along this job class route.

Low Resolution Video A low resolution video job has service requirements and performance objectives that are similar to those of a high resolution video job.

The low resolution video job is also represented by a stream of video frames. The default high resolution video frame size consists of kilobits of uncompressed data. Applying the video telephony compression techniques will reduce the amount of data being transmitted by at least one third. The workstation must service a low resolution video frame in no more than one thirtieth of a second. Failing to do so may again lead to traffic congestion , data loss or visual discontinuity.

A low resolution video job is assumed to be active for the entire time interval under investigation. That is, a job is "activated" at the start of the evaluation period and "destroyed" at the finish of the period. As previously, this assumption was made to ensure that the steady stream nature of a low resolution video job is fully accounted for in the evaluation of peak load workstation performance. Since this job class is assigned the second highest preemptive service priority, this assumption has a similarly interesting effect on the service times of all other jobs.

The amount of service time available at each resource along the low resolution video job's route is reduced for the entire evaluation period by the amount of time required to process this high priority job. The visit of one low resolution video job and one high resolution video job to the system bus will leave a one second less their collective service times, per every second of the evaluation period, for other jobs wanting service at the bus. The significance of this is it will tend to continue to magnify any congestion problems that may arise within the model at resources that are visited by multiple job classes, especially where relatively high priority job class routes overlap.

Establishing these low resolution video job characteristics as performance objectives results in the following:

1. Source to sink workstation service time per frame of data to be less than or equal to one thirtieth of a second; and
2. A minimum buffering size of one frame per active job for each workstation along this job class route.

Voice Though a voice job is aural rather than visual, it shares a conceptual similarity to the two previously discussed video jobs. A voice job is represented by a stream of voice packets. The default voice packet size consists of kilobits of uncompressed data. Applying telephony coding and compression techniques can reduce the amount of data being transmitted by approximately one half, three

quarters or more. The workstation must service a voice packet in no more than one sixty-fourth of a second. Failing to do so may again lead to traffic congestion, data loss and a jittery conversation.

A voice job is assumed to be active for the entire time interval under investigation. A job is “activated” at the start of the evaluation period and “destroyed” at the finish of the period. This assumption was made to ensure that the bursty characteristic — “talk spurts” — of a voice job is fully accounted for in the evaluation of peak load workstation performance. Since this job class is assigned the third highest preemptive service priority, this assumption has an effect on the service times of all other jobs that is a straight forward extension to the one described for the high and low resolution video jobs.

Accordingly, the performance objectives for the voice class of jobs are:

1. Source to sink workstation service time per frame of data to be less than or equal to one sixty-fourth of a second; and
2. A minimum buffering size of one frame per active job for each resource centre along this job class route.

Interactive The interactive job class is the first job class described in this chapter that does not have a definitive service deadline. This job *can* theoretically wait any amount of time without violating any time-sensitive service requirements; whether a user of the interactive service *will* wait, of course, is another matter.

As indicated in literature [36], users of computer systems suffer from decreased productivity and increased anxiety as their interactive response time creeps beyond one second. The author postulates that subsecond interactive response time should be targeted for peak load conditions, and that three hundred millisecond to five hundred millisecond interactive response time should be targeted for non peak load states. This position is based on the assumption that the BISDN user population will have experienced subsecond interactive response time, possibly on a personal computer, and will hesitate or refuse to use BISDN applications on a workstation that delivers slower than previously experienced response time. Succinctly restated: ‘there’s no going back’.

An interactive job is represented by a stream of data packets. The default interactive data packet size consists of kilobits of uncompressed data. It is assumed that two major reasons for compressing data — faster network travel and more reliable transmissions — do not significantly apply to this type of job (i.e. one that

is predominantly consists of relatively small packets confined to intra-workstation travel).

An interactive job is assumed to be active for the entire time interval under investigation. This assumption permits the use of interactive service time as a continuous, albeit crude, barometer on workstation performance. Thus even if the workstation is able to effectively service the higher priority video and voice jobs, a resulting decoloration in interactive job service times could indicate unsatisfactory performance. Frequently in practice, interactive response time provides instantaneous feedback on how “well” your workstation is managing its workload.

Restating this interactive job class information, and conjecture, in the form of performance objectives results with the following:

1. Source to sink workstation service time per interactive packet to be less than or equal to one second under peak load conditions;
2. Source to sink workstation service time per interactive packet to be less than or equal to five hundred milliseconds under non peak load conditions;
3. A minimum buffering size for interactive packets at each visited service centre can not be determined with the available information. Assume, for heuristic reasons, that common current personal computer cache sizes are applicable. Hence, visited resources require a minimum interactive job related buffer space of sixty-four kilobytes.

Data A data job is represented by a stream of data packets. The default data packet size is in the range of bytes to kilobytes. It is assumed that standard data compression algorithms can be applied to reduce the amount of data being transmitted by approximately one third.

Due to its relatively low priority, a workstation does not have a deadline for servicing a data packet. The real possibility of data packet transmission being controlled by some form of communications handshaking is ignored in the model as it was considered outside the scope of the work.

A data job's arrival pattern and session duration time is assumed to be characterized by Poisson distributions.

Thus the performance objectives for a data job are:

1. Source to sink workstation service time per data job packet is not determinable from the above information. Assume that jobs that are not completed after

one hour will be terminated by some authority and automatically rescheduled after normal hours of operation (e.g. say, three a.m.); and

2. A minimum buffering size is not determinable from the above information. Assume that no more than thirty megabytes of buffering is available for data jobs in the entire system. This value is approximated from a sixty-four kilobit basic service being used for one hour.

Facsimile A facsimile job is represented by a stream of packets adhering to Group 4 standards. The default facsimile packet size is measured in kilobytes and can be compressed.

Facsimile jobs have the lowest priority. The workstation does not have a deadline for servicing a facsimile packet. Conceptually, facsimile jobs should provide comparable service to current fax add-on boards to personal computers, or stand alone facsimile machines. Therefore an implicit deadline of “acceptable” servicing can be assumed to be in the order of minutes.

A facsimile job’s arrival pattern and session duration time is also assumed to be characterized by Poisson distributions.

The performance objectives for a facsimile job are:

1. Source to sink workstation service time per data job packet is not determinable from the above information. Assume that jobs that are not completed after one hour will be terminated by some authority.
2. A minimum buffering size is not determinable from the above information. Assume that no more than thirty megabytes of buffering is available for data jobs in the entire system. This value is approximated from a sixty-four kilobit basic service being used for one hour.

Workstation Requirements

As stated in the section’s introduction, workstation requirements can be expressed in terms of the proportion of *time* a resource centre is busy and the amount of buffer *space* required. The first issue is addressed by classical queuing theory; it is assumed that the second may be determined by aggregating individual job class space requirements at each resource centre.

Time Classical queuing theory indicates that until a threshold level is reached, system performance is mildly affected by small changes in the workload it is

servicing. Once this threshold is exceeded, a minor increase in the offered workload results in a disproportionate decrease in system performance. This effect is depicted in Figure 51.

Figure 51 The effect of a workload increase on user response time.

2 of 2

PM-1 3 1/2"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

PRECISIONSM RESOLUTION TARGETS

PIONEERS IN METHYLENE BLUE TESTING SINCE 1974

15800 COUNTY ROAD 5, SUDBURY, ONT. N3B 1G4
TEL: 917 426 7087 FAX: 917 426 7087 TLA 9100000000

Stallings indicates that system performance collapses once the proportion of a systems's busy time exceeds eighty to ninety per cent of its capacity [37]. Clearly, more desirable system performance may be possible if system busy time is less than eighty per cent of system capacity. Based on the approximate data in Figure 51 and heuristics, assume that system capacity must not exceed seventy-five per cent of each service centre's capacity. This behaviour was observed in the simulation results.

Space The amount of buffer space required at each resource service centre is approximated by interpreting the previous chapter's modelled workload information and the previous section's workload requirements. Figures 52 and 53 summarize key aspects of this information. The former provides job class routing while the latter lists job size and duration information.

Figure 52 Service centres visited by each job class.

Service Centres	Job Classes					
	Facsimile Job	Data Job	Interactive Job	Voice Job	Low Resolution Video Job	High Resolution Video Job
BUS	✓	✓	✓	✓	✓	✓
CPU		✓	✓			
DIGITIZER	✓					
DISK	✓	✓	✓	✓		
FACSIMILE BOARD	✓					
GRAPHICS MEMORY	✓		✓		✓	✓
HIGH RESOLUTION VIDEO BOARD						✓
VERY HIGH SPEED HIGH RESOLUTION VIDEO MODEM						✓
IOPORT	✓	✓	✓			
LOW RESOLUTION VIDEO BOARD					✓	
HIGH SPEED LOW RESOLUTION VIDEO MODEM					✓	
PROGRAM MEMORY		✓	✓			
VOICE BOARD				✓		

Figure 53 Job size and activation rate.

Job Class	Job Size (kilobits)	Job Duration
Facsimile	64	Stochastic
Data	64	Stochastic
Interactive	64	Continuous, every second
Voice	64	Continuous, every 1/64 second
Low resolution video	2,621	Continuous, every 1/30 second
High resolution video	49,152	Continuous, every 1/60 second

An estimate of the bus service centre's buffering requirement is given in Figure 54. *This estimate assumes the bus throughput is sufficient to avoid congestion.* The buffer space is identified to hold one high resolution video frame, one low resolution video frame and multiple packets from each of the non-video jobs. If it is reasonable to assume that the group of non-video jobs require one megabyte of buffer space, then total bus buffer space may be approximated as eight megabytes.

Figure 54 Calculation of minimum buffer space requirements for the bus service centre

Bus Buffer Size Requirements

Bus Buffer Space $\approx$ one high resolution video frame + one low resolution video frame
+ W voice packets + X interactive packets
+ Y data packets + Z facsimile packets.

$$\approx 49,152,000 + 2,621,440 + \{(W + X + Y + Z) 65,536\} \text{ bits}$$

$$\approx 51,773,440 + \alpha \text{ bits}$$

Assume α = one megabyte, then Bus Buffer Space $\approx 6.171875 + 1.0$ MB

$$\approx \lceil 7.171875 \rceil \text{ MB}$$

$$\approx 8 \text{ MB}$$

The one megabyte buffer assumption for non-video jobs is based on the following argument. First, observe that high and low resolution video frames are orders of magnitude larger than the packet size of the non-video jobs. As

a consequence, the non-video jobs will have little affect on space requirements unless hundreds or thousands of packets are buffered. Second, if thousands of non-video packets are buffered, the potential service delay inferred from this number of non-video packets would likely make coherent voice conversation and interactive job servicing impractical. This leaves the bulk of the non-video buffer space for the relatively time-insensitive data and facsimile jobs. While these jobs could conceivably consume this amount space, it would represent maximum session durations as indicated by Figures 21 and 25 in chapter three. Consequently, assuming one megabyte of buffer space for non-video jobs may be fairly reasonable.

To summarize, given these assumptions and arguments, the bus buffer space requirement is estimated to be eight megabytes. The buffer space requirements for remaining resources can be estimated in a similar fashion and these estimations are summarized in Figure 55. For the sake of completeness, the bus buffer space requirement is included in the list.

Figure 55 Buffer space requirements for system resources

Service Centres	Buffer Requirements
BUS	8 MB
CPU	1 MB
DIGITIZER	1 MB
DISK	1 MB
FACSIMILE BOARD	1 MB
GRAPHICS MEMORY	8 MB
HIGH RESOLUTION VIDEO BOARD	1 MB
VERY HIGH SPEED HIGH RESOLUTION VIDEO MODEM	6 MB
IOPORT	1 MB
LOW RESOLUTION VIDEO BOARD	1 MB
HIGH SPEED LOW RESOLUTION VIDEO MODEM	1 MB
PROGRAM MEMORY	1 MB
VOICE BOARD	1 MB

The buffer *space* estimates that appear in Figure 55 are, of course, exclusive of graphics and program memory *resource* requirements. These estimates are now examined.

The initial basis of the graphics memory requirement is its buffer size. That is, to simultaneously display high and low resolution video class jobs, a minimum of one frame of each these types is necessary. This value was previously determined to be approximately 6 Megabytes.

To this amount, more memory could be added to provide additional graphic capabilities. For example, concurrent execution of two high and low resolution video jobs or memory bank interleaving could double the initial amount. Alternately, support for higher resolution monitors could increase the graphics memory requirement from twelve (i.e. $2048 \times 2048 \times 24$) to seventy-two megabytes (i.e. $4096 \times 4096 \times 36$). Within this range, an initial configuration of twenty megabytes may be reasonable.

The final space requirement to be estimated belongs to the program memory resource. The sizing of this argument is partially influenced by goals of having each of the six job class types simultaneously memory residence and of minimizing the utilization of system resources by memory management system tasks.

The former might limit the amount of paging and/or swapping performed; this could potentially reduce time delays associated with job scheduling and memory management activity. The later might limit the amount of system traffic and hence could contribute to improved system performance through reduced contention for crucial (i.e. program memory and bus) resources.

The amount of program memory is nearly impossible to accurately predict given the current vagueness of BISDN application detailed requirements. However gross ball park approximations may be warranted as a basis for planning and collegial discussion.

The first point to consider is that popular workstations in the Department are configured with twenty megabytes of memory. It seems unlikely that any less memory would be required in a BISDN workstation.

Second, if historic trends described in the background chapter can be applied to a future period in time, then workstations in a decade's hence may be configured with a tenfold increase in memory. Using the above value of twenty megabytes as a current value, this results in a BISDN era memory configuration of approximately two hundred megabytes.

This memory estimation is difficult to defend objectively. However it is arguable as a representation of the correct magnitude of the required program memory configuration based on projections from historical system growth.

In summary, workstation time and space requirements were approximated in the previous material. In developing the individual workstation targets, it is apparent that the time, or utilization, objectives could be determined in a manner consistent with classical queuing theory. On the contrary, space requirements were more arbitrary and dependent on more assumptions. For this reason, time objectives are of paramount importance, and space requirements are ancillary, in the evaluation of the modelled workstation's performance.

This section examined the workload and workstation time and space objectives and requirements. Workload service time objectives are of primary importance to evaluating model performance. This is especially the case considering some queuing models assume infinite resource queues. In this light, the discussion of space requirements should be seen as a design consideration rather than an evaluation criterion.

This chapter presented substantial breadth and localized depth to the description of computer system modelling and performance evaluation. It began with a historical perspective that traced some major developments from the early application of single server queuing theory to the subsequently more complex and realistic application of separable queuing networks. This was followed by sections dealing with queuing network modelling and solution techniques. A discussion of simulation modelling analysis and performance targets completed the chapter's contents.

CHAPTER 5 METHODOLOGY

This chapter describes the method used to derive performance data and test the hypothesis. In essence, the broader methodology of testing the hypothesis started with the development of the workstation and workload models in chapter three. Chapter four then identified simulation as the most appropriate evaluation technique and provided details of the simulation program written in SimscriptTM. This chapter describes how the results from the simulation program were obtained.

As mentioned in the "Simulation Modelling" section of the previous chapter, several statistical measures of resource visitation and utilization, and job related birth-to-death and intra-resource cycle, or execution, times were captured. Results were obtained by holding the workstation's resource parameters constant, at the default rates specified in the "Resource Parameters" section of chapter 3, and by varying the workload's job activation and size parameters.

Each change in the workload parameters resulted in a separate simulation run. In turn, each program run represented the execution of a particular workload combination for a period of eight simulated hours. At the end of each program run, performance statistics were produced. The performance results from several runs were compared and analyzed to determine if the workload could be effectively serviced and if any particular resource stood out as a system bottleneck. The workload parameters and individual simulation runs are described next in sufficient detail to make it possible to repeat the experiments and arrive at the same results.

The first simulation run activated a facsimile job scheduler only. The second simulation run activated both a facsimile and data job scheduler. The third run activated a facsimile, data and interactive job scheduler. The fourth, as you may anticipate, activated a voice job scheduler in addition to the three previously mentioned schedulers. In each of these simulation runs, the increase in the simulated workload resulted from the activation of a higher priority job scheduler rather than more jobs of the same priority. In this way, the marginal effect of adding higher job classes to the workload mix could be examined.

The fifth simulation run activated the four previously mentioned schedulers plus a low resolution video job scheduler. The low resolution video job was specified as a 525 kbit per frame, 15 frame per second, workload. This low

resolution video job represented a 256 x 256 x 8 bit frame, consistent with ISDN experiments reported earlier in chapter two. Resources specifically dedicated to processing low resolution video jobs — the low resolution video board and the high speed modem communications interface — were adjusted to be compatible with the low resolution video job size.

Subsequent simulation runs specified that the facsimile, data, interactive, voice and low resolution video job schedulers remain active. In these runs, the job parameters for all but the low resolution video jobs were held constant. The low resolution video job's refresh rate was increased to the more common thirty frames per second and the low resolution video job size associated with each frame was increased. As with the original low resolution job, the resources specific to processing low resolution video jobs were adjusted to remain compatible with the frame size. Low resolution video job sizes of approximately 220 Mbps were investigated.

The low resolution video job size parameter was not increased by a constant amount. This was done as it became evident that either relatively small or large low resolution job sizes had comparatively marginal impact on system performance. In the case of the former, service times and utilization of resources were not significantly affected. In the case of the latter, service times and resource utilization were already problematic. Consequently, more gradual changes in selecting the low resolution video job size were made in the most sensitive regions, namely below the 220 Mbps range.

After these main results were obtained, the "technology factor" parameter the author included in the developed SimscriptTM program, was increased to reflect the impact of technological change. This "factor" was implemented as a multiplier that is applied to all of the workstation's resources.

In summary, the information presented above that describes the workload parameters, the information in chapter four that describes the developed SimscriptTM simulation program and the information in chapter three that describes the resource parameters and the workstation and workload models is sufficient and necessary to reproduce the author's results presented in chapter six.

CHAPTER 6

RESULTS

This chapter describes results obtained from more than 20 runs of a 10,000 line SimscriptTM simulation program. The simulation results are believed to be accurate as manually derived results for a very simple workload — a single job class that was active for a very short period of simulated time — were identical to the results from a simulation run with the same workload parameterization.

In general, the results indicated that a workload consisting of facsimile, data, voice, interactive and low resolution video jobs was sufficient to cause a bottleneck in the modelled workstation. Specifically the system bus became congested at a level that precluded the activation of a high resolution video job scheduler. This result affirms the thesis' objective of demonstrating that a current state-of-the-art conventional personal computer would be unable to effectively service a BISDN application workload.

Additionally, five simulations were run with a parameter adjusted to represent a doubling of technological capacity in all system resources. The results, though not plotted, were predictable. A pattern similar to the original runs resulted except the workload required to saturate the system approximately doubled. Given this increase, it was still impossible to activate a high resolution video job without incapacitating the system.

As described in the previous chapter, the initial workload consisted of a facsimile job. The results from this simulation run indicated that the workload was effectively serviced by the workstation model. Resource utilization, resource queue buildup and job cycle times had acceptable values with little variation. For example, the average cycle time for a facsimile job was approximately 1 second and the maximum cycle time was approximately 1.5 seconds. The total time a facsimile job spent waiting for system resources to be available was approximately 1E-15 seconds. The most heavily utilized system resource was the facsimile board at approximately 22%. The digitizer was utilized less than 3% and all other resources were utilized less than 1%.

The result of the next three simulation runs —the activation of facsimile and data jobs, the activation of facsimile, data and interactive jobs, and the activation of facsimile, data, interactive and voice jobs — was that the workstation was able

to effectively service this workload without significantly affecting resource utilization, resource queue buildup or job cycle times. In particular, the introduction of interactive and voice jobs had virtually no effect on the average and maximum cycle times of the facsimile and data jobs. Similarly, the introduction of voice jobs had only a 2 millisecond impact on the cycle time of interactive jobs. At this workload level, the bus was utilized less than 1%. Jobs visited the bus interrupt and waiting queues, but only utilized them less than 3E-7%.

A presence of jobs in the interrupt and waiting queues — however brief — indicates that jobs competed for service at the bus resource and that higher priority jobs interrupted in-service jobs. Thus even with a relatively light workload, a pattern of jobs contending for service from the bus and in-service jobs being interrupted, had begun.

Low resolution video jobs were subsequently added to the workload. Initially, servicing of the lower priority jobs was not significantly affected by this change in workload. The average and maximum cycle times for facsimile and data jobs were unaffected. The average cycle times for interactive jobs increased by approximately 2% and the average cycle times for voice jobs increased approximately 6%. However the maximum cycle times for interactive and voice jobs, while still representing effective service levels, increased by approximately 40%.

At this workload level, the system bus was utilized less than 4%, but the bus interrupt and waiting queues were more utilized than previously — though still less than 1%. Other system resources were less utilized than the bus except for facsimile and low resolution job-specific resources, which have capacities dependent on their respective job sizes.

This pattern of marginally slower cycle times and relatively low resource utilization continued until the low resolution video job size began dominating the system's bus resource. For example, a low resolution video job size of approximately 2600 kbits per frame or 78 Mbps, resulted in bus utilization of 39%. This workload also resulted in the cycle time for voice job being slowed down sufficiently to lose one voice packet every second. Though this loss could be avoided through the use of a voice compression algorithm.

A continued increase in the size of the low resolution video jobs had predictable results. A low resolution job size of 6 Mbits per frame, or 180 Mbps resulted in the system bus being utilized 84%. The average cycle times for fac-

simile and data jobs were a few seconds. The average cycle time for interactive jobs was still subsecond, and for voice jobs, the average cycle waiting time indicated that approximately 6 voice packets per second would be lost. While standard voice compression algorithms could address this issue, the maximum cycle waiting time amounted to 3 seconds making any semblance of conversation impossible. Similarly, the average and maximum cycle waiting times for low resolution video jobs were also prohibitively long at this workload level. The system was clearly saturated.

The results described above are illustrated in the figures that follow. Each of these plots represents the effect of the increasing low resolution video workload, expressed as a percent of bus capacity, on job cycle time.

Figure 56 plots the maximum cycle times for each job class. It illustrates that as the offered workload approaches seventy percent of the system bus capacity the maximum amount of time each job waits for service dramatically increases. While these maximum cycle times may be non-critical for facsimile and data jobs, the times are oppressive for interactive, voice and — as illustrated in Figure 62 — low resolution video jobs.

Figure 56 Maximum cycle times.

The average cycle time plots illustrated in Figure 57 present a similar, but more gradual, pattern. In general, the cycle time values for each job class are approximately half the maximum values illustrated in Figure 56. Additionally, a workload representing a larger — approximately, eighty — percent of the bus capacity is required before average cycle times dramatically increase.

Figure 57 Average cycle times.

A clearer illustration of how each job class is affected by increases in the offered workload is shown in the next four figures. Each of these figures contains plots of both average cycle time and average wait time for a specific job class. Figures that have relatively less vertical distance between these two plot lines, than there is between the base axis and the average wait time plot, would indicate that on average a job spends more time waiting for service than being serviced. A relatively high average wait time would indicate that the job class is being blocked and forced to wait due to some resource bottleneck.

Similarly, a figure with relatively less vertical distance between the base axis and a job's average wait time plot line than between the average cycle and wait time plot lines, would indicate that the majority of a job's cycle time is involved in being serviced. A relatively low average wait time would indicate that the job class is not experiencing much blockage and is not being forced to wait for service.

Figure 58 plots average cycle and wait times for a facsimile job against an offered workload that is expressed as a percent of bus capacity. It indicates that

for workloads up to approximately seventy percent of the bus capacity, both plots are relatively flat and unaffected by the workload. Accordingly, nearly all of the average cycle time represents service time instead of wait time. However beyond this level, both average cycle and average wait times increase substantially. Thus the increase in the average cycle time is attributable to increased waiting time, which in turn indicates the modelled system is unable to service this workload.

Figure 58 Average cycle and wait times for a facsimile job.

Figure 59 plots the average cycle and wait times for a data job against an offered workload expressed as a percent of bus capacity. It indicates that nearly all of the average cycle time represents average waiting time. This pattern is consistent at all offered workloads. The plots are reasonably flat at approximately the four second mark for workloads below seventy percent of the bus capacity, with elevated average cycle and wait times beyond this workload level.

Figure 59 Average cycle and wait times for a data job.

Figure 60 plots average cycle and average wait time for interactive jobs against the offered workload expressed as a percent of bus capacity. Both plots gradually increase as the workload approaches seventy percent of bus capacity, then increase dramatically beyond this level. The average cycle time is initially composed of only service time, however as the workload increases the average wait time component of the average cycle time also increases. Beyond this seventy percent level, the interactive job becomes blocked and isn't serviced.

Figure 60 Average cycle and wait times for an interactive job.

Figure 61 plots average cycle and average wait time for a voice job against the offered workload expressed as a percent of bus capacity. This figure indicates that for workloads up to approximately seventy percent of bus capacity that average cycle time increasingly involves more average waiting time. Beyond this level, the average cycle time dramatically increases as a result of a similar rise in the average wait time. This indicates that voice jobs can become excessively delayed and not effectively serviced beyond the seventy percent bus capacity level.

Figure 61 Average cycle and wait times for a voice job.

Figure 62 plots average cycle and average wait time for a low resolution video job against the offered workload expressed as a percent of bus capacity. For workloads up to approximately seventy percent of bus capacity, this figure indicates that the average wait time for a low resolution video job is sufficiently low to permit effective servicing of a low resolution video frame. However beyond this level of workload capacity, the average wait time exceeds the frame refresh rate and hence indicates that the system is unable to effectively service the job.

Figure 62 Average cycle and wait times for a low resolution video job.

The next three figures, respectively illustrate bus server, bus interrupt queue and bus wait queue utilization. Only bus data is plotted as the other system resources were consistently, relatively lightly utilized and had shorter waiting queues. For example, when the bus resource's utilization exceeded 80%, the remainder of the resources — except for the low resolution video modem and video board interface — were utilized less than 10%. The bus conspicuously stood out as a major bottleneck to system performance.

The little "nook" near the top of the plot in Figure 63 can be explained by a common characteristic of simulations that can produce unreliable results near model saturation levels.

Figure 63 Average bus server utilization

Figure 64 illustrates the average and maximum number of jobs waiting in the bus interrupt queue. The top plot illustrates that the average number of jobs waiting in the bus interrupt queue increased as the workload increased to approximately 70% of the bus capacity. Beyond this level, the average number of jobs waiting in the bus interrupt queue decreased, increased and then decreased. This relatively abrupt change in average values can likely be explained by erratic simulation results occurring near model saturation levels. However in absolute terms, on average no more than one fifth of a job waited in the bus interrupt queue.

Figure 64 also illustrates that for the majority of workload levels, no more than two jobs waited in the bus interrupt queue. A near bus saturation level of workload was required for the number of jobs waiting in the bus interrupt queue to increase to three.

Figure 64 Average and maximum number of jobs waiting in the bus interrupt queue

Figure 65 illustrates the average and maximum number of jobs waiting in the bus waiting queue. The average number of jobs waiting — jobs that have not yet been serviced at this visit to the resource — remained near zero until the offered workload reached 70% of the bus capacity. Beyond this workload level the average number of jobs waiting for service by the bus rapidly increases. A similar but more extreme pattern is illustrated in the maximum number of waiting jobs plot. For example, more than 1500 jobs wait for service when the workload represents 90% of the bus resource capacity.

Figure 65 Average and maximum number of jobs waiting in the bus wait queue

As previously mentioned, high resolution video jobs could not be added to the modelled workload. The reason for this was that a single job would simply saturate the current system. This can be easily observed by considering a relatively minimal high resolution video bandwidth requirement, say 503 Mbps — representing a 1024 x 1024 x 8 bit image refreshed 60 times per second. The modelled system bus, with a resource capacity of 320 Mbps, could not service this load. Considering that, on average, jobs visit the bus 1.5 times during their travel through the system, effective bus capacity is reduced to 214 Mbps. This condition magnifies the situation.

If future technological developments warranted a doubling of current resource capacities, then the minimal high resolution video job of 503 Mbps from the previous example would still be unable to be serviced. A more demanding high resolution video job — a 2048 x 2048 x 24 bit image refreshed 60 times a second — represents 6 Gbits of data per second. The current modelled bus capacity would need to increase 25 times to address this requirement. An alternative to

this intrinsically conventional approach to handling more demanding requirements is one of the topics discussed in the next chapter.

In summary, the results demonstrated that the modelled workstation was unable to effectively support a workload with a high resolution video component (e.g. a BISDN multimedia application involving high resolution video). Inspection further demonstrated that technological advances alone would not, necessarily, be sufficient to address these broadband requirements. Lastly, the results indicated that the system bus was the bottleneck to effective servicing of the offered workload. These three observations permit the conclusion that a conventional workstation would not effectively service a BISDN workload with a high resolution component. Five doublings of technological increases would be necessary for the current workstation model to support the modelled workload.

CHAPTER 7

DISCUSSION

This chapter discusses the significance of the workstation and workload models and of the reported results. It additionally introduces the idea that a parallel system architecture could be considered for addressing BISDN application requirements.

The workstation and workload models were written in SimscriptTM. The program consists of approximately 10,000 lines of code — a size, that in retrospect, might have been halved. This conclusion comes from the following four observations related to determining an appropriate level of model abstraction.

First, the level of detail incorporated into the workload's job routing options may not have been necessary. While the detail was intended to provide a realistic environment, possibly the routing options could have been restricted to a few major choices. In addition to increasing program size and complexity, these options might simply represent a level of detail that did not provide a proportionate contribution to the information gathered.

Second, the author's priority based, preemptive resume job scheduling algorithm was infrequently used, except under congestion conditions. Although SimscriptTM does not provide such a built-in scheduler, the scheduler it does provide, in hindsight, might have been sufficiently sensitive to the models' workload and workstation characteristics.

Third, the number of job classes in the workload model might have been reduced without significantly compromising the results. Perhaps three jobs classes would have been sufficient — a relatively time-insensitive background or batch job, a time-sensitive interactive job and a time-crucial video job.

Four, regardless of the accuracy and validity of a developed model, a change in a system component's capability can have a dramatic effect on the results being reported. For example, a change in bus capacity from 320 Mbps to 640 Mbps might have more impact on a simulations results than might the expansion of the workload model from n to $2n$ job classes.

These previous four points were decidedly more obvious during and near the end of the thesis process than they were near its beginning and middle.

The simulation results are believed to be reasonably accurate. This claim is based on tracing workloads as they navigate the workstation model, tracing the event changes within Simscript'sTM simulation infrastructure, duplicating some simulation results manually, and applying some theoretical results.

The results described in the previous chapter indicate that the workstation model, with current personal computer components and subsystems, would be unable to effectively service the workload model representing, BISDN applications. It additionally noted that the workstation model's components would need to increase their current capabilities by 20 to 25 times to effectively service this load.

An alternative to relying on an implicit *technological fix* strategy, is to replace the IAS or conventional architecture workstation with one having a parallel architecture. This approach is appealing because it can focus directly on the system bottleneck by providing multiple bus connections to system resources and the interfaces to the external world. It is also appealing because a parallel architecture may be more consistent with the envisioned BISDN environment (i.e. integrated services) and applications (e.g. multimedia).

The workstation might be at any location along a parallel continuum. At one extreme, a simple second bus configuration could be employed. This architecture could have all resources connected to the two bidirectional buses. Job scheduling could be on a contention basis. A variation of this could implement two unidirectional buses — analogous to FDDI or DQDB MAN topologies. Another variation could have the workstation with three buses — two for high priority, time-crucial jobs, and the third for other data — a quasi 2B + D basic rate ISDN approach. Ultimately, the number of buses present in the architecture could increase until a full $m \times n$ matrix switch could interconnect all system resources.

However the author contends that no more than four buses would be required to effectively service the requirements of BISDN applications. This claim is based on the two assumptions. First, that all buses will be available simultaneously to a particular resource. This would permit a resource to, in effect, multiplex its workload to the intended destination. In an ATM/SONET BISDN environment, this packetization and prioritization of packets could be possible. The second assumption is that technological advances result in a doubling of system capabilities before BISDN services are available.

In summary, this chapter began with a discussion of the model of an IAS

or conventional architecture workstation and its BISDN workload. The author observed that, in hindsight, certain modelling decisions may have resulted in additional complexity without having a significant impact on the results. Especially, for example, if program features are compared to the potential impact of a change in a component's capacity specification. However, the author is confident that the workstation model is accurate, for the reasons given.

Subsequent to a discussion of the current workstation model and the reliability of the results, a general description was provided for a multiple bus workstation architecture. This architecture can be referred to as a parallel personal computer, or PPC. While any increase in the number of buses would represent an improvement in performance over the modelled workstation, a system with four buses connecting all resources was selected as being sufficiently suitable for effective BISDN application servicing. This configuration was influenced by an assumption that technology may double its capabilities prior to the availability of BISDN.

CHAPTER 8

CONCLUSIONS AND FUTURE RESEARCH

The potential to provide 150 Mbps to 600Mbps BISDN services has lead the BellCos to introduce new technologies and protocols — optical fibre, SONET and ATM. In contrast, research into workstations for delivering integrated digital services has remained fixated on the IAS architecture.

This thesis has demonstrated, through the development of simulation workstation and workload models, that BISDN services can not be effectively serviced now, and not in the future unless technological advances result in a 20 to 25 time increase in subsystem capabilities.

Having demonstrated that an IAS or conventional architecture was an unlikely architecture for satisfying BISDN application requirements, a candidate parallel architecture was introduced that consisted of four buses interconnected to all resources. Assuming modest technological advances in between now and when BISDN is introduced, such an architecture could satisfy BISDN application requirements.

Future research could examine the same issue from a few different perspectives. First, the results derived from the simulation approach utilized in this thesis could be compared with models solved using analytical evaluation techniques. This would establish or contest the accuracy of the results, interpretations and conclusions of this work.

Second, a standardized workload could be established by the Department to encourage the portability and cross referencing of Department research into BISDN communications and services. This activity could provide a common focus for research group intercommunications and for linking the various research activities of the Department.

Third, detailed identification of the requirements for and the specifications of a parallel personal computer to address BISDN applications could be researched.

Four, building on the above research point, research into a multi-tasking, real time operating system for a parallel personal computer could be undertaken.

Five, research into BISDN applications and user interfaces directed at multimedia applications could be examined.

Six, as BISDN facilities and components begin to be developed, a prototype parallel workstation could be designed and built. The three previous research future activities would logically culminate in this sixth topic — the prototype station.

BIBLIOGRAPHY

- [1] B. Avi-Itzhak and D.P. Heyman. Approximate queueing models for multiprogramming computer systems. *Operations Research*, 21, 1973.
- [2] J.J. Bae and T. Suda. Survey of traffic control schemes and protocols in atm networks. *Proceedings of the IEEE*, 79(2), 1991.
- [3] T.I. Bajenescu. High speed optical man integrating voice, video and data switching. In *IEEE International Conference on Communications*, 1990.
- [4] F. Baskett, K.M. Chandy, R.R. Muntz, and F. Palacios. Open, closed and mixed networks of queues with different classes of customers. *Journal of the Association for Computing Machinery*, 22(2):248–260, 1975.
- [5] J. Bourne, M. Balmes, and J. Justice. Heathrow —[] experience and evolution. In *IEEE International Conference on Communications*, 1990.
- [6] R.M. Bryant, A.E. Krzesinski, M.S. Laksmi, and K.M. Chandy. The mva priority approximation. *Association for Computing Machinery Transactions*, 2, 1984.
- [7] L.P. Buzen. Computational algorithms for closed queueing networks with exponential servers. In *Journal of the Association for Computing Machinery*, volume 10, 1973.
- [8] W. Byrne, T. Kilm, B. Nelsona, and M. Soneru. Broadband isdn technology and architecture. *International Journal of Digital and Analog Cabled Systems*, 3(1):23–28, 1989.
- [9] J.S.C. Chen and T.E. Stern. Throughput analysis, optimal buffer allocation, and traffic imbalance study of a generic nonblocking packet switch. *IEEE Journal on Selected Areas in Communications*, 9(3), 1991.
- [10] J.P. Gaver. Probability models for multiprogramming computer systems. In *Journal of the Association for Computing Machinery*, volume 14, 1967.
- [11] E.H. Goeldner. The network evolution towards b-isdn: Applications, network aspects, trials (e.g. berkorn). In *IEEE International Conference on Communications*, 1990.
- [12] W.J. Gordon and G.F. Newell. Closed queueing systems with exponential servers. In *Operations Research*, volume 15, 1967.

- [13]J. Hayes. *Computer Architecture and Organization*. McGraw-Hill, 1988.
- [14]I. Iliadis and W.E. Denzel. Performance of packet switches with input and output queueing. In *IEEE International Conference on Communications*, 1990.
- [15]J.R. Jackson. Jobshop-like queueing systems. In *Management Science*, volume 1, 1963.
- [16]M. Kawarasaki, H. Saito, and H. Yamada. An analysis of statistical multiplexing in an atm transport network. *IEEE Journal on Selected Areas in Communications*, 9(3), 1991.
- [17]L. Kleinrock. Isdn: The path to broadband networks. *Proceedings of the IEEE*, 79(2), 1991.
- [18]N. Komiya. Personal multimedia workstation. In *IEEE International Conference on Communications*, 1990.
- [19]S.S. Lavenberg. A perspective on queueing models of computer performance. *Performance Evaluation*, 10:53-76, 1989.
- [20]E.D. Lazowska, J. Zahorjan, G.S. Graham, and K.C. Sevvick. *Computer Systems Analysis Using Queueing Network Models*. Prentice-Hall, 1984.
- [21]P. Lewis and G.S. Shedler. A cyclic-queue model of system overhead in multiprogramming computer systems. In *Journal of the Association for Computing Machinery*, volume 18, 1971.
- [22]S.Q. Li and J.W. Mark. Traffic characterization for integrated services networks. *IEEE Transactions on Communications*, 38(8), 1990.
- [23]M.M.K. Liu. From fiber to the home to full broadband isdn. In *IEEE International Conference on Communications*, 1990.
- [24]B. Maglaris, R.R. Boorstyn, S. Panwar, T. Spiritos, P. O'Reilly, and C. Jack. Routing of voice and data in burst-switched networks. *IEEE Transactions on Communications*, 38(6), 1990.
- [25]M.K. Molloy. *Fundamentals of Performance Modelling*. Macmillan, 1988.
- [26]K. Nakamura, M. Takahashi, N. Inuzuka, N. Yanai, S. Marau, and K. Kaneda. Personal multimedia teleconferencing terminal. In *IEEE International Conference on Communications*, 1990.
- [27]A. Papoulis. *Probability, Random Variables, and Stochastic Processes*, chapter 12. McGraw-Hill, 1984.
- [28]J. Poll. The ibm-apple alliance. *New York Times*, 1991.

- [29]M. Reiser. A queueing network analysis of computer communication networks with window flow control. *IEEE Transactions on Communications*, 27, 1979.
- [30]M. Reiser and H. Kobayashi. Queueing networks with multiple closed chains: Theory and computational algorithms. *IBM Journal of Research and Development*, 19, 1975.
- [31]P. Schweitzer. Approximate analysis of multiclass closed networks of queues. In *International Conference on Stochastic Control and Optimization*, 1979.
- [32]S. Shenker and A. Weinrib. Greed is not enough: Adaptive load sharing in large heterogeneous systems. *IEEE Transactions on Communications*, 38(6), 1990.
- [33]D.S. Spears. Broadband isdn —[] service visions and technological realities. *International Journal of Digital and Analog Cabled Systems*, 1:3–18, 1988.
- [34]K. Sriram, R.S. Mckinney, and M.H. Sherif. Voice packetization and compression in broadband atm networks. *IEEE Journal on Selected Areas in Communications*, 9(3), 1991.
- [35]W. Stallings. *Computer Organization and Architecture*. Macmillan, 1987.
- [36]W. Stallings. *ISDN: An Introduction*. Macmillan, 1989.
- [37]W. Stallings. Queueing analysis. *Byte*, 1991.
- [38]S. Treves. Experience from early applications of broadband professional communications. In *IEEE International Conference on Communications*, 1990.
- [39]C. Weiss. Desk top video conferencing - an important feature of future visual communications. In *IEEE International Conference on Communications*, 1990.
- [40]W.W. Wu and A. Livne. Isdn: A snapshot. *Proceedings of the IEEE*, 79(2), 1991.

END

24|12|92|

FIN