

2000

Traffic Profiles and Performance Modelling of Heterogeneous Networks

Muller, Georg

<http://hdl.handle.net/10026.1/2364>

<http://dx.doi.org/10.24382/4709>

University of Plymouth

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

**Traffic Profiles and Performance Modelling
of Heterogeneous Networks**

by

Georg Müller
BSc (Hons)

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Electronic, Communication and Electrical Engineering
Faculty of Technology

In collaboration with
AT&T (UK) Ltd, England

May 2000

Traffic Profiles and Performance Modelling of Heterogeneous Networks

Georg Müller

Abstract

This thesis considers the analysis and study of short and long-term traffic patterns of heterogeneous networks. A large number of traffic profiles from different locations and network environments have been determined. The result of the analysis of these patterns has led to a new parameter, namely the 'application signature'. It was found that these signatures manifest themselves in various granularities over time, and are usually unique to an application, permanent virtual circuit (PVC), user or service. The differentiation of the application signatures into different categories creates a foundation for short and long-term management of networks. The thesis therefore looks from the micro and macro perspective on traffic management, covering both aspects.

The long-term traffic patterns have been used to develop a novel methodology for network planning and design. As the size and complexity of interconnected systems grow steadily, usually covering different time zones, geographical and political areas, a new methodology has been developed as part of this thesis. A part of the methodology is a new overbooking mechanism, which stands in contrast to existing overbooking methods created by companies like Bell Labs. The new overbooking provides companies with cheaper network design and higher average throughput. In addition, new requirements like risk factors have been incorporated into the methodology, which lay historically outside the design process. A large network service provider has implemented the overbooking mechanism into their network planning process, enabling practical evaluation.

The other aspect of the thesis looks at short-term traffic patterns, to analyse how congestion can be controlled. Reoccurring short-term traffic patterns, the application signatures, have been used for this research to develop the "packet train model" further. Through this research a new congestion control mechanism was created to investigate how the application signatures and the "extended packet train model" could be used. To validate the results, a software simulation has been written that executes the proprietary congestion mechanism and the new mechanism for comparison. Application signatures for the TCP/IP protocols have been applied in the simulation and the results are displayed and discussed in the thesis. The findings show the effects that frame relay congestion control mechanisms have on TCP/IP, where the re-sending of segments, buffer allocation, delay and throughput are compared. The results prove that application signatures can be used effectively to enhance existing congestion control mechanisms.

Index

Abstract	I
Index	II
List of Figures	VIII
List of Tables	X
Glossary of Terms	XI
Acknowledgement	XVI
Declaration	XVII

<u>Chapter 1: Introduction</u>	<u>1</u>
--------------------------------	----------

1.1 Introduction	2
1.2 Layout of the Thesis	4

<u>Chapter 2: Frame Relay</u>	<u>7</u>
-------------------------------	----------

2.1 Introduction to Frame Relay	8
2.2 Expanding Network Requirements	9
2.3 Basic Concepts	9
2.4 Service Provider Multi-Service Networks	10
2.5 Comparison of Asynchronous Transfer Mode with Frame Relay	11
2.6 Comparison of Frame Relay with Time Division Multiplexing	13
2.7 Comparison of Frame Relay with Cell Relay	14
2.8 Advantages of Frame Relay	15
2.9 Frame Formats	16
2.10 Routing	18
2.11 Permanent Virtual Circuit	19
2.12 Proprietary Technology	20
2.12.1 Minimum Information Rate	21
2.12.2 Peak Information Rate	21
2.12.3 Excess Information Rate	21
2.13 Private WANS	22
2.14 Carrier Services	23
2.15 Quality of Service	24
2.15.1 Throughput	25
2.15.2 Delay	26

2.15.3 Race for QoS	26
2.15.4 Network Services	29
2.16 Implementation	31
2.17 PVC Pricing	32
2.18 PVC without CIR	33
 <u>Chapter 3: Congestion Control Mechanisms</u>	 <u>34</u>
3.1 Introduction	35
3.2 Congestion	36
3.2.1 Flow Control	39
3.2.2 Different Mechanism Philosophies	41
3.2.3 Fairness	42
3.2.4 Beginnings Of Congestion Control	43
3.2.5 Open Loop Congestion Control	44
3.2.6 Closed Loop Congestion Control	46
3.3 Congestion Control In Frame Relay	48
3.3.1 Explicit Congestion Notification	50
3.3.2 Source ECN	50
3.3.3 Destination ECN	51
3.3.4 Discard Eligibility	52
3.4 Proprietary Congestion Control Mechanism: PM	53
3.4.1 Credit Manager	54
3.4.2 The Frame Relay Port Card	57
3.4.3 The Network Trunk Card	59
3.4.4 Round Trip Delay	61
3.4.5 CIR in PM	61
3.4.6 FECN and BECN	62
3.4.7 Cell Loss Priority	62
3.5 TCP/IP Congestion Control And Avoidance	63
3.6 Conclusion	64

Chapter 4: Network Analysis, Models and Tools	67
4.1 Introduction	68
4.2 Queuing Models	68
4.3 Queuing Systems	69
4.4 Poisson Distribution	70
4.5 Packet Trains	71
4.6 Simulation Modelling	72
4.7 Network Management and Monitoring	74
4.8 Monitoring Objectives	75
4.8.1 Different Types of Monitoring	76
4.8.2 Monitoring Agents	76
4.8.3 Simple Network Management Protocol	77
4.8.4 Protocol Analyser / Monitors Used for the Project	79
4.8.5 Data Collection	80
4.8.6 W&G Network Analyser	81
4.8.7 Sniffer on a PC	82
4.9 Estimating Network Traffic	83
4.10 The Network Traffic Estimation Process	84
4.10.1 Performance-Measurement Matrix	85
4.10.2 Defining of Measurement Tasks and Requirements	86
4.10.3 Analysis of Structure, Applications and Protocols	87
4.11 Protocol Modelling of Proprietary Mechanism	87
4.11.1 Representation of the Fixedpacket Overhead	88
4.11.2 Calculation of Fixedpacket Overhead	88
4.12 Analysing TCP/IP	90
4.13 Markov Model	93
4.14 Granularity	94
4.15 Flow Definition	95
Chapter 5: Analysis Results of Obtained Data	96
5.1 Network Profiling	97
5.1.1 Utilisation at University of Plymouth	97
5.1.2 Utilisation at AT&T	99
5.1.3 Packet Size Distribution at the University of Plymouth	101
5.1.4 Packet Size Distribution at AT&T	103

5.1.5	Conclusion	104
5.2	Daily Traffic of Frame Relay Network	105
5.2.1	Individual PVC Profiling	106
5.2.2	Distribution	108
5.2.3	Variability	109
5.2.4	Traffic Patterns	111
5.2.5	Locality	112
5.3	Application Signatures	114
5.3.1	Introduction	114
5.3.2	Traffic Patterns	116
5.3.3	Visual Study	117
5.3.4	Packet Size Distribution	119
5.3.5	Responder / Originator Ratio	121
5.3.6	Extension of Packet Train Model	123
5.3.7	Summary	126
5.4	Traffic Characteristics of WWW Sessions	126
5.4.1	Flow Inter-Arrivals	127
5.4.2	Measuring Conditions	129
5.4.3	Empirical Results	130
5.4.4	Conclusions	133
 <u>Chapter 6: Network Planning System</u>		 <u>135</u>
6.1	Introduction	136
6.2	Estimating Network Traffic	137
6.3	Perception	138
6.4	Rule of Thumb	139
6.5	The Network Traffic Estimation Process	139
6.6	Definition of Overbooking	140
6.7	The New Methodology	144
6.8	Development of the Basics	146
6.8.1	Asset Identification and Valuation	148
6.8.2	Threat Assessment	149
6.8.3	Uncertainty Factor	149
6.8.3.1	Life Cycle	150
6.8.3.2	The Trial Phase	150

6.8.3.3	The Introduction Phase	151
6.8.3.4	Growth Phase	152
6.8.3.5	Maturity Phase	152
6.8.3.6	Upgrade Phase	152
6.8.4	Strategy	153
6.8.4.1	Re-routing	153
6.8.4.2	Overbooking risks	155
6.8.4.3	Overbooking on a Trunk	155
6.8.4.4	Overbooking on a Port	157
6.8.4.5	Overbooking on Time of Day	162
6.8.4.6	EIR Guarantees for LAN Inter-Connection	163
6.9	Methodology	166
6.10	Conclusion	170
<u>Chapter 7: New Congestion Control Technique</u>		<u>172</u>
7.1	Introduction	173
7.2	Existing Models	175
7.3	New Approach	179
7.3.1	Packet Train Profiler	181
7.3.2	New Mechanism	182
7.4	Assumptions for the Simulation	187
7.5	Simulation of PVCs	189
7.5.1	Effects of Congestion Window on Throughput and Delay	189
7.6	Simulation for Congestion at the Trunk	196
7.6.1	Results of Trunk Overload	197
7.6.2	The Influence of the BD-B Queue Level Threshold on Throughput	200
7.7	Simulation for Congestion at the Port	202
7.7.1	Overload on the Transmit Port queue	202
7.7.2	No Overload at the Transmit Port queue	205
7.8	Conclusion	207

Chapter 8: Conclusion	212
References	219
Appendix A: Simulation Code	A-1
Appendix B: Simulation Screens	B-1
Appendix C: Simulation Results	C-1
C.1 Delay, Goodput and Loss Results	C-2
C.2 Utilisation levels by Night	C-7
C.3 Utilisation levels by Day	C-15
C.4 Application Ratios	C-22
C.4.1 Application 1	C-22
C.4.2 Application 2	C-23
C.4.3 Application 3	C-25
C.4.4 Application 4	C-26
C.4.5 Application 5	C-27
C.4.6 Application 6	C-28
C.4.7 Application 7	C-29
C.5 Session Duration for Different Departments	C-31
C.5.1 Session Duration - Secretarial	C-31
C.5.2 Session Duration – Maintenance Department	C-32
C.5.3 Session Duration – Researchers in Multimedia Lab	C-33
C.6 No Overload on Transmit Port Queue	C-35
C.6.1 No Overload PM	C-35
C.6.1 No Overload PTP	C-37
C.7 Overload on Transmit Port Queue	C-39
C.7.1 Overload PM	C-39
C.7.2 Overload PTP	C-41
C.8 BD-B levels during trunk overload	C-44
C.8.1 BD-B levels PTP	C-44
C.8.2 BD-B levels PM	C-47
C.9 Autocorrelation Function for Packet Trains in Applications	C-50

Traffic Profiles and Application Signatures

Strategies for Content Migration on the World Wide Web

Traffic Characteristics of WWW Sessions

A New Methodology on Overbooking in Frame Relay Networks

List of Figures

Figure 2.1:	Frame Relay Frame Format	16
Figure 2.2:	Fixedpacket Format	17
Figure 3.1:	Congestion and Effects on Throughput	37
Figure 3.2:	Open Loop Congestion Mechanisms	46
Figure 3.3:	Closed Loop Congestion Control Mechanisms	48
Figure 3.4:	Frame Relay Port Card	58
Figure 3.5:	Network Trunk Card	60
Figure 4.1:	Poisson Distribution of Arrival Rates	71
Figure 4.2:	The Type 8137 Assigned to the IPX Protocol	83
Figure 4.3:	Parameters for a Two-State MMPP Source	93
Figure 5.1:	Utilisation at University of Plymouth by Night	98
Figure 5.2:	Utilisation at University of Plymouth by Night during Holidays	98
Figure 5.3:	Utilisation at University of Plymouth by Day	99
Figure 5.4:	Utilisation at University of Plymouth by Day during Holidays	99
Figure 5.5:	Utilisation at AT&T by Day	100
Figure 5.6:	Utilisation at AT&T by Night	100
Figure 5.7:	Packet Size Distribution at the University of Plymouth - IP	102
Figure 5.8:	Packet Size Distribution at University of Plymouth - IPX	103
Figure 5.9:	Packet Size Distribution at AT&T	104
Figure 5.10:	Overall Frame Relay Traffic	105
Figure 5.11:	Utilisation for 5 Days at Peak Hours	107
Figure 5.12:	Utilisation for 5 Days at Peak Hours	107
Figure 5.13:	Utilisation at Peak Hour 256 kbit/s CIR	109

Figure 5.14: Change in Dynamic Utilisation over May 1997	110
Figure 5.15: Illustration of a Step Change for a Single PVC	111
Figure 5.16: IPX to TCP/IP Ratio used purely on a LAN	113
Figure 5.17: LAN/WAN Ratio	113
Figure 5.18: Ratio of Computers involved in Traffic at the University of Plymouth	114
Figure 5.19: Measurement at 5 Per Cent Utilisation	118
Figure 5.20: Measurement at 20 Per Cent Utilisation	119
Figure 5.21: Measurement at 30 Per Cent Utilisation	119
Figure 5.22: Packet Size Distribution at 5 Per Cent Utilisation	120
Figure 5.23: Packet Size Distribution at 20 Per Cent Utilisation	120
Figure 5.24: Packet Size Distribution at 30 Per Cent Utilisation	120
Figure 5.25: Packet train one	124
Figure 5.26: Packet train two	124
Figure 5.27: Autocorrelation for Packet Sizes inside a Packet Train	125
Figure 5.28: Session Duration for the Administration LAN Segment during Day 1	131
Figure 5.29: Session Duration for the Administration LAN Segment during Day 2	131
Figure 5.30: Inter-Arrival Time of Users	132
Figure 5.31: Logarithm of Inter-Arrival Times	133
Figure 6.1: Usual Overbooking Table	144
Figure 6.2: Design Methodology	146
Figure 6.3: Three Strategies for Setting MIR Relative to AR	158
Figure 6.4: Routing Problem	161
Figure 6.5: Minimum Average Expected EIR	164
Figure 6.6: An Indication for the Improvement of EIR Availability	166
Figure 6.7: Methodology	169
Figure 7.1: Frame Relay Network	176
Figure 7.2: PM and TCP/IP Control Loop	179
Figure 7.3: Information Flow from Frame Traces to PTP	182
Figure 7.4: Two Flow Controls connected With Packet Train Profiler	183
Figure 7.5: Algorithm State Diagram	183

Figure 7.6: Layout for Simulation Of PVC	190
Figure 7.7: Throughput on Increased Window Size	191
Figure 7.8: Mean Round Trip on Increased Window Size	192
Figure 7.9: Layout for the Simulation	196
Figure 7.10: BD-B Queue Level Distribution During Simulation with PTP	197
Figure 7.11: BD-B Queue Level Distribution During Simulation with PM	198
Figure 7.12: Trunk Throughput Versus BD-B Queue Threshold – PTP	201
Figure 7.13: Trunk Throughput Versus BD-B Queue Threshold – PM	201
Figure 7.14: Layout for the Transmit Port Queue Simulation	203
Figure 7.15: Overload on Transmit Port Queue in PM	204
Figure 7.16: Overload on Transmit Port Queue in PTP	204
Figure 7.17: No Overload on Transmit Port Queue in PTP	206
Figure 7.18: No Overload on Transmit Port Queue in PM	207

List of Tables

Table 4.1: Performance Measurement Entities	74
Table 5.1: Responder to Originator Ratio	123
Table 5.2: Autocorrelation for Packet Sizes Inside Packet Train	124
Table 7.1: Performance Metrics for Frame Relay, PM and PTP	194

Glossary of Terms

ABR	Available Bit Rate
ACF	Auto Correlation Function
ANSI	American National Standards Institute
AR	Access Rate
ARPANET	Internet
ATM	Asynchronous Transfer Mode
AT&T	American Telegraph and Telefon Company
BBN	Packet switches from Bolt, Bereneck and Newman
BD-B	Queue in the switch
BECN	Backward Explicit Congestion Notification
Be	Excess burst Rate
Bc	Comitted Burst Rate
Bit/s	Bits Per Second
CCITT	See ITU-T
CDV	Cell Delay Variation
CIR	Committed Information Rate
CLP	Cell Loss Priority
CLR	Cell Loss Rate
Cmax	Max amount of saved Credits that can be accumulated during silent periods and can be used in busy period
CMR	Credit Manager Rate
CRC	Cyclic Redundancy Check
CSMA\CD	Carrier Sense Multiple Access with Collision Detect

DCE	Data Communication Equipment
DDS	Digital Data Service, or DATAPHONE Data Service. DDS is a fixed-bandwidth point-to-point digital service.
DE bit	Discard Eligibility bit
DNS	Domain Name Service
DOS	Disc Operations System
DRAM	Dynamic Random Access Memory
DTE	Data Terminal Equipment
ECN	Explicit Congestion Notification
EIR	Excess Information Rate
EISA	Extended Industry-Standard Architecture
EMI	Electromagnetic Interference
EPROM	Erasable Programmable Read Only Memory
FCS	Frame Check Sequence
FDDI	Fiber Distributed Data Interface
FECN	Forward Explicit Congestion Notification
FIFO	First In First Out
FRAD	Frame Relay Access Device
FRP	Frame Relay Port
FTP	File Transfer Protocol
Goodput	Rate at which the data are successfully transmitted up to the TCP layer. Retransmitted data are counted only once
GUI	Graphical User Interface
HTML	Hypertext Mark-up Language

IP	Internet Protocol. All the networks that make up the Internet speak IP as a common language.
IPX	Inter Packet Exchange protocol (also Ping Protocol)
IPX switch	Inter Packet Exchange equipment
ISA	Industry-Standard Architecture
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider. A company or organisation selling Internet access. ISPs may offer many levels of access from dial-up host-based services to direct network connections (WAN).
ISU	Integrated Service Unit. See CSU/DSU.
ISO	International Organisation for Standardisation
ITU	International telecommunication Union
ITU-T	Telecommunication Standardisation Sector of the ITU
kB	Kilobyte
kbit/s	Kilobits per second. A thousand bits per second.
LAN	Local Area Network
LAPD	Link Access Procedure
LIFO	Last In First Out
MAC	Media Access Control
MAN	Metropolitan Area Network
MB	Megabyte
Mbit/s	Megabits per second. One million bits per second.
MDR	Minimum Data Rate
MIB	Management Information Base
MIR	Minimum Information Rate

MMPP	Markov Modulated Poisson Process
nntp	Net News Transport Protocol
NTC	Network Trunk Card
NTG	Normalised Throughput Gradient
OBF	Overbooking Factors – Oversubscription Factor
OSI	Open Systems Interconnection
PBX	Private Branch Exchange
PC	Personal Computer
PDU	Protocol Data Units
PIR	Peak Information Rate – The max rate a PVC can send information
PM	Proprietary Mechanism
PP	Ping Protocol
PPP	Point to Point Protocol. A replacement for SL/IP designed to provide IP services via modem.
PSTN	Public Switched Telephone Network
PTP	Packet Train Profiler
PVC	Permanent Virtual Circuit
QIR	Quiescent Information Rate. Foresight starts transmitting at this rate, when the line was silent for some time (time period can be set by the network provider)
QoS	Quality of Service
RAM	Random Access Memory
RTD	Round Trip Delay
rlogin	Remote Login
RMON	Remote Monitoring Agent

ROM	Read Only Memory
RSVP	Resource Reservation Protocol
SL/IP	Serial Line/Internet Protocol. A protocol designed to provide IP over serial connections. Commonly used for dial-up, although the new PPP is better equipped for this application.
SMTP	Simple Mail Transfer Protocol
SNA	System Network Architecture
SNMP	Simple Network Management Protocol
TCP	Transfer Control Protocol
telnet	Remote Login Terminal emulation
TEP	Traffic Estimation Process
TDM	Time Division Multiplexing
UNI	User Network Interface
VC	Virtual Circuit
VLAN	Virtual LAN
VP	Virtual Path
VPN	Virtual Private Network
WAN	Wide Area Network. A network that spans a large geographic area.
WWW	World Wide Web. Multimedia hypertext-based system that uses Hyper Text Mark-up Language (HTML)to provide access to services and information.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to the following people:

Peter Sanders, my Director of Studies, for his encouragement and support and for his confidence in my abilities

Benn Lines and Paul Reynolds, my Supervisors, for invaluable research input and help with this thesis;

John Allen, Network Analysis Manager of AT&T (UK) Ltd in Redditch (presently Director of Research at Netscient Ltd), who was my industrial supervisor and provided valuable help and input to the research programme;

The Network Planning Group of AT&T (UK) Ltd in Redditch, for their continuous support and constructive criticism of my research;

Stephen Furnell from the Network Research Group for his total commitment and support; and

Last but not least thanks to my parents, Alex and Mette for their support.

DECLARATION

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other university award.

The study was financed with funding from Plymouth University and was carried out in collaboration with AT&T (UK) Ltd.

Relevant conferences and scientific seminars were regularly attended and the work presented. A large part of the practical research was undertaken at AT&T (UK) Ltd's Planning Centre in Redditch. In addition several papers were prepared for publication, details of which are listed in Appendices.

Signed:

Date:


30.05.2000

Chapter 1: Introduction

1.1 Introduction

The explosive growth of computer networks and the massive interconnection of heterogeneous network infrastructures in recent years have created the need for analysis and modelling tools. There is a large number of existing products that can be used to obtain network utilisation. The use of these products mostly evaluates network problems and allows management operations and failure measurement. The estimation of future network traffic is as important as the influence and impact of different protocols and their applications on network performance.

Selecting a LAN or WAN architecture and system that will prove the optimum service to the users requires analysis and knowledge of the environment and the users' needs. A network is a productivity-enhancing tool. Eliminating redundancies and sharing resources is a major task of a network, but if not used properly, it can actually decrease productivity. There are many factors which influence a network planning and therefore must be included in the planning process and predicting of networks and their growth.

This thesis shows that it is possible to identify traffic patterns depending on various protocols in a Local Area Network (LAN) environment. The increasing demands of interconnecting LANs through Wide Area Network (WAN) technology create optimisation problems and the need for innovation in solving these problems.

The analytic approach enables a broad view of network loading and is very useful for network planning and design. However, understanding the dynamic load of a network and the reasons for packet losses and capacity overflow in various circumstances is crucial.

Reasons for this can be high utilisation levels on trunks created by bursty traffic on Permanent Virtual Circuits (PVC). Bursty traffic, however, is the “nature” of Frame Relay and has the advantage of statistical overloading of a trunk and sharing of trunk and port resources. The problem is, therefore, the planning of capacity with bursty traffic. As more network applications are used for client/server processes, the data traffic in this environment is typically more bursty than in a traditional hierarchical network.

The three main goals of this dissertation have been to use traffic profiles for characterisation, to evaluate these long term profiles and develop a new methodology for network planning, and to develop a congestion control mechanism which is using short term profiles for better congestion management. The research in this thesis therefore focuses mainly on these three areas.

Firstly, the collection and analysis of traffic profiles for short-term and long-term periods that have both led to new classifications of traffic. Long-term profiles have been analysed in granularities of months, weeks, days and hours to have an impression of data loads and growths on Frame Relay systems. The virtual connections show the same repeatable throughput patterns, which have led to the new classification: *PVC life cycle*. Short-term profiles are analysed in granularity of seconds. These profiles have shown unique and repeatable pattern, and have led to a new model: *application signatures*.

The second aim of this thesis is to develop a methodology, which allows a rule of thumb assessment and design of a network with future and existing network traffic based on the analysis of traffic profiles. It is often desired to have an overbooking of the Frame Relay circuits with a minimum risk of packet losses. To do this, the PVC life cycle and the

Traffic Estimation Process (TEP) are used. The result is the development of a *generic methodology*, which allows the commercial overbooking of Frame Relay networks.

Thirdly, the application signatures have been used to develop a system, the *Packet Train Profiler (PTP)*, which can be integrated into existing congestion control mechanisms to influence the performance of existing permanent virtual circuits, ports and the whole system. To analyse the effects of the PTP, the congestion control mechanisms in AT&T's Frame Relay networks has been simulated with and without PTP. The results are discussed and show enhancement of the network efficiency and throughput.

In order to achieve these three goals, the main characteristics of Frame Relay itself are explained and a vendor-specific implementation has been used. Also, different congestion control mechanisms are investigated to get an overview of the subject.

1.2 Layout of the Thesis

Chapter 2 serves as an introduction to Frame Relay and the field of network technology. Frame Relay is standardised but many implementations exist, which interpret the standards differently and solve the requirements in a proprietary form. One of the proprietary frame relay technologies was chosen and introduced in this chapter.

Chapter 3 investigates the different congestion control mechanisms. The problem of congestion is old and a variety of mechanisms have been developed solving some problems but also creating new ones. The chapter further describes a proprietary mechanism, which is used by the network equipment provider of AT&T. The system is simulated and analysed in chapter 7. The proprietary mechanism is furthermore used as an example in the later simulation and developed into a new mechanism.

Chapter 4 describes the modelling and analysis of networks and also of the Transmission Control Protocol and the Internet Protocol, known as well as the Internet protocol suite TCP/IP. The techniques described here are used in the follow-up chapters. The investigation, selection and development of network models for LAN, WAN and gateway systems and tools, which have been used to obtain the results, are undertaken in this chapter. Also properties of the existing Traffic Estimation Process (TEP) are shown and discussed.

Chapter 5 describes the development and production of a set of traffic profiles for LANs and WANs and also different protocols. The traffic profiles in this chapter were collected over a period of several months, and in different locations and networks. Traffic profiles are in constant change, and analysis of selected customers' traffic to establish profiles of LANs and WANs is necessary. To establish a view of short-term predictability, measurements for packet size distributions, packet inter-arrival times and probability of packet arrivals have been analysed. The results are displayed and resulted in a new model: the *application signature*. The application signature can be used for short-term prediction of traffic and the use is demonstrated and simulated with a congestion control mechanism in chapter 7. Also a look has been taken on long-term traffic profiles. Traffic patterns over the hours of the day, days of the week etc. have been analysed. One of the key questions is how predictable the network build-up is and what models to use for a future forecast. Investigations are workload characteristics from the network perspective, including the distribution on active protocols. The traffic generated by the customers is dictated by different parameters like user habits, client/server applications and protocols. Analysis of network traffic by time of day, week and month has resulted in the classification of *traffic*

life cycles which has led to the development of the *new overbooking methodology* in chapter 6.

Chapter 6 uses some of the findings from chapter 5 and the existing properties of the TEP to develop a new methodology for overbooking and the network planning process. The environmental factors, user behaviour and other factors influence a traffic profile, and a constant update of these profiles is necessary. The typical overbooking factor of the Frame Relay traffic is set at a constant 30 per cent by many network providers. This has no sound explanation just the “gut feeling” of experienced network planners and initial research in laboratories have set these rules. The figures from Chapter 5 have shown that overbooking levels can vary from 0 per cent up to 500 per cent. Applying the methodology to overbooking values of PVC of Frame Relay services will take this into account and solve many problems by “balancing” the network. The chapter shows how the methodology can be used and gives recommendations of overbooking.

Chapter 7 describes the development of an access network optimisation tool, based on queue delay constraints and input traffic conditions. The simulation used the previous findings of the packet-train model and new findings, which are described in Chapter 5. The implementation of a simulation tool using the traffic and analysis techniques into an experimentation of the Frame Relay network is considered appropriate, as it uses the network’s proprietary mechanism to show the problems existing in today’s mechanisms. To simulate the Frame Relay Network, a model is developed representing the physical object. The execution provides results and information on the performance of this model.

The concluding remarks and ideas about future research are shown in Chapter 8.

Chapter 2: Frame Relay

2.1 Introduction to Frame Relay

Frame Relay is growing in activity and interest at a unique rate. There are already many network providers and users in place and the market keeps expanding. Compared to the widely-available infrastructure of X.25 and leased lines it is still a small percentage. Most observers, however, agree that this will change rapidly over the following few years.

Today a Frame Relay service from public carriers is available world-wide. Its reliability and effectiveness have been demonstrated since the first networks were established in 1992. Frame Relay evolved from X.25 packet switching and uses variable-length frames to transport the user traffic across the interface.

Frame Relay, compared with the older networks such as X.25, eliminates much of the protocol processing carried out by the network and thereby reduces the portion of transmission delays due to protocol processing. X.25 is a protocol that was designed to support a traditional data-networking environment. The protocol contains functionality and features that have now been made redundant by intelligent end stations, higher layer protocols and higher quality digital networks. The simplification of the protocol focuses on the elimination of error recovery functions inside the network, which take most of the time and resources. The logical step of this was the reduction of buffer requirements in the switches. Therefore, Frame Relay works on the Open System Interconnection (OSI) level 2 and does not guarantee error free end-to-end transfer. Instead, the endpoint devices (e.g. Personal Computers (PCs) or workstations) are responsible for delivery of the data, not the network itself. This protocol processing, which is still necessary to guarantee the accurate delivery of the data, is left to the higher layers inherent in the transported data.

2.2 Expanding Network Requirements

Due to a number of events occurring in both enterprise and service provider networks, including on-going advances in computing power, more desktop interaction, the internet, more transactions, more visual content, an explosion of new applications, etc., greater demands have been placed on both the private and public networks.

To maximise capacity utilisation and flexibility, networks are moving from dedicated circuits with fixed capacity between devices to virtual networks. A virtual network comprises logical connections (virtual circuits) which dynamically share physical capacity on an as needed basis with other logical connections (virtual circuits) or networks.

2.3 Basic Concepts

When discussing Frame Relay one normally deals with two concepts. The first is Permanent Virtual Circuits (PVCs). A circuit is permanent due to a route having been set up prior to the transfer of the data and virtual due to the fact that many PVCs from the same source can share one single electrical interface. Frame Relay is basically a multiplexed interface to a packet-switched network. The route is a logical connection and the switches do not have to make a routing decision, as the established route is always the same. It is permanent as the connection is not only established for one call, but transfers the user traffic within the frame without regard to its contents, thereby providing service which is effectively as transparent as a leased line.

The second concept is the Committed Information Rate (CIR). This is the contracted transmission rate the user has negotiated with the telephone company. The CIR usually promises the user the ability to transmit at a specified speed all the time. However, it is

sometimes possible to transmit at much higher speeds, by virtue of the nature of Frame Relay, but the user will always attain the negotiated CIR. This can be achieved through the use of Committed Burst (Bc) and Excess Burst (Be) rate. Bc is the committed amount of data that the network is willing to accept in an increment amount of time Tc. The Be is the excess amount of data that the network is willing to accept in the same increment amount of time Tc. Frames which are sent at Be are marked and can be discarded by the network.

In explanation, Frame Relay sends data out in frames. While the message is being sent, the packets are sent at every opportunity when network capacity is available. Frame Relay provides a user with multiple independent data links to one or more destinations. Traffic on these data links is statistically multiplexed to provide efficient use of access lines and network resources. And here is the advantage for users and network providers. Users will always be able to occupy the capacity at the negotiated CIR but sometimes the data will “burst” above the CIR to much higher transmission rates. For example, a user can buy in at 28-kbit/s but due to the “bursty” nature of data transmission he will often send at 56-kbit/s or even higher with no additional expense in telecommunication costs or degradation in data or response time. It is also more cost-effective for the network provider as well as the user.

2.4 Service Provider Multi-Service Networks

The demand to provide Local Area Network (LAN) interconnections has driven most of the public service providers to consider ways to quickly react to this opportunity. Frame Relay has proved to be a reliable, cost-effective, standards-based service for transmitting LAN data, which tends to be very bursty in nature. Typically, LANs access the network at periodic intervals, and when they do, they often require large amounts of capacity for short

periods of time. It is not cost effective to provide sufficient capacity for every LAN connection on a full-time basis.

As Frame Relay traffic increases and customers demand more capacity for advanced applications, there are significant service advantages when Frame Relay is implemented on them. Since cell network platforms only allocate capacity when there is demand, the unused capacity from idle Frame Relay connections can be used by active connections. This allows the active connections to "burst" or to send large amounts of data for a short interval above their committed information rate. Then, as the connection goes idle, the capacity can be utilised for yet another connection.

A further advantage of frame relay networks is the flexibility of offering a PVC to interconnect all LAN sites in a mesh topology as opposed to using physical circuits that require a large investment in interface hardware and data circuits. Frame Relay networks offer features to minimise delay, maximise throughput, and avoid congestion.

2.5 Comparison of Asynchronous Transfer Mode with Frame Relay

Frame Relay has had spectacular success in the recent past for transporting packet data over Wide Area Networks (WANs). Carriers have priced frame-relay services at a substantial discount over equivalent private-line services. Similarly, private networks have been upgraded with frame-relay modules to provide an equivalent transport mechanism. A simple software upgrade has been the only requirement to convert the embedded base of routers from circuit connectivity to Frame Relay. Additionally, Frame-Relay Access Devices (FRADs) allow legacy data, such as System Network Architecture (SNA) traffic, to also take advantage of Frame Relay.

After some early mishaps, carriers chose to use price as the differentiator for Frame Relay services. A similar strategy seems inevitable with Asynchronous Transfer Mode (ATM). Those carriers losing the Frame Relay game will try to use ATM as a way of regaining market share. In the long run, it seems that this will result in approximately equal pricing of Frame Relay and ATM services. Users will then gravitate to one or other, based on other differentiators. Frame Relay is a data-only technology, and it is unlikely that the use of Frame Relay for voice, video, and multi-media communications will ever be more than experimental. Thus, these applications will migrate to ATM or stay on private lines. Frame Relays' primary advantages are that ATM is inefficient below broadband rates, and interfacing to frame-relay services is simple and inexpensive. Thus, Frame Relay may find a niche for data-only applications at narrowband rates, although even here there will be competition, especially from ISDN.

Users will, however, want interoperability between ATM and Frame Relay services. Consequently, the ATM Forum and the Frame Relay Forum are obliging with appropriate standards. Since Frame Relay can be carried "on top of" ATM, this leads to the likely scenario that wide-area back-bones will migrate to ATM in a few years, and Frame Relay will be used as an access technology for data-only applications at 2Mbit/s and below. Most equipment vendors are, therefore, assuming that future ATM-based networks and carrier-edge switch architecture will require both ATM and frame-relay connectivity.

2.6 Comparison of Frame Relay with Time Division Multiplexing

In Time Division Multiplexing (TDM) fixed slots of capacity are allocated. Once this capacity is allocated, it cannot be used by any other connected stations. In comparison to this is a statistical multiplexing environment, where capacity is dynamically shared among multiple connections. When a station has a frame of data to send and the circuit is idle, it gets all the capacity for as long as is necessary to send the frame. If the circuit is not idle, the frame is queued for transmission.

Because Frame Relay is a protocol based on statistical multiplexing, it allows the expensive access circuit into the AT&T network to be better utilised for bursty communications for multiple locations. Frame Relay allows the allocation of the entire access link to a single PVC for the duration of a burst. In contrast, TDM solutions limit the available capacity to a fraction of the overall access circuit. With the Frame Relay network, the access circuit is not subdivided into smaller capacity increments among connected end-points as is done in TDM, therefore, the emission time of a data burst is smaller, improving end-to-end performance of the application.

As statistical multiplexing is the basis of the design for LAN protocols such as Ethernet, which optimises the allocation of capacity on the LAN media among end stations, Frame Relay makes the WAN more LAN-like. It allows networks to be optimised for support of such traffic and provides better utilisation of capacity on both the network access and backbone trunking system.

Using a Frame Relay network access, the number of wide area ports necessary on a bridge, router or other end device is reduced. With Frame Relay, all connections are

supported on one hardware interface. This reduces expenditure on hardware. However, since the Frame Relay links between switches are shared, they may experience congestion.

2.7 Comparison of Frame Relay with Cell Relay

When comparing Cell Relay with Frame Relay it is useful to look at the attributes that can directly affect a user connected to a network using one of these technologies.

This comparison is meaningless however if the type of application to be used over the network is not accounted for.

Applications that generate

- short transactions;
- packetised voice;
- digital video; and/or
- large images

can benefit from Cell Relay's high speed and low latency.

Applications that exchange objects such as:

- files;
- large volumes of email;
- small images;

and other LAN to LAN type traffic can benefit from Frame Relay's efficiency and improved speed over previous packet technologies.

The two technologies can also complement each other and provide benefits to users and network providers when Frame Relay is used as an access protocol to a Cell Relay backbone network.

2.8 Advantages of Frame Relay

Frame Relay is popular for a variety of reasons, which are discussed in the following sections. The major advantages are

- good performance for LAN applications;
- less process intensive than X.25, resulting in higher network throughput with lower delay;
- provides a good interface for cell based technologies, e.g. ATM; and
- Frame Relay technology is suited to Internet access because of its transmission rate and its “bursting” nature.

Savings when utilising Frame Relay are substantial as providers and users are buying less equipment and paying less maintenance on equipment. Users sharing data line expenses with the others in the frame cloud without seeing a depreciation in response time or service. Also, Frame Relay is not “distance sensitive” so users are not paying a per mile leased line charge.

2.9 Frame Formats

In Figure 2.1 the Frame Relay format, which is defined by the ITU-T can be seen. Protocols and equipment can see this format, when connecting to the switches. Inside the proprietary Frame Relay network, the switch is like a black box, and equipment manufacturers create their own formats.

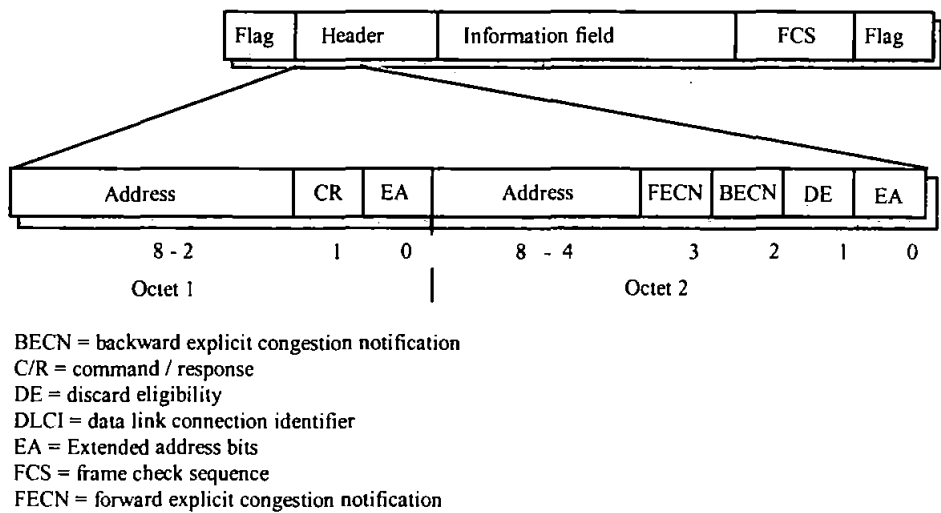
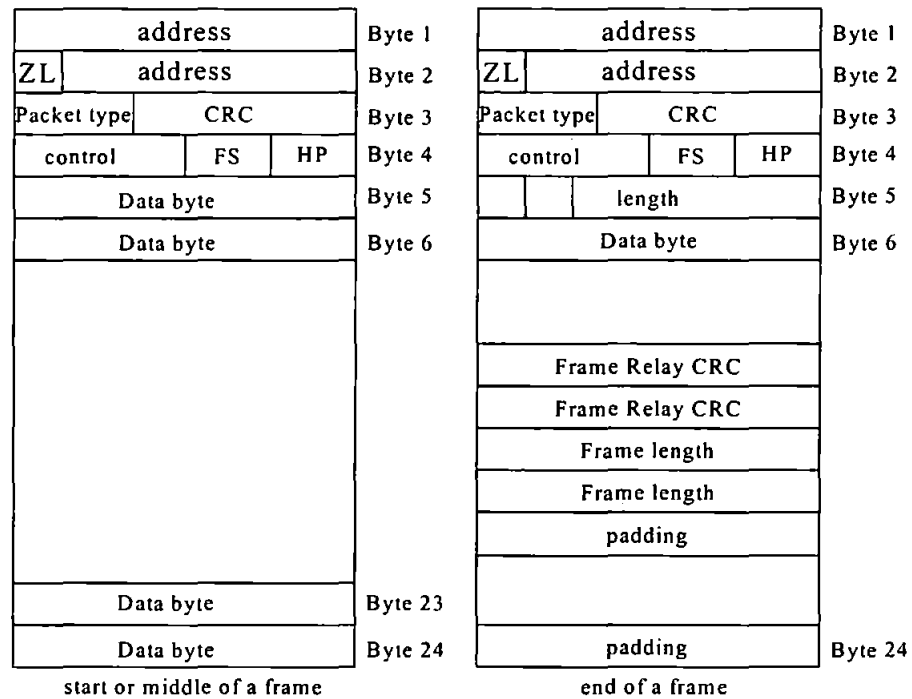


Figure 2.1: Frame Relay Frame Format

The nodes of the AT&T network use proprietary equipment which has its own format. It is the Fixedpacket format and can be seen in Figure 2.2. There are two different types of Fixedpackets, the start and the middle of a frame as well as the end of a frame. The end of a frame is represented by a Fixedpacket, which differs from the start/middle Fixedpacket because it fills the empty space with padding bytes. This is because not every Frame Relay frame will fit exactly in a whole number of fixed length Fixedpackets. The Fixedpacket type field is used to indicate whether the Fixedpacket is a Frame Relay or idle Fixedpacket. The idle Fixedpackets are used to send essential network information when no Frame Relay Fixedpackets are generated by the end users to carry this information.



Control bits 1,2 = indication of start/middle or end frame
Control bits 3 = FECN
Control bit 4 = CLP

Figure 2.2: Fixedpacket Format

The cyclic Redundancy Check (CRC) code is only computed for the Fixedpacket header, so the header information is checked for errors but the payload is not. The four control bits are used for two purposes. The first two bits identify the frame as start/middle/end of frame only. The last two bits carry the Forward Explicit Congestion Notification (FECN) bit and the Cell Loss Priority (CLP) bit. These two bits are used for congestion control in the AT&T network. The next two bits carry the up, down, fast down and non-messages used for flow controlling PVCs with PM.

The length field in the end of frame Fixedpackets indicates the amount of the data bytes in the Fixedpacket. The rest of the frame is filled with padding bytes. The frame length field (2 bytes) carries the information about total length of the frame. When reassembling the

frame this information is used to check if Fixedpackets have been lost and thereby to check if this is a valid frame or not.

2.10 Routing

The nodes allow two possible modes of routing:

- preferred routing
- non-preferred routing

Preferred routing allows the network provider to manually set up a route for a PVC. All frames on this PVC will then follow this route. Only in case of trunk failure is an alternative route taken. It will return to the preferred path when the outage is cleared.

Non-preferred routing is an algorithm which sets a path through the Frame Relay network. This path will exist till a trunk failure occurs. The algorithm is very simple and works by calculating the minimum number of hops from a start node to a destination node and then chooses the path with the largest unreserved capacity. On failure of a trunk the algorithm is re-run to choose a route from the trunks available, but the problem is that the path is not altered when the failure is cleared. This is a big limitation of the algorithm as large delay paths may arise and continue to persist until manually cleared by operator intervention.

AT&T in Europe decided to use preferred routing in its network to have direct control over the routing and thus avoid using nodes in America as intermediate hops for “European” traffic. Another reason is that price differences for trunks, which vary a lot in

Europe, are not taken into account by *non-preferred routing*. By the use of preferred routing the predictability of the operation of the network was possible.

2.11 Permanent Virtual Circuit

In the AT&T backbone for LAN interconnection services no call set-up negotiations are used because the connections are fixed PVCs. The transmission parameters of a PVC are negotiated when a customer orders his connections and are laid down in a service-level agreement specifying quality of service measures like throughput, delay and availability. In this way there is no feedback from the congestion situation in the backbone to the quality of service parameters. Therefore, a close monitoring of the traffic profiles is necessary and requires permanent modification in routing, overbooking etc.

This gives two problems. First, using the ITU-T ISDN Frame Relay definitions, LAN interconnections can cause serious problems, as fairness among end users is not guaranteed. The congestion management system in the Frame Relay is not able to flow-control the end users who are responsible for the congestion in the backbone. Currently the most used end-user protocols are not able to process congestion information from the backbone and translate it to flow control information to the end user causing congestion. Moreover, even if the end user could process congestion information, the end-user is not obliged to react to the congestion information to decrease its traffic flow. This can create unfairness in the network by ignoring the congestion information, thus, stealing capacity from other users, who are obeying the congestion information.

Second, the currently used end protocols like TCP/IP can make congestion worse when the network tries to recover from congestion by discarding data. In this situation it is likely

that the end-user protocols come out and start sending frames into the network to get acknowledgement and thus add to congestion, especially in the case of TCP/IP and large packets. The existing congestion control mechanisms are directly related to the performance of existing permanent virtual circuits, ports and the whole system. To understand the utilisation patterns of LAN interconnections some measurements were done in the operational backbone. The connections and trunk show the same repeatable throughput patterns.

A way around these problems is to have a congestion management system in the Frame Relay network that does not rely on end user response to congestion information and therefore enable fairness among users.

2.12 Proprietary Technology

As mentioned before, the proprietary switch studied in this research has a Frame Relay interface to the outside complying with the ITU-T standards, but it uses its own proprietary Fixedpacket format on the inside. Fixedpackets are fixed-length cells while Frame Relay uses variable length frames. Initially it was thought that by using fixed-length cells, instead of variable-length frames the delay of traffic in the backbone would be less variable and the transmission of voice with fixed short cells would be possible over this network. However, over time the technology evolved and voice is using different buffers in the switches and is therefore independent of the data transmission in the network. The thesis does not investigate any of these aspects. The Fixedpacket format for data still remains.

2.12.1 Minimum Information Rate

The Minimum Information Rate (MIR) is a parameter that is set by the network provider to ensure a minimum availability of capacity to the connection. Please note that this is not the CIR, but a parameter to ensure a specified throughput at all times. If, the network provider decides that the trunks should be oversubscribed, the sum of all MIR can exceed trunk capacity. This allows the switch to assign more connections to the packet trunk. If packets are sent into the network at a higher rate than MIR, an internal cell loss (CLP) priority bit will be set.

There is an inverse relationship between overall connection delay and network congestion. An increase in MIR results in a decrease in end user delay but increases the probability of packet trunk congestion.

2.12.2 Peak Information Rate

The Peak Information Rate (PIR) parameter that is set by the network provider and is used to set an upper limit to the transmitted data rate when PM is being used. When there is unused packet trunk capacity, the transmitted rate is allowed to climb to the PIR. The PIR cannot be higher than the Access Rate of the port.

2.12.3 Excess Information Rate

The Excess Information Rate (EIR) is the difference between the PIR and the MIR. The data sent with Excess Information Rate, implying that the information rate is above MIR, is marked discard eligible. Please note that all capacity is available for sharing. However, if all PVCs use their available MIR at the same time, data sent at EIR is discarded faster.

2.13 Private WANs

The term private WAN is used to define the Wide Area Network infrastructure of a single organisation - the infrastructure that enables applications critical to that organisation's business. The enterprise network may be owned and operated by the organisation, outsourced to a systems integrator, or even provided as a service by a carrier. While enterprise networks have existed for many years, they became critical to the operations of major US organisations in the early 1980s with the arrival of what were commonly known as T1 multiplexers. As WANs became global in the late 1980s, the products became known as capacity managers. These devices allocate relatively expensive wide-area capacity among various applications. Capacity managers transmit voice, data, and video traffic over the WAN and provide extensive recovery mechanisms in the event of a transmission-facility or equipment failure. They, in turn, are fed by a variety of devices, including Private Branch Exchanges (PBXs), front-end processors, terminal controllers and routers.

Originally, capacity managers operated at the physical layer only and were usually implemented using Time-Division Multiplexing (TDM) technology for maximum interoperability with the service providers' networks. However, as network applications migrated from mainframe-based to LAN-based processors, capacity managers added higher-layer functions such as Frame Relay and multi-protocol routing. These new multi-service capacity managers are presently the mainstay of most large enterprise networks world-wide. With the migration of data traffic towards the LAN in the late 1980s and early 1990s, an alternative architecture for the WAN emerged. Today, many enterprises also operate router-based WANs that interconnect multiple LANs throughout an organisation.

Capacity-manager and router networks are often integrated in the same enterprise, with the router traffic using dedicated or frame-relay capacity allocated by the capacity manager from the available pool.

2.14 Carrier Services

Carriers now offer a wide range of services to enterprise customers. The carrier services of interest for this discussion are those provided by carriers on a shared-network basis. Here, equipment is installed by the carrier to offer services to multiple organisations. In many cases, these services may appear to be dedicated to that organisation and are typically called Virtual Private Networks (VPNs). In other cases, they may provide switched connectivity to the entire community of connected users, as is the case with the Public Switched Telephone Network (PSTN).

Many US enterprises have turned much of their domestic voice network over to carrier VPN services and are looking (somewhat unsuccessfully) for equivalent services on a worldwide basis. As a result, US networks are now frequently segregated into voice-only and data-only networks, with the voice network based primarily on carrier services and the data network based primarily on enterprise equipment. Carriers worldwide have taken many initiatives to provide data-oriented virtual network services, such as packet-switched networks based on X.25. However, in the US, X.25 networks were not widely adopted, primarily because of the ready availability of relatively low-cost private-line services. More recently, frame-relay networks have filled the gap and are being used by many US corporations to provide data connectivity.

Carriers have encouraged this trend by tariffing frame-relay services below the equivalent private-line services.

Carriers have now created integrated service offerings in an attempt to satisfy all the communications needs of large enterprises. The equipment used is frequently identical to the equivalent enterprise networks, but is owned and operated by the carrier. This is especially advantageous to enterprises in the case of global networks, which are admittedly difficult for all but the largest organisations to deploy and manage. Newly emerging global carriers are fulfilling the need for managed global network services.

2.15 Quality of Service

Quality of Service (QoS) classes are specified in the traffic contract and according to [Dem89] and the Frame Relay standards are mainly:

- Throughput
- Delay
- Bc and Be
- CIR

These are usually measured from the originating end to the destination end. The traffic contract is an agreement between the end user and the network provider that defines the set of performance standards. QoS classes are defined by measurement parameters such as cell loss rate (CLR) and cell delay variation (CDV). Each service class supports a different QoS class to meet the needs of different applications. Typically, the network provider guarantees a certain QoS for user traffic conforming to the traffic contract. A service

request may have more than one service level associated with it. Multiple service levels can be used to offer lower-priority alternatives to the customer in the event that the primary service level is not available.

Today's computer networks provide a best-effort service, which is often unreliable and unpredictable and is usually unacceptable for many emerging high-priority applications with mission-critical or real-time network requirements. Levels of network performance and function, termed here network services, can be applied to the network to support high-priority applications while still providing best-effort service to traditional, lower-priority applications. As these are “best effort services”, there are often other parameters included in the QoS contracts.

Examples of these QoS parameters are

- Lost Frames
- Delivered Frames
- Error Frames

2.15.1 Throughput

Throughput for Frame Relay is defined as the number of protocol data units (PDU) that have been successfully transferred in one direction per unit time over a PVC. In ITU-T recommendation Q.933 this interval is defined in bits per second. In this definition the PDU is considered to be all bits between the flags of the Frame Relay frame. Successful transfer means that the frame check sequence (FCS) check has acknowledged that the transfer has been completed successfully.

2.15.2 Delay

The Delay is the amount of time it takes a frame to get through the system, crossing two boundaries. When the first bit crosses the input boundary at time T_1 and the last bit of the frame crosses the output boundary at time T_2 , the delay is $T_2 - T_1$. This is simple in theory, but very difficult to measure in practice, especially when looking at end-to-end delay. In the LAN interconnection case the total network delay is made up of the LAN serialisation and access delay (CSMA/CD), delay in the transmitting router, access line delay, Frame Relay serialisation and network propagation delay.

2.15.3 Race for QoS

QoS requires allocating resources in switches and routers to allow data to get to their destination quickly, consistently and reliably. As applications increasingly demand high capacity and low delay, the QoS is becoming a top purchasing criterion for internetwork hardware buyers and a way for vendors to differentiate their products.

However, it is not always easy to understand the product literature. The QoS has its own language, and vendors use a baffling array of terms and concepts to describe how their products provide this capability.

There are only a few ways to provide QoS in networks. The simplest approach is to throw capacity at the problem, which is known as over-engineering the network. The QoS can also be provided using features and capabilities such as data prioritisation, queuing, congestion avoidance and traffic shaping. It is considered that policy-based networking will one day tie all these features together in an automated system that ensures an acceptable end-to-end QoS.

Over-engineering is the simplest and, arguably, the most effective means of ensuring QoS in the network. Pressure from competitors, new chip fabrication processes that allow a greater number of functions to be integrated into one circuit, and new manufacturing efficiencies allow switch vendors to continually offer faster products at prices comparable to existing ones. However, the problem is not solved by just increasing capacity on a link, as initial investment costs have to be secured. The QoS measures must be implemented without costly hardware upgrades or complex changes to network management. Network managers might be more inclined to consider implementing other types of QoS systems rather than relying on over-engineering.

Most likely, a combination of over-engineering and other QoS features will be implemented as the solution of choice. In the WAN, over-engineering is less practical. WAN capacity costs create the need for efficient QoS in the WAN. The WAN capacity costs will still be a significant expense for most corporations in the future.

While “best-effort performance” is acceptable to many applications and users, particularly batch or non-interactive applications, there is an emerging class of applications where it is not acceptable. This class of applications requires consistent and guaranteed performance.

Examples of this class of high-priority applications include:

- Mission-Critical Telemetry and Remote Control.
- Real-Time high-performance Imaging.
- Interactive Speech and Vision

In addition, for many environments it is desirable or necessary to provide a mechanism to account and bill for network resource usage.

Some examples of these applications, such as medical imaging, remote control of various devices, and demand access to telemetry streams, have been shown to be successful in network testbeds. Network testbeds are usually dedicated, providing dedicated capacity for their applications, which is, in effect, a guarantee of the network capacity and delay characteristics. Creating an operational network dedicated to a particular application is expensive and inefficient, and is usually not an option. Thus, these high-priority applications are integrated with existing applications onto traditional, best-effort operational networks.

However, when applications such as these are integrated into best-effort operational networks, one or both of the following events frequently occur:

1. A session of the high-priority application is successful, but causes a serious reduction of performance in the network, resulting in other application sessions "halting" or "dropping" from the network; or

2. In contending with other application sessions, this high-priority application is not successful.

Either of these events is unacceptable.

One way of solving this problem is by providing high-priority applications with the network resources they need, while allowing lower-priority applications to get traditional best-effort service, all over the same operational network. This may be done:

- In real-time or "as fast as possible", to support a user that initiates an application session, or
- Scheduled for a future time (and possible date).

However, initial investments must be protected and a compromise must be found to migrate existing technology with newer, emerging ones and use these together. Therefore it is even more important to understand the concepts of network services and QoS and how services in the network can support high-priority applications while still providing best-effort performance to other, lower-priority, applications. Defining and characterising services in the network and then describing potential network service mechanisms are important steps.

2.15.4 Network Services

A network service is a level of performance and/or function that is provided by the network to applications and users. Predictable, reliable, and guaranteed services can also be made available on the network, and these are here termed services for high-priority

users/applications. Services can be applied at user or application levels to support specific high-priority applications and users (as described above), or can be applied at protocol or organisational levels for traffic shaping to make the network more efficient. Network services can be applied at the application and user levels. Some examples of network services are:

- The guarantee of an end-to-end performance at a certain level of Capacity: Minimum Data Rate.
- The guarantee of the security function, defined in the network as open access within a user group, access between user groups on an as-needed basis, and blocked access to the Internet.

Service guarantees such as these are based on defining network characteristics that can be configured in the network elements or have to be planned and accounted for by the network planning system. For network services to be effective in supporting automatic QoS, they must follow three rules:

- Rule 1: Services are applied end-to-end, between source and destination, at all network elements in the path of the information flow. This includes the systems' device drivers, operating systems, and application interfaces.
- Rule 2: Services are configurable using QoS characteristics at each network element in the path of the application flow.

- Rule 3: Services are verifiable within the applicable network.

These rules are necessary conditions for services to be meaningful within the network and to their applications. For example, if a network service cannot be applied to all network elements in the path of the application flow, the service cannot meet its guarantees to the application/user in an automatic way.

2.16 Implementation

Signing up for a Frame Relay service is relatively easy. However customers still need to know certain parameters, which are often difficult enough to estimate.

Subscribers need to order local access lines (which can be done through the provider) and decide on a port speed on the carrier's Frame Relay switch. Port speeds range from 56 kbit/s to E1 (2 Mbit/s); access lines are typically 56/64-kbit/s or E1.

Customers also must specify the PVC, the logical link between any two sites. It should be noted that more than one PVC can be set up from a single port. They must also set a CIR, which is the guaranteed throughput for each PVC over a certain period of time. Although the carriers will help customers determine CIRs and port speeds, there is no substitute for good knowledge of network traffic patterns.

As previously mentioned, one of the things making Frame Relay so interesting is the fact that customers can get more than they pay for. Subscribers typically set their CIR at a lower level than the port speed on the Frame Relay switch. If the network has any idle capacity, data are allowed to burst up to the full speed of the port at no extra charge.

What is more, nearly all service providers let users oversubscribe ports. Oversubscription or overbooking means allowing the sum of the CIRs accessing a port to exceed the port speed. This can result in congestion since when traffic backs up at the switch, frames will be lost and retransmissions will be necessary.

Service providers recommend that oversubscription does not exceed a certain percent of port speed, but by studying traffic patterns and setting CIRs accordingly it is possible to go as high as a few hundred per cent. For example, a 64-kbit/s port can be oversubscribed with four 32-kbit/s PVCs. This is with the assumption that not all of the PVCs send at the same time. But a network manager who knows that two of those circuits are used for delay sensitive traffic might add another pair of 32-kbit/s PVCs to the port.

2.17 PVC Pricing

The first thing to realise is that dollar figures alone do not tell the whole story. Usually PVCs are unidirectional, but pricing is still for duplex connections, as two PVCs are configured between two locations. However, not all PVCs are duplex: Simplex PVCs allow traffic to be transmitted in only one direction. Users looking to send data back and forth must buy a second simplex PVC.

Some companies do not offer distance-sensitive pricing. This means that charges apply the same amount per PVC regardless of the distance between sites. Two circuits with the same CIR, one between London and Birmingham, the other between London and Glasgow, will cost the same.

Usually providers offer fixed monthly fees for unlimited usage. Some companies also offer usage-based billing. With this option, the carriers measure how much data are being transmitted over the network and bill according to a per-kilobyte fee. This could be advantageous to customers who do not expect to ship much data.

2.18 PVC without CIR

Some companies offer a service on their Frame Relay network known as “zero CIR”. This PVC has no guaranteed throughput. However, a PVC without a CIR is providing the customer with unknown risks: If the network is congested, a file being transmitted over a zero-CIR circuit may be dropped. Especially on busy routes this option can result in very poor performance and upset customers.

The main benefit of zero CIR is saving: It can save up to 40 per cent off the price of an equivalent leased line. That makes it very attractive for small, remote sites with delay-insensitive applications like file backup and transfers, which actually make up a large percentage of the PVCs. Not all service providers offer zero CIR as the QoS cannot be guaranteed and as previously mentioned customers could receive some unexpected negative surprises. One carrier actually encourages customers to go with zero CIR and estimates that 90 per cent of the ports on its Frame Relay network subscribe to zero CIR.

Chapter 3: Congestion Control Mechanisms

3.1 Introduction

A problem with Frame Relay implementations in non-ISDN networks, like the proprietary Frame Relay backbone used by AT&T, is that there is no call set-up phase with QoS parameter negotiation. With ISDN-implementations, QoS parameters are negotiated during the set-up phase of a call. These QoS parameters are controlled by the congestion management system of the ISDN network so that the negotiated QoS depends on the state of congestion of the network. This is necessary, as Frame Relay is developed to show capacity-on-demand features by allowing a user to write a spare capacity of another silent user.

The Frame Relay implementation of the proprietary system used in the AT&T network differs in many ways from the ITU-T defined standards. First of all, the congestion control bits (FECN and BECN) are used as in the standards. The backbone, however, does not rely on the end user's response to these bits. This is because according to the ITU-T standards, users are not obliged to react to congestion notification. Instead, the proprietary congestion control mechanism uses different bits in the packet header to take care of congestion management inside the backbone. It is a closed-loop congestion control mechanism called PM.

When a LAN end user wants to send information to another LAN end user connected via a backbone the following occurs. The LAN protocol puts an address on the LAN frame to be sent in parcels into the router. The router encapsulates the LAN frame in one or more Frame Relay frames and puts it on the access line on a permanent circuit reserved for the

LAN connection. At the switch, this frame is split up into fixed length Fixedpackets. These Fixedpackets are transported along the fixed route of the PVC on a trunk to the destination switch. Here the Fixedpackets are reassembled into a Frame Relay frame. At the receiving route these Frame Relay frames are stripped to LAN frames and finally sent to the end user. It has to be noted that the interface of the Frame Relay card is compliant with the ITU-T standards. The network itself, however, has to be viewed as a black box. In this chapter a closer look is taken at the architecture of the backbone switches, trunks, and PVC in the Fixedpacket format to be able to present a model of the IPX switch and PM. The information presented here is based on the IPX manuals and meetings with employees of AT&T and a personal interpretation of the IPX and PM system. When modelling in this reverse engineering process an attempt is made to stay as close to reality as possible.

3.2 Congestion

The concept of congestion in the network layer is a very simple one. The performance of any system will degrade if the amount of work that the system is forced to do is more than it can cope with. In this context, if there are too many packets present in a given part of the network, it is said that the network is congested. This situation is shown graphically in Figure 3.1.

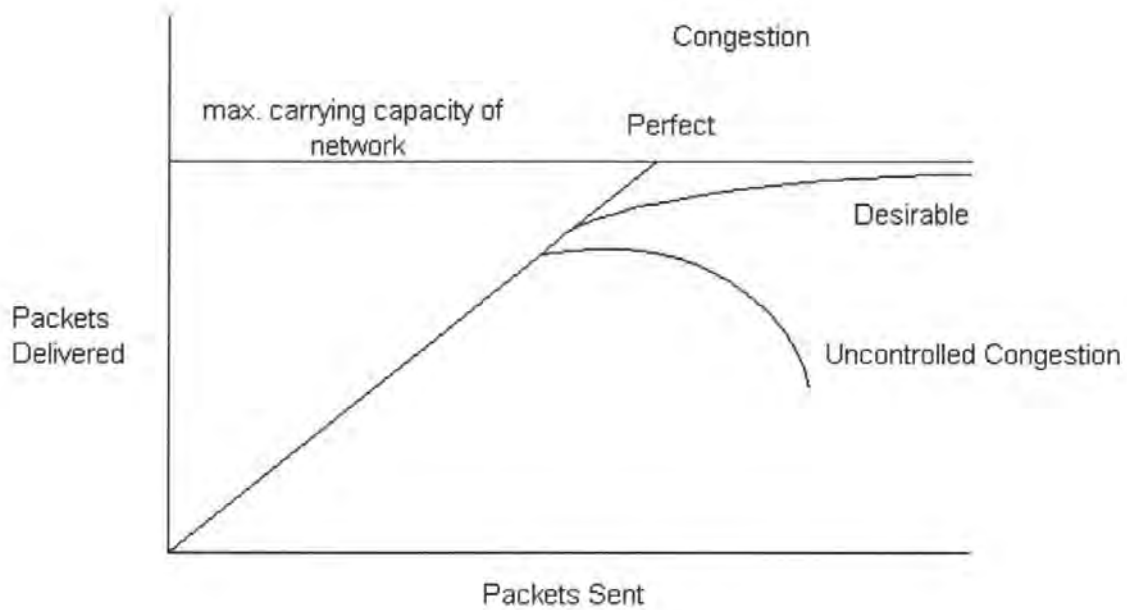


Figure 3.1: Congestion and Effects on Throughput

The obvious question needing to be asked is how does such congestion come about. This is essentially concerned with the number of packets that are being forwarded within the system. The system has usually a large amount of sources, which are varying their input into the system, and, therefore, the load in the system is not easily predictable.

Regardless of the line capacity, if the mechanisms, which are dealing with the congestion, are too slow or too aggressive, then queues will build up and create delay. It is clear from the diagram that performance degrades very sharply when congestion crosses a certain level.

The aim is usually to keep congestion as much at the edge as possible, to use resources in the best possible way. Congestion management is therefore a search for a compromise between too much delay and under-utilised capacity.

There is a knock-on effect caused by congestion that must be considered. It is rather like what happens with a set of stacked dominoes. Since the congested link is not acknowledging receipt of packets, the source that is sending them is unable to free up its own much needed buffers (since it must keep trying to send the packets). As a result, congestion could back up across the whole network.

Obviously, there must be ways of reducing the effects and minimising the possibility of congestion occurring. This area is known as congestion control.

There are many different ways to control congestion. Control systems have been present for as long as life itself. Proper functioning of the biological systems clearly requires controls of a more or less complicated nature. A simple example of a manually controlled system is the manoeuvring of an automobile. The vehicle operator in a closed-loop fashion continuously exerts control over various outputs of the system, such as velocities and orientations of the car in a traffic lane. Outside of this closed system is another control mechanism which can be applied. A traffic light with sensors aims by means of different colours and traffic patterns to control vehicles and pedestrians and their speeds, priorities etc., which can be an example of another automatic control system.

3.2.1 Flow Control

Before discussing flow control as a method of dealing with congestion, it must be pointed out that there is in fact quite a difference between congestion control and flow control. Congestion control can be viewed as a means of making sure that the entire subnet is able to handle the given traffic. Flow control, on the other hand, refers to the handling ability between a specific source and destination.

The basic principle behind flow control is that it limits each user of the subnet to the mean rate of traffic. Flow control is used to control the traffic sources so that they do not send too much data into the network at any moment. If a trunk in the network is overloading, all the sources using that link must be told to slow down. This property is also its major failing. The flow control method does not allow for bursts of heavy traffic, which are quite common when dealing with human users. What this means is that at any given time, a user may decide to do something over the network, which has a very high overhead in terms of transmission and so produces congestion. Major differences between flow control techniques are the time it takes to tell the source about congestion and the time it takes to get it under control.

There are two major factors involved in determining the control-time of a flow control loop.

- The propagation latency of the control mechanism
- The oscillation cycle of the control mechanism.

With this in mind, certain networks have used flow control as their method of controlling congestion. Internet, with their TCP/IP protocol is one such network.

TCP/IP protocol operates a binary rate flow control algorithm between the two end-stations. TCP is slowing down the data flow at the source if a returned packet has indicated that data was lost, otherwise it slowly speeds up the flow. TCP/IP by losing data each cycle in an overload situation and under-utilising the link on the other part of the cycle. It takes TCP about one second to reduce the data flow after detecting increased congestion. This is a major negative aspect of TCP and as long as its flow control is only implemented in the end-stations its performance cannot be substantially improved.

One of the most important mechanisms is Explicit Rate Flow Control. This flow control scheme is used in many modern technologies like Frame Relay. Explicit Rate Flow Control will be discussed more in detail for the proprietary adoption of Frame Relay technology.

In the last decades the control of network traffic has emerged as an important area of technological and scientific research. It usually addresses the point-to-point and multi-point control problems. It is expected that the multi-point control problems start to occur more regularly as movies-on-demand and live video streams become an ever increasing part of the Internet use. The traffic characteristics of the new applications in today's network environments are different from those of traditional applications in another way. Whether its moving documents and files between locations, transferring an image, or

accessing a remote server. These new applications tend to periodically demand higher capacity, especially when a serving facility is accessed by a transaction that requires all or much of the available capacity, such as an image transfer or a database query. Because of these new applications, flow control becomes plays an ever increasing part the design of new network technologies.

3.2.2 Different Mechanism Philosophies

As a result of longer lasting network congestion, the efficiency, i.e. the throughput of a packet-switched network, drops drastically. Congestion cannot be avoided through the use of larger buffers, faster processors and high-speed links, but the effects of congestion can be limited. For an efficient and fair congestion control, scheduling and buffer management (router-based congestion control) is necessary. It has to be combined with feedback to the network clients and end adjustment of the sources (source-based congestion control). Router-based congestion control is suitable to handle short-lasting congestion and to facilitate a fair best-effort service. Source-based congestion control is needed to deal with longer-lasting congestion. Sources are expected to adjust their sending rate according to the network's feedback. In addition, the discarding mechanism in the buffers is important, as it affects the control mechanism in a very direct way.

The earlier efforts of traffic control in communication networks were based on buffer management algorithms for flow control at the link level. The slide-window scheme of

flow control was implemented on many networks (APARNET, TYMNET, and DECNET) for controlling the transmission rate of node-to-node or end-to-end [Ger80]. However, the buffer based mechanism of flow control is not effective in preventing congestion to occur when the communication traffic becomes abnormally high at some hot-spots of a network.

3.2.3 Fairness

Rate-based applications are those which do not use a congestion window to control the amount of data outstanding in the network; rather, they choose their sending rate based on what is appropriate for the application. For example, voice applications send at rates appropriate to the real-time transmission of voice communications.

These applications also make TCP extremely susceptible to capacity stealing by other applications, which do not implement any congestion-control techniques, and poor performance when multiple TCP segments are lost. The widespread development of new, non-TCP based applications poses two major threats to the Internet. First, these applications could contribute to a new congestive collapse of the Internet. Second, these applications will consume an unfairly large portion of resources when run side-by-side with "good-neighbour" TCP applications.

Most rate-based applications have some latitude in their choice of data rates. In order to run these applications over the same path as TCP applications, a TCP-friendly congestion control algorithm should be implemented into these applications. These applications

should simply choose to send at a rate no higher than a TCP connection would achieve when operating along the same path.

3.2.4 Beginnings of Congestion Control

One of the first methods attempts to tackle the fundamental reason for congestion within the subnet and is called Isarithmic Congestion Control. Proposed by Davies in 1972, this method says that if congestion is caused by there being too many packets in the subnet, then it can be solved by setting an upper limit on the number allowed to be present at any given time. He approaches this by introducing a system of permits within the subnet. If the source wishes to transmit a packet, it must first obtain and destroy one of the permits, which are circulating within the system. Once the packet has been delivered to its destination, the receiver regenerates the permit back into the subnet.

The isarithmic congestion control method is the base for many implemented congestion control mechanisms. However, it has some great weaknesses, as the method does nothing about the fact that individual switches can become congested. Even though there are a limited number of packets allowed in the subnet, all of these could conceivably be going to the same switch, causing congestion. Nevertheless, this method opened research to new ideas and formed a good starting point for further research.

Thus, permit management and individual congestion problems at switches mean that the isarithmic congestion control method is the first step for a suitable solution to the problem.

Since the introduction of the isarithmic congestion control mechanism, research became a high priority in network design and research due to ever-growing network capacity and intensive network applications.

Many congestion control strategies have been proposed based on various concepts. In general a congestion control scheme is a control policy which is designed to achieve given goals in a distributed network environment.

However, it is often difficult to characterise and compare various features among different congestion control schemes. The characteristics of how each algorithm extracts information for their control decisions are used as the basis for the two main classifications which are:

- Open-loop congestion control algorithms.
- Closed-loop congestion control algorithms.

Several subcategories can exist under each category but only the two main categories are described in this section.

3.2.5 Open Loop Congestion Control

Open-loop congestion control mechanisms do not depend on feedback information from the congested spots in the network. The control decisions are made on the knowledge of local parameters like link capacity. They also do not monitor the state of the network

cloud. The congestion control algorithm serves purely based on its own knowledge of local parameters like available buffers in the access switch. As open-loop schemes are not aware of traffic conditions in the network, they tend to either control the rate of flow at the sources of traffic, or intend to control the network traffic at the destination switch. In cases where the access mechanism is unsuccessful and the traffic in the network creates congestion, the network discards data at the point of congestion regardless of any future arrival of traffic. Another problem with open loop mechanism is that the network cannot access spare capacity to fairly distribute it among users. Figure 3.2 shows the information flow of these mechanisms.

Subcategories under the open loop congestion control are:

- the open loop with source control algorithm, and
- the open loop with destination control algorithm.

The open loop with source control algorithm imposes the control on traffic at the source end and uses mainly the local knowledge of the network. Algorithms included in this category are the Virtual Clock scheme [Zha90], and the bit-round fair queuing method [Dem89].

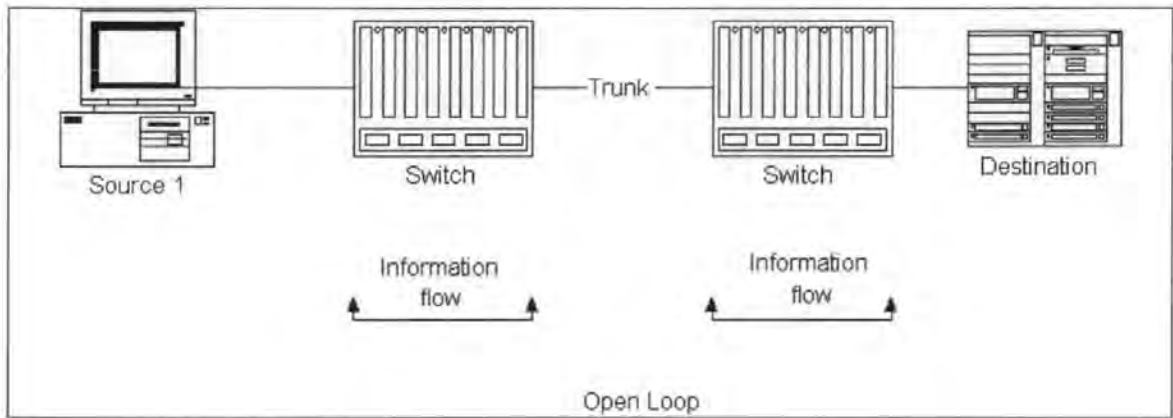


Figure 3.2: Open Loop Control Mechanisms

The Open Loop with Destination Control algorithms control operations at the destination end without any knowledge of feedback. Algorithms included in this category are the selective packet discarding schemes [Yin90], the isarithmic control policy [Dav72], and the packet discarding scheme by [Tan81].

3.2.6 Closed Loop Congestion Control

Closed loop congestion control algorithms make their control decisions based on the monitoring of the capacity within the network with a mechanism, and sends information back to the source indicating congestion levels. This allows allocation of spare capacity, knowing the data will reach its destination. The allocation of capacity is usually based on the end-to-end network availability. Feedback information can be local or global: local means the feedback information comes only from immediate neighbours and global means the feedback information goes all the way from destination to source. The advantages of

closed loop congestion mechanisms are lower probabilities of network congestion, fairer allocation of capacity inside the network for shared services, and more effective capacity utilisation. The biggest disadvantage is the implementation of additional control logic for the feedback information. Figure 3. 3 shows the general information flow of the closed loop mechanisms.

Subcategories under the closed loop congestion control are:

- the Implicit Feedback,
- the Persistent Global Feedback,
- the Persistent Local Feedback,
- the Responsive Feedback

Closed Loop Control with Implicit Feedback algorithms maintain control through feedback information between destination and source. The feedback is based on explicit information regarding the traffic conditions in the network, e.g. missing packets or timeouts. The slow-start scheme of Jacobson [Jac88] is the best known in this category as it is used by the Internet.

The Closed Loop Control with Persistent Global Feedback algorithms control traffic with feedback information between destination and source on a periodic basis, i.e. the adaptive admission congestion control scheme [Haa91] and the binary feedback scheme [Ram90], and the Q-bit control scheme [Ros92].

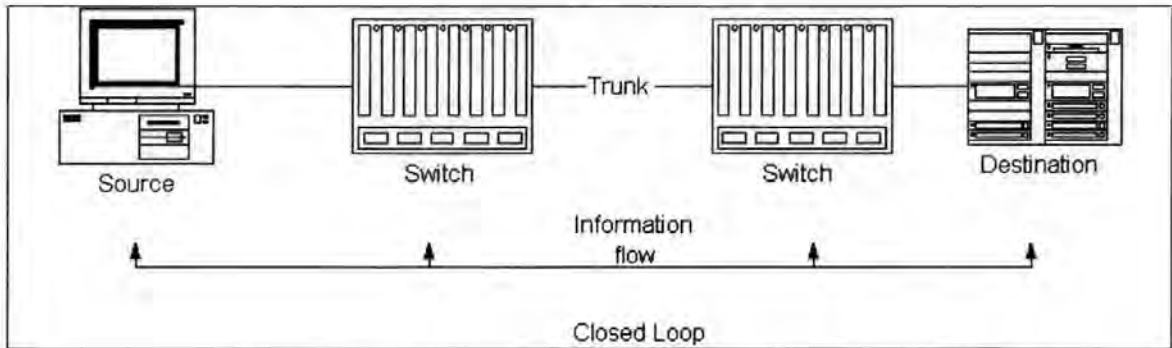


Figure 3. 3: Closed Loop congestion Control Mechanisms

The Closed Loop Control with Persistent Local Feedback algorithms control congestion through feedback information between adjacent nodes. The feedback information is constantly sent between the nodes. The hop-by-hop control scheme [Mis92] is such an algorithm.

The Closed Loop Control with Responsive Feedback algorithms control traffic by triggering information feedback when a certain network condition has been reached. The feedback information is either sent between direct neighbours or the destination and the source. Mechanisms under this category are the source quench scheme [Tan81] and the rate-based congestion control [Com90].

3.3 Congestion Control in Frame Relay

Frame Relay standard utilises bits in the header to indicate the presence of congestion in the network. Notification of congestion conditions may be sent by the network to the

access devices through the use of Forward and Backward Explicit Congestion Notification bits (FECN and BECN). If there is a packet passing by a congested queue in the reverse direction, that packet is also tagged with this information; this is called BECN, or Backward Explicit Congestion Notification. Access devices are responsible for restricting data flow under congestion conditions. In order to manage congestion and fairness, frames may be selectively tagged for discard with the Discard Eligibility bit (DE bit). This may happen, when traffic sent at a rate exceeding a certain threshold. The Frame Relay standard does not require the implementation of flow control techniques, as these are vendor-specific issues. For the user, the core Frame Relay network is a “black box” which has different vendor-specific congestion control algorithms, and therefore product performance differences.

This scheme of congestion control in Frame Relay can be classified as closed-loop control with explicit, responsive, global feedback. There are many other congestion control strategies for frame-relay networks that fall into the categories of open-loop control, implicit feedback, etc. [Dos88, Ger89].

Window-based control scheme [Dos88] is an open-loop scheme, with a source control system. The protocol reserves a number of buffers for the entire window size per virtual circuit (determined at call set-up time). The call set-up control limits the number of virtual circuits created. Implicit priority is given to delay-sensitive short frames. No feedback is required in this scheme.

3.3.1 Explicit Congestion Notification

Explicit Congestion Notification (ECN) is a form of flow control to signal the onset of network congestion to external devices in a Frame Relay network. These external devices are usually routers connected to the Frame Relay switches. ECN detects congestion primarily at either the source or at the destination of network permanent virtual circuits.

To be effective, external Frame Relay devices should respond correctly to control the rate at which they send information to the network. This feature results in data transmission at the optimum rate for the channel and reduced possibility of packet loss due to excess bursts by the external device.

3.3.2 Source ECN

Network congestion usually occurs at the source of traffic. This can occur when the traffic being generated by the source device momentarily exceeds the allocated network capacity. ECN can be used at the source to relieve congestion at this point.

For this example, frames originating at the left-hand side of Figure 3.4 (page 58) and arriving at the destination user device are queued at the input buffer in the Frame Relay Port (FRP). The FRP monitors the depth of the data in the input buffer for each PVC.

When the Frame Relay data in this queue exceeds a pre-set threshold, the FRP declares that this particular PVC is congested.

When congestion is declared, the forward direction is notified by setting the FECN bit in the data header of all outgoing data frames towards the destination. This will be detected by the destination user device (the router) and may or may not take action to limit the data being applied to the network [STRATA].

At the same time, the BECN bit is set in all outgoing data frames for this PVC towards the source device to notify equipment in the reverse (backwards) direction. The source device may be able to restrict the rate at which it transmits frames in an attempt to reduce the congestion.

3.3.3 Destination ECN

In a similar manner, the two ECN bits may also be set as a result of congestion detected at the destination side of a network. This may result when a large number of PVCs from all over the network all terminate on a single FRP port at the destination node. It is for this example looked at what happens at destination node.

As frames are received from the source-user device they are queued at the output buffer in the FRP at the destination node. The FRP monitors the depth of the output buffer for each

port on the FRP. When the Frame Relay data in this queue exceeds a pre-set threshold, the FRP declares that this particular port is congested [STRATA].

When congestion is detected, the FRP sets all FECN bits in frames for all PVCs transmitted in the forward direction to the destination user device as well as all BECN bits in frames for all PVCs terminating at this port in the network. The net effect is approximately the same except the ECN mechanism affects all PVCs on a port at the destination whereas source ECN affects only individual congested PVCs.

3.3.4 Discard Eligibility

Under conditions of network overload, it may become necessary for the Frame Relay service to discard frames of user data. Discard Eligibility is a mechanism defined in Frame Relay standards that enables a user device to mark certain frames to be discarded preferentially if this becomes necessary. This frame discard mechanism can be disabled on a port by port basis by software command.

One bit in the Frame Relay frame header (DE) is used to indicate frames eligible for discard. User devices may optionally set this bit in some of the frames that they generate. This indicates to the network that it is necessary to discard first the frames with the DE bit set to prevent congestion.

User Frame Relay data are buffered at the entry point to the network in a separate buffer for each Frame Relay port.

When configuring a port, the network operator enters a “DE Threshold” parameter, specified as a percentage of the input buffer. During normal operation, when the PVC buffer is filled above this threshold, any new frames received with DE set will be immediately discarded at the port. Only frames not marked DE will be accepted and queued for transmission. This has the effect of discarding frames before they are applied to the network where they may cause congestion. This function, however, is effective only if the user sets the DE bit.

3.4 Proprietary Congestion Control Mechanism: PM

The AT&T network uses PM, which is an optional software feature that provides a closed-loop feedback mechanism for controlling the rate at which users can apply data to the network. It improves the efficiency of the network for carrying Frame Relay data, especially during periods of light usage and maintains consistent performance during peak loading without dropping frames. PM controls the data flow at the access ports (FRP), and is not dependent on the user end devices as is congestion prevention using FECN and BECN.

PM provides congestion avoidance by monitoring the transmission of Fixedpackets carrying Frame Relay data throughout the network and adjusting the rate at which data are allowed to enter the network. PM allows the FRP card to send packets at a rate that varies

between a minimum and a maximum based on the state of congestion in the network along the route.

If the initial Fixedpackets do not experience any congestion on the network, the information rate is stepped up in small increments towards a maximum set by the Peak Information Rate (PIR).

If a node processes the Fixedpackets where there is congestion (trunk card buffers close to being full), an FECN bit in the Fixedpacket header is set. When this packet is received at the destination node, the FECN bit is examined by the destination FRP card. The far end FRP card then sends a message in the reverse direction to indicate the congestion to the near end node. When the congestion message is received at the source FRP, the data rate is reduced in larger increments towards a Minimum Information Rate (MIR), the minimum guaranteed data rate for the connection. The FRP restricts the capacity allocated to the connection by reducing the rate at which it issues credits to the PVC.

3.4.1 Credit Manager

The Credit Manager is the heart of the proprietary congestion mechanism. The connection transmits and receives data rates, which are controlled separately as the congestion may be present in one direction but not the other. The data rate allocated to the connection is directly controlled by the rate at which the Credit Manager allocates credits. The credit allocation is always between MIR in congestion times and rises slowly to PIR when capacity is available.

With PM the credits are allocated dynamically, providing a much better utilisation of network capacity. The net effect is to provide greater throughput for bursty connections when the network has available capacity, yet preventing congestion on the network when the extra rate cannot be accommodated.

PM provides congestion avoidance by monitoring the transmission of Fixedpackets carrying Frame Relay data throughout the network and adjusting the rate at which data are allowed to enter the network. PM allows the interface to send Fixedpackets at a rate that varies between a minimum and a maximum based on the state of congestion in the network along the route.

The Credit Manager is the basis of congestion avoidance and notification within PM. The flow of data are actively regulated into the network by the Credit Manager from each Frame Relay virtual circuit. Data are admitted to the network at a certain rate dependent on the current state of resources within the network and parameters assigned to the virtual circuit by the network administrator.

The size of initial bursts of Frame Relay data are also limited by Credit Manager software control. This limit is set by the network manager and can have different implications for the network. If the limit is set too high, then too many Fixedpackets can overflow the buffers and packets drop. Setting the limit too low will not take account of the nature of

Frame Relay and bursts are handled more like in a time division multiplexing (TDM) system and frames are not let into the network quickly enough.

The network exchanges one Fixedpacket for one “credit” every time a Fixedpacket is entering the network. In order to achieve its configured minimum capacity, credits are accrued by each connection at a rate sufficient to allow it to send the required number of packets. It is allowed to accumulate a limited number of credits for future use should a connection not need these to send packets immediately.

Cmax provides this maximum credit value in packets to a connection. Credits are continuously accumulated by a connection at a rate up to a maximum of Cmax. Therefore, it also represents the maximum number of Fixedpackets that may be sent in rapid succession by a connection. A burst of Fixedpackets is transmitted as long as credits are available, whenever credits are available. Upon the connection having used all available credits it is required to cease sending packets until a further credit has been awarded. At this moment Fixedpackets are basically transmitted at the rate of the credit manager providing credits, which depends on the state of the network and purchased CIR. In the meantime other connections are allowed to send a Fixedpacket into the network in exchange for one credit. This goes in strict rotation.

If a connection is silent, it can collect tokens. The maximum number of tokens which can be collected is limited by the Cmax. Cmax can be set by the network provider, to enable a connection an initial burst rate. As Frames received from the user equipment are broken

into multiple Fixedpackets, Cmax is typically set to the number of packets resulting from the average frame size. However, the research displayed in Chapter 5 has shown that the packet size distribution of Ethernet LANs is often bimodal, and the average frame size does not take account of this distribution. The theory behind this is that frames should be allowed into the network, without any unnecessary delays in the Frame Relay Port (FRP).

3.4.2 The Frame Relay Port Card

The Frame Relay Port Card (FRP) is the interface between the access line and the switch. The card can handle both input frames from the router and the Fixedpackets from the backbone. The scenario will be discussed as frames arrive from the router. The input frames from the router are first checked for CRC errors and the correct length. If an error is detected, the frame is immediately discarded. If the frame length is out of the range 5 to 4096, the frame is also rejected. The third possibility, when a frame is not admitted into the network, has to do with congestion control when the DE bit is set and the Virtual Circuit (VC) queue level is above the DE-threshold for frames. After these checks, the frame is put into an input buffer called VC-queue at full Access Rate (AR). Each PVC has its own VC-queue in which it can accumulate Frames in FIFO way.

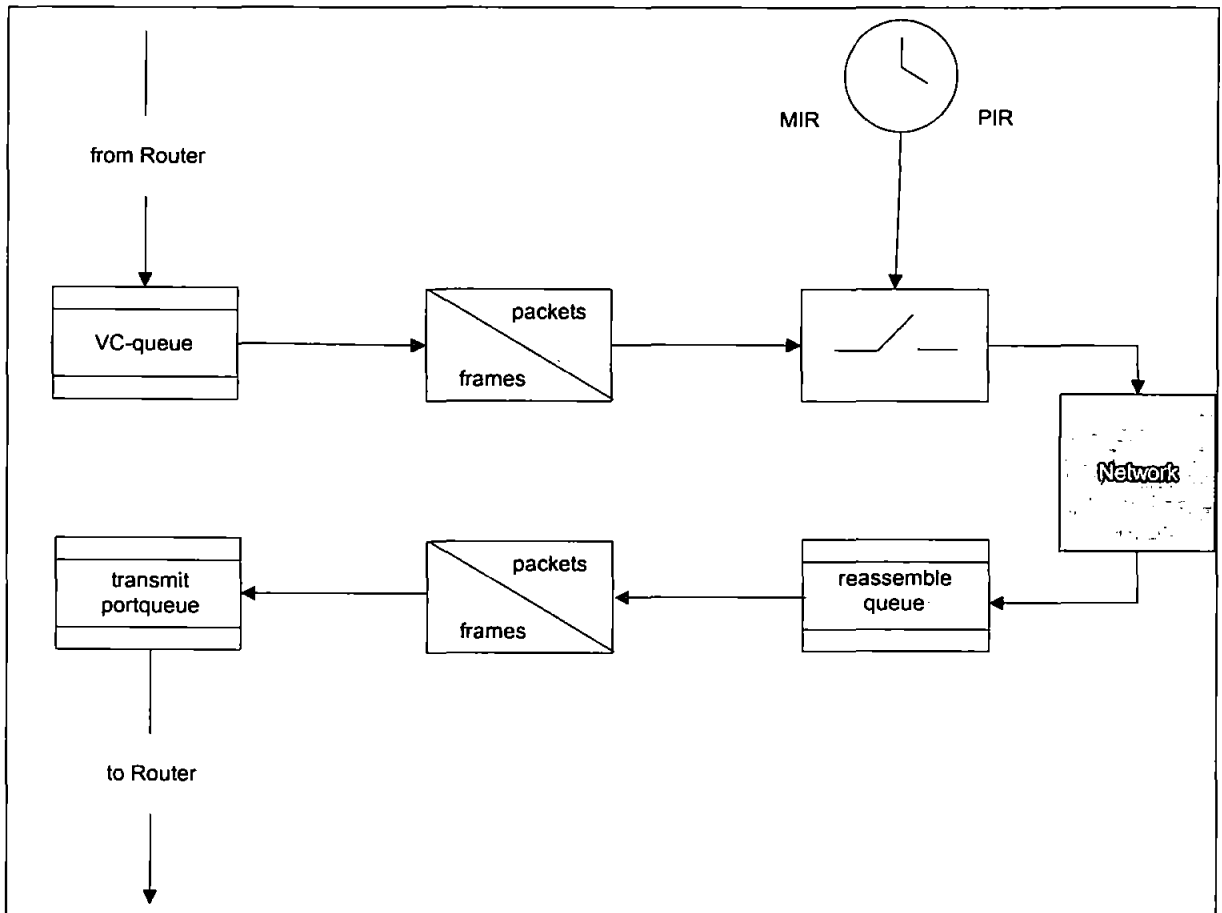


Figure 3.4: Frame Relay Port Card

When the network is ready to serve the VC-queue, the frames are first split into 24 byte Fixedpackets. The logical rate at which Fixedpackets are taken out from the VC-queue is the Credit Manager Rate (CMR). The CMR is dependent on the state of congestion of the network.

When the switch receives Fixedpackets from the network it clocks these onto the FRP card where these are reassembled. All the Fixedpackets resulting from the same Frame have followed the same route through the network on the PVC. This means that they all come

out sequentially and do not have a number sequence. If the reassemble queue is filled with a frame, the frame is put into the Transmit port queue after checking the frames' CRC and the frame length. A special end of frame Fixedpacket exists for the indication of the end of frame. The frame is discarded if the CRC is incorrect or if there are Fixedpackets missing. Notice that unlike the VC-queue, which is specified for every PVC, the Transmit port queue buffer is shared between all PVCs ending on this particular port.

3.4.3 The Network Trunk Card

The network trunk card (NTC) is the interface between the switch and the trunk. Each NTC in a node co-ordinates the sending or receiving of Fixedpackets on a trunk to another NTC at the far end of the trunk. Each trunk has its own pair of NTCs whereby each NTC has its own in and output buffer. The input-buffer is merely required for synchronisation on the internal databus, which serves in TDM fashion. The output buffer is needed due to the fact that the speed of all possible incoming traffic from all NTCs of all PVC can exceed the trunk speed and traffic must be stored before sent. This buffer is the BD-B buffer.

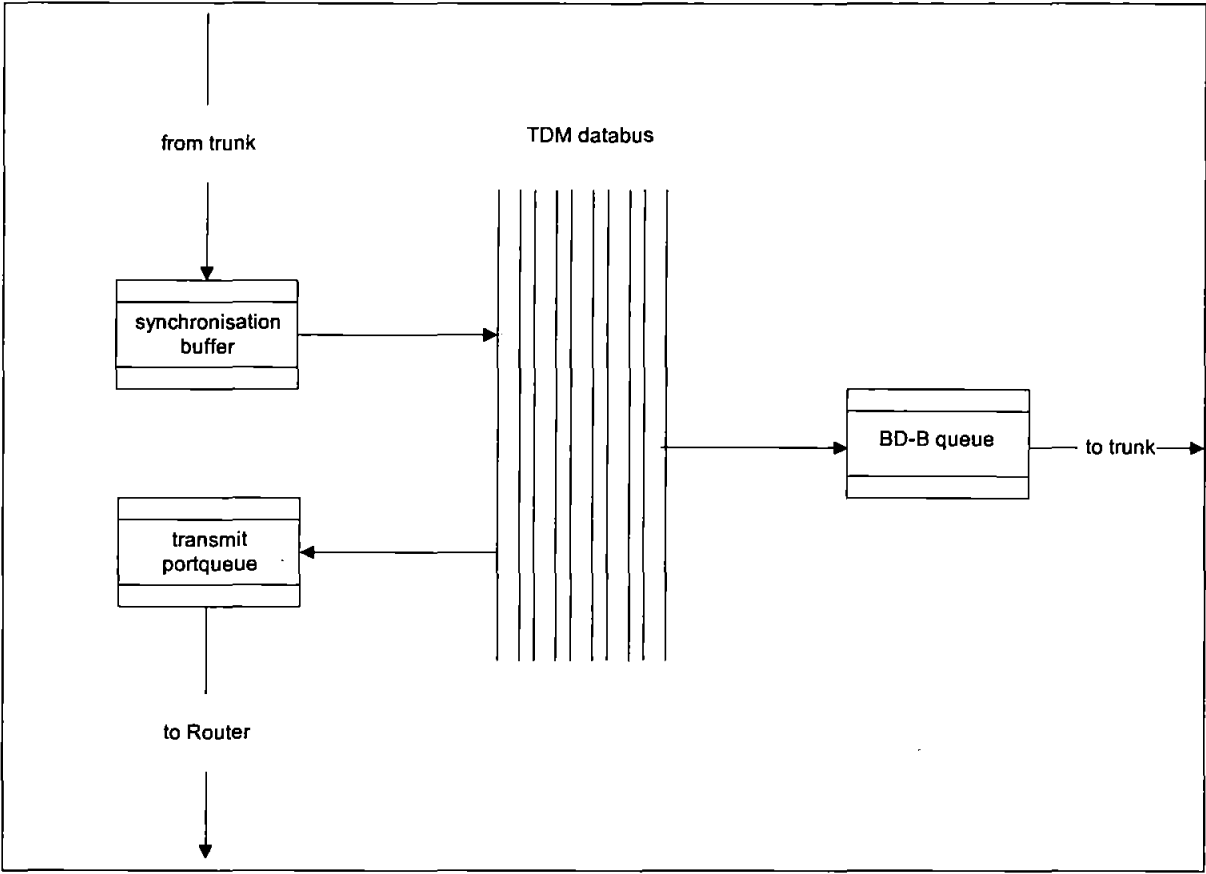


Figure 3.5: Network Trunk Card

Looking at Figure 3.5, it can be seen that the traffic flows are switched from a trunk via the synchronisation buffer to the transmit port queue out of an FRP and then to a router or to the BD-B queue of an NTC onto a trunk. Both output buffers can overflow because they are downstream from the TDM databus. The TDM databus can switch data at much higher speed to the output lines than the output lines could serve the queue, so these buffers have to be guarded for overflow by the congestion management system.

3.4.4 Round Trip Delay

As part of the calculation of the PM algorithm, the switch measures the round trip delay for each PM connection in the network. This delay is measured automatically by sending a special test packet over the connection. The far end FRP returns the packet and the round trip delay is measured when this packet is received at the near end. This delay essentially consists of transmission delay, as the test packet is a high-priority packet type and experiences minimal processing and queuing delays. The network delay is measured periodically.

3.4.5 CIR in PM

CIR is generally specified by the usage subscribed to, or purchased, by the user. In the proprietary system used by AT&T, the CIR is a value which the network provider commits to an average throughput over a set period of time. This means that the throughput is between a minimum rate and a maximum rate, but if the user sends consistently, then the average throughput of the CIR will be presented. If there is no congestion on the network, the user will experience higher throughput. The system uses CIR to determine the setting of the DE bit in the Frame Relay frame and the CLP bit in the Fixedpacket header.

3.4.6 FECN and BECN

With Explicit Congestion Notification, the FRP card signals network congestion to external devices by setting bits in the Frame Relay data frame header. These bits, BECN and FECN, are used as defined by the ITU-T standard and were explained in the previous section. The external devices detect the number of frames received with these BECN and/or FECN bits set and should adjust their transmitting rate up or down accordingly. The BECN and FECN bits are moved into Fixedpacket headers.

3.4.7 Cell Loss Priority

CLP is a feature that allows Fixedpackets to be discarded in a selective manner. If the network is congested and packets must be discarded, packets with the CLP bit set will be discarded before other packets. The CLP bit is located in the control byte of the Fixedpacket header and is just used inside the network.

The switch sets the CLP bit for either of two conditions:

- For all Fixedpackets associated with frames where the user device has set the DE bit.
- For Fixedpackets transmitted above the CIR

There are two CLP thresholds associated with switch data queues, a high threshold and a low threshold. If the high threshold in the queue is exceeded, Fixedpackets or cells will be

discarded to prevent network congestion. They will continue to be discarded until the queue drops below the low threshold. This is a disadvantage and unfair, as it is not weighted depending on which PVC uses more of its fair share.

3.5 TCP/IP Congestion Control and Avoidance

TCP's ability to recover from multiple packet loss in the same window without retransmission timeout is unsatisfactory. Furthermore, TCP's estimate of the amount of data outstanding in the network could be improved to allow a more accurate congestion control in a recovery phase. TCP's congestion control is based on packet loss as implicit feedback. Therefore, at least one packet has to be lost before TCP is able to detect and react to congestion. Congestion control using other implicit feedback like changes in throughput could allow handling congestion before packet loss occurs. However, this packet loss should happen before it enters the Frame Relay network. A packet loss at the router would ensure that no unnecessary packets are travelling through the network and the Congestion Mechanism of TCP would still work. This, however, can have other disadvantages mainly by forcing the traffic to stay outside the network and creating unnecessary delay.

Applications that do not seriously consider congestion issues can contribute to widespread congestive collapse in the Internet. For these reasons it is vitally important that all applications implement some form of congestion control. This ensures fairness among users as well.

There are severe consequences to competing unfairly with TCP. Under heavy loads, TCP will back off, reducing its capacity utilisation. In a later chapter it will be shown more precisely what damage an unbalanced flow can do to TCP, by driving up the packet drop rate.

The basic algorithm incorporated by TCP is Congestion Avoidance [Jac88]. The Congestion Avoidance algorithm probes for available network capacity by slowly increasing a congestion window (used to control how much data are outstanding in the network). When congestion is detected (indicated by the loss of one or more packets), TCP reduces the congestion window by half. This rapid backoff in response to congestion is the key to TCP's success in avoiding congestive collapse in the Internet.

3.6 Conclusion

It was discussed that a network is a distributed control system, in which a congestion control schemes and control policies are executed in order to maintain a certain level of stable conditions. It was described that in a congestion situation the response time rises slowly with the load due to the fast increment of the throughput. Then after the knee point is reached, the delay curve jumps significantly while the throughput stays flat. Finally, the delay grows indefinitely when the network becomes congested.

In order to continuously maintain a network in a healthy working condition, certain measures or mechanisms have to be provided to prevent the network from operating in the congested region for any significant period of time. The proprietary control mechanism PM was described and it was also shown that different components in a network, including the host machines of sources and destination parameters, as well as switching nodes are involved in the congestion control process.

Many congestion control algorithms have been proposed and developed but none has taken existing traffic profiles into consideration. The strategy of congestion avoidance is preventative in nature and is aimed to keep the operation of a network so that congestion does not get out of control. A number of different congestion control algorithms have been proposed and developed, ranging from Random Drop [Man90], Source Quench [Fin89], Isarithmic scheme [Dav72], Slow Start and Search [Jac88, Wan91], Virtual Clock [Zha90], Binary Feedback [Ram90], to rate-based congestion control [Com90]. All these algorithms vary in terms of their operating conditions, functional principles, and performance behaviours.

Network control systems can be very complex but the implementation is based purely on reaction to the occurred traffic condition rather than action to the anticipated traffic volumes. The development of these congestion control policies has still not solved the underlying problem: without the consideration of the existing traffic profiles no knowledgeable action can be taken. The research in this thesis will therefore use the

described mechanism PM to develop a new congestion control technique which will consider specific traffic profiles to control traffic more effectively.

Chapter 4: Network Analysis, Models and Tools

4.1 Introduction

Various tools have been used as an implementation technique for models for quite some time. Different implementations of models rely on the ability of the modeller to describe a model in mathematical terms. Typically, as a network system can be viewed as a collection of queues with service, wait, and arrival times defined analytically, queuing analysis can be applied to solve problems, which occur there.

Some of the major reasons to analyse models are:

1. Such models capture essential features of systems like
 - queuing delays;
 - service times; and
 - arrival times.
2. Assumptions or analysis are realistic in relation to the real world.
3. Algorithms and processes are available in machine form to speed up the analysis.

4.2 Queuing Models

Queuing models provide a concise way to develop analysis of queuing-based systems.

Queues are waiting lines and queuing theory is the study of waiting line dynamics.

Queuing models operate as follows:

An arrival comes into the queue. If the server is busy, the customer is put in a waiting line, and depending on the buffer size of the queue, the customer is either rejected if the queue is full, or has to queue in the waiting line till the previous arrivals have been served or if the arrival has a higher priority it is served first. If the queue is empty, the customer is brought into the service location and is processed by the server. The delay occurring through this serving is the service rate time.

Additionally, the policy it uses for accepting and removing customers is also defined. Examples of queuing disciplines typically used are “first in first out” (FIFO) and “last in first out” (LIFO).

4.3 Queuing Systems

There are many cases in data communications and computer networking when it is important to predict the effect of changes in a design or the impact of different protocols and applications applied to the system. Very often the load on a system is expected to increase and therefore design change is expected or may be necessary.

For example, consider an organisation that supports a number of terminals, personal computers, and workstations on a 10 Mbps Ethernet LAN. An additional department in the building is to be cut over onto the network. Can the existing LAN handle the increased workload, or would it be better to provide a second LAN segment with a bridge or router between the two?

There are other cases in which no facility exists, but on the basis of expected demand, a system design needs to be created. For instance, a department intends to equip all of its

personnel with a personal computer and to configure these into a LAN with a file server. Based on experience elsewhere in the company, the load generated by each PC could be estimated. The concern is system performance. In an interactive or real-time application, often the parameter of concern is response time. In other cases, throughput is the principal issue. In any case, projections of performance are to be made on the basis of the existing load information or on the basis of estimated load for a new environment [Nuss90].

4.4 Poisson Distribution

The Poisson distribution is a widely used distribution for queue modelling. The general law is that the chance of an event occurring at any instant of time is constant, and if the average number of successes in time s is λ , then the probability of x successes in time s is

$$P(x) = \frac{(\lambda \cdot s)^x e^{-\lambda s}}{x!}$$

Therefore if the time-unit is $s = 1$, then the arrival rate is λ . The mean and variance of the distribution is $=\lambda$. One assumption of the Poisson distribution is that there is no change in the probability of future arrivals. The name "memoryless" is derived from this implication. This means that the time between two arrivals is random and are following an exponential probability distribution, and the time between the arrival of one pair of packets is independent of the time between another pair of arrivals [Murd78]. Figure 4.1 shows the Poisson Distribution of the arrival rates.

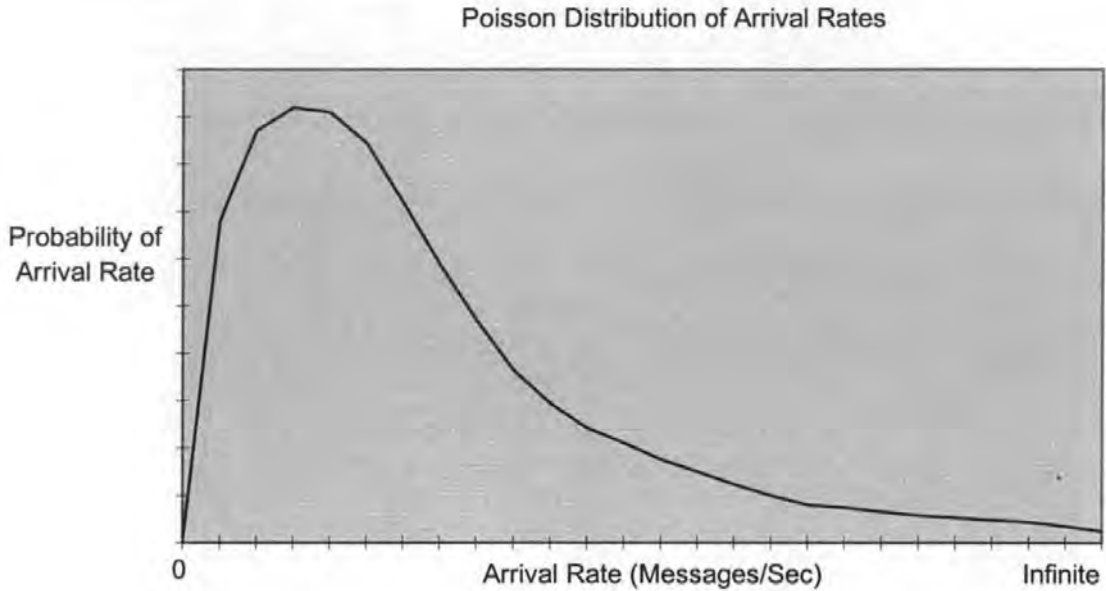


Figure 4.1: Poisson Distribution of Arrival Rates

4.5 Packet Trains

In [Jain86] Jain & Routhier published some results of network measurements and the study of the arrivals of network packets. They demonstrated how the *Poisson* models deviate from real-world measurements and proposed an alternative model. They described the various characteristics of their model, which is based on the measured data.

They offered the *packet train* model of packet arrivals to describe traffic on a token ring local area network at MIT. They defined a *packet train* as a burst of packets originating from the same source and heading to the same destination. If the spacing between two packets exceeds some inter-train gap, they are said to belong to separate trains. Jain describes the packet train model as a general model of which the Poisson is a special case. In his model, the inter-train time is a user parameter, dependent on the frequency with which applications use the network. The inter-car interval for a train is a system parameter

and depends on network hardware and software. They concluded that network packets tend to arrive in a sequence and called it "packet trains", as they looked like a multiple-wagon train. This model of network traffic is therefore based on measurement and experience.

In Poisson arrival models, the inter-train and inter-car interval parameters are merged to give a single parameter: mean inter-arrival time. Compound Poisson models use an exponential inter-train interval distribution and a zero inter-car interval.

The packet train model reflects the fact that much of network communication involves many packets spaced closely in time between the same two endpoints. Request/response applications will accordingly yield bi-directional packet trains.

The paper pointed out the following applications of the train model:

1. Protocol modelling.
2. Predicting the likelihood of the next packet destined for the same target.
3. Determining the number of buffers in routers/gateways/bridges.
4. Determining when to stop a temporary circuit (also known as a dynamic circuit).
5. Determine whether reservation switching is suitable.

[Chiu92]

4.6 Simulation Modelling

Simulations have been used for some time and have been applied to the modelling and analysis of many systems including economics, transportation, computers, factories and

many more real world applications. The simulation model for the implementation of the proprietary congestion control mechanism is used in PM.

Simulation is a dynamic tool that provides a very good insight into the detailed operation of the congestion mechanism. The modeller has the ability to define models of systems with simulation and put them into action and try them out before implementing them into real switches. A wide range of experiments can then be performed in a set and controlled environment.

There are many simulations based on the system being studied, but there are two main classes, which are of interest:

- continuous, and
- discrete.

These techniques provide the necessary methods to model most networking systems of interest. A continuous model is one whose processing states change in time based on time varying signals or variables. Discrete simulation relies on event conditions to change states. Both simulations provide dynamic means to construct and analyse queuing-based systems. They dynamically model the mathematical occurrences analysed in analytical techniques.

Many languages and programs are available for use in developing the computer executable version or a graphical implication of a model:

Simscript, Slam, Network 2.5, WITNESS and SIMUL8.

The choice of the language is not only based on the users' needs and preferences, since any of these will provide a useable modelling tool for implementing a simulation, but also the costs of the simulation packet and the time to build the model [Law91]. The simulation package chosen for this project is WITNESS developed initially by AT&T and now owned by the Lanner Group.

4.7 Network Management and Monitoring

In a shared network, much of the required information can be collected by an external or a remote monitor that simply observes the traffic on the network. This arrangement offloads much of the processing requirement from operational nodes to a dedicated system. In Table 4.1 different types of measurements are listed which give some idea of the kind of measurements that are of interest. These measurements can be used to answer a number of questions.

Name	Variables	Description
host communication matrix	source, destination	(number, %) of packets, data packets, data octets
group communication matrix	source, destination	as above, consolidated into address groups
packet-type histogram	packet type	(number, %) of (packets, original packets) by type
data-packet-size histogram	packet size	(number, %) of data packets by data octet length
Throughput-utilisation distribution	source	(total octets, data octets) transmitted
Packet inter-arrival time histogram	Inter-arrival time	time between consecutive carrier (network busy) signals
Communication-delay histogram	packet delay	time from original packet ready at source to receipt
collision-count histogram	number of collisions	number of packets by number of collisions
transmission-count histogram	number of transmissions	number of packets by transmission attempts

Table 4.1: Performance Measurement entities

Source: Amer et al (1983).

All networks require some level of network management in order to ensure reliable service. Monitoring the health of the network can help identify problems before they become detrimental to network users. It also can help predict trends in traffic patterns and volume.

As network management is a demanding task and should provide consistent and cross-vendor network management, the access to network management information is being standardised. The standardisation was focused on defining which information can be monitored and the parameters that can be controlled in a network. This provides a transparent way to manage and monitor objects by different vendors and their products. Standard Simple Network Management Protocol (SNMP) is supported by most vendors and is one major component today [Mull90].

4.8 Monitoring Objectives

The ultimate goal of monitoring the network is to be able to manage the network. To gain a proper perspective of what to monitor, it is essential to understand the requirements for network management. Network management involves the planning, installation, and operation of all the network components to meet the needs of an enterprise. Such needs can be translated into specific levels of performance (bandwidth, response time), reliability, availability, security and accounting.

A substantial effort has been put into characterising the basic network management functions as part of the definition of management information. The OSI standards divide the network management functions into the following five areas [OSI]:

- Fault Management;
- Configuration Management;
- Performance Management;
- Security Management; and
- Accounting Management.

For further reading and reference the network management functions are described in [Terp92] [RFC1709] [Salah94].

4.8.1 Different Types of Monitoring

There are basically a few ways of tracing packets. The first one is a Network Analyser, which is a machine and/or software, built for the purpose of network tracing and network monitoring. Another possibility is a Remote MONitor (RMON) and its Management Information Base (MIB) which is a standard set of capabilities and interfaces able to communicate through SNMP. A third method of monitoring is a MIB embedded into a managed device like bridge or router, which is usually an inexpensive way of monitoring, compared to RMON. This mechanism allows monitoring communications between a large number of systems without any additional physical intervention into the network.

4.8.2 Monitoring Agents

The applications and implementations of network monitoring agents will vary, and are very much dependent on the vendors. However there are functions that are common to nearly all such mechanisms. They include the following:

1. Filtering - using only selected information from the input stream.
2. Collecting - storing the filtered data either temporarily or permanently (or until such time as it is needed by a management application).
3. Reporting - providing stored data to a management application as required.

4.8.3 Simple Network Management Protocol

Implicit in the SNMP architectural model is a collection of network management stations and network elements. Network management stations execute management applications which monitor and control network elements. Network elements are devices such as hosts, gateways, terminal servers, and the like, which have management agents responsible for performing the network management functions requested by the network management stations. The SNMP is used to communicate management information between the network management stations and the agents in the network elements.

Goals of the Architecture:

1. The SNMP explicitly minimises the number and complexity of management functions realised by the management agent itself. This goal is attractive in at least four respects:
 - The development cost for management agent software necessary to support the protocol is accordingly reduced.

- The degree of management function that is remotely supported is accordingly increased, thereby admitting the fullest use of Internet resources in the management task.
 - The degree of management function that is remotely supported is accordingly increased, thereby imposing the fewest possible restrictions on the form and sophistication of management tools.
 - Simplified sets of management functions are easily understood and used by developers of network management tools.
2. A second goal of the protocol is that the functional paradigm for monitoring and control be sufficiently extensible to accommodate additional, possibly unanticipated aspects of network operation and management.
 3. A third goal is that the architecture be, as much as possible, independent of the architecture and mechanisms of particular hosts or particular gateways [Case88].

Network monitoring serves three primary purposes:

1. Constant observation of the 'health' of the network, network components, and external network connectivity. Network servers and workstations can be monitored in this way as well as routers and bridges. Operations staff can be provided with network monitoring stations that will display alerts immediately upon detecting a variety of problems or anomalies.

2. Collection of statistics on the performance of the network and patterns of traffic in order to identify needed enhancements or re-engineering. Using the same SNMP capabilities mentioned above, data on packet forwarding and total traffic volume could be collected and used to generate periodic reports on network utilisation.
3. More rapid problem resolution. When problems do occur, SNMP tools can help to pinpoint the source of the problems. Such problems include routing anomalies, domain name service (DNS) query failures, multiple network addresses, network utilisation problems or even attempts at breaking into network accessible host computers.

Since network management and monitoring is a technically demanding task and requires special equipment and software, it should be a centralised function in the initial design of network systems. [RFC1709]

4.8.4 Protocol Analyser / Monitors Used for the Project

Protocol Analysers have many of the functions of a network management console, but are restricted in their view of the network to one segment. The protocol analyser can be used if there are serious problems that make the use of the network management station impossible.

Another role for protocol analysers is to trace messages between two machines that are having trouble communicating. Decode applications display the sequence of messages and help to identify the problem. This type of function is particularly useful to the

development of new network software, or when new applications are being deployed on the network and there are problems getting them to work.

A final function of protocol analysers is to generate traffic. The protocol analyser can generate load in order to stress networking devices or to see what the effects of adding traffic to the network will be before changes are made.

Protocol analysers are implemented in hardware and software or just as pure software running on general-purpose computers. This has significantly reduced their cost and improved portability by allowing low-cost portable computers to be used, or by allowing the software to be installed pervasively so that any machine already attached to the network can be used for protocol analysis [Phaal95].

Some monitoring agents may provide certain of these functions at a minimal level. For example, a real-time data analyser like the W&G DA30 may trace information directly to the display screen and also store selected and filtered information to the hard disc, whereas the ETHERDUMP only can filter a certain packet type out of the arrival traffic.

4.8.5 Data Collection

The analysis in the following chapters requires a data collection mechanism that can capture time-stamped header information from packets as they traverse the LAN. Since the processor implementing such a mechanism must transfer information about each of the packets from the interface driver through the operation system into the collection application, the performance impact of the collecting process is a concern. Even for the relatively brief intervals the high volume of data which traffic collection generates, even

when limited to packet headers and no user data, requires that only essential header information is presented.

4.8.6 W&G Network Analyser

The Wandel & Goltermann DA30 is a multi-port dual LAN/WAN analyser which was kindly donated by Wandel & Goltermann Ltd in 1995 to the University of Plymouth. This enabled the author to use a state-of-the art equipment for the project.

The architecture of the analyser is basically divided in three sections: the network interfacing, protocol analysis and the user interface. Each of these has its own processor working on its task in parallel. This is its main advantage and gives it the power of simultaneous analysis.

The network interface processor is an INMOS transputer, which can handle a set-up and implementation of 256 hardware address filters, time stamping and communications with the other processors.

The protocol analysis processor is also a transputer, which analyses high-frame-rate data streams at network speed. It executes multiple parallel processes, one for each layer of the OSI protocol stack and one for central control and interaction with the other processors. This enables the DA-30 to decode a seven -layer protocol stack in real time.

The third processor is a 485SLC, which manages the user interface during set-up and definition procedures. It runs on a disc operations system (DOS) and enables the use of the stored data in standard applications. [Oper93]

4.8.7 Sniffer on a PC

The Ethernet protocol works by sending packet information to all the hosts on the same circuit. The packet header contains the proper address of the destination machine. Only the machine with the matching address is supposed to accept the packet. A machine that can accept all packets, no matter what the packet header says, is said to be in promiscuous mode. This allows a so-called sniffer program to set a PC or a UNIX workstation into such a mode and trace all packets arriving on the Ethernet.

One such sniffer program is ETHDUMP. It is a companion utility of ETHLOAD, which can be found on the Internet and is a complete freeware in its version 1.x. The software puts the controller in promiscuous mode and dumps all received frames into a file, which can be viewed with an ASCII viewer. The sniffer was used as well as the W&G network analyser, for traffic profiling in this thesis.

The format is as follows:

time_from_start	size_in_bytes	mac_destination	mac_source	ethernet_type	ethernet_data
5	2	6	6	2	64-1500

time_from_start = packet on the Ethernet from the start of the dumping

size_in_bytes = the size of the packet

mac_destination = mac destination address

mac_source = mac source address

ethernet_type = the protocol which is carried in this particular packet

ethernet_data = the raw data inside the Ethernet packets

An example of a traced packet:

; Packet trace started at Tue Feb 06 15:47:41 1996

0000000001 0072 0000A903A5D0 08002B34FA32 8137 FF-FF-00-39-00-11-00-00-0C-
15-00-00-00-00-00-01-04-51-00-00-E2-50-08-00-2B-34-FA-32-40-03-22-22-8D-49-09-
00-3E-00-12-53-59-53-3A-4D-41-49-4C-2F-46-32-30-44-30-30-30-31-2F

In this example can be seen that the

packet size is	114 bytes
destination address is	0000A903A5D0
source address is	08002B34FA32
type of the packet is	8137

Figure 4.2: The type 8137 assigned to the IPX protocol

4.9 Estimating Network Traffic

The planning and the installation of a new network require knowledge about the network demands. One possibility, which is often used for small networks, is "trial and error". The problem with this method is that at the time a high level of utilisation is noticed, the network has to be subdivided in different segments through the use of a local bridges or routers. This can disturb the network users and if the utilisation level is not recognised soon enough this will slow down the whole operational process in a company. To provide a better level of performance to local area network users, it is important to understand how to estimate network traffic. Therefore, the results of a traffic estimation process can be used and determine if any changes should be made to the LAN.

4.10 The Network Traffic Estimation Process

Assuming there is no access to monitoring equipment to analyse an existing local area network or the network is in a planning stage, the development can be reasonable by estimating the traffic by considering the functions each network user performs. In [Held92] a methodology was developed to estimate network traffic on a local network. A traffic estimation worksheet is used to predict the average and peak traffic that could arise on a typical Ethernet and Token-Ring network.

To facilitate the traffic estimation process, a number of network users can be placed together into a 'User Class' category. To estimate the traffic for future users from the same class, the results can be multiplied by the number of workstations grouped into the specific 'User Class' to obtain an estimation of network traffic for a similar group of network users [Held93]. This process can be repeated for different user classes and added together. A similar approach of the traffic estimation process (TEP) is used by many wide area network and service providers to estimate capacity requirements for permanent virtual circuits (PVCs) when interconnecting LANs via Frame Relay.

The limitation of this approach however is that it cannot take account of the WAN environment. This approach is used not only for the CIR but also for oversubscription. The claims of [Held93] in regard to user classes could not be found in the literature in relation to Frame Relay or the Internet. Therefore, new traffic traces for Frame Relay and Internet related traffic had to be collected and analysed. The results can be seen in Chapter 5. These results are then used to develop the initial TEP into a new methodology for the use on Frame Relay.

4.10.1 Performance-Measurement Matrix

Table 4.1 on page 74 has shown the variables and descriptions of the interesting performance measurement entities. These areas are of interest to the network manager and network planner. Other questions are concerned with response time and throughput by user-class and determining how much growth the network can absorb before certain performance thresholds are crossed. Additional network environment variables that influence network planners are:

- Are any time-critical protocols in use (e.g. SNA)?
- What type of cabling is used, is there a mixture of cable types?
- Routing schemes and topology of interconnection?
- Are there any unusual reliability requirements or security requirements?
- Interface characteristics and requirements?
- Are distributed file servers in use, (difficult to predict future traffic)?
- Are hypertext applications in use, e.g. WWW?
- How many word processing applications and spreadsheets are in?
- Is electronic mail implemented, and is it used just locally or connected to the Internet?
- What are the reliability considerations?
- What a network management platform is chosen?
- What computer types are used?
- Is an interconnection to existent MAN or WAN necessary?
- Are very large databases in use?

A network measurement centre performs collection, processing, and storing of measurement data. The general measurement task is to register and evaluate statistics for building up appropriate traffic models for network modelling and analysis. The problem comprises the following areas to be solved before implementation.

4.10.2 Defining of Measurement Tasks and Requirements

The reason for measurement dominates the task itself:

- Evaluation of traffic flows and user traffic characteristics;
- Evaluation of network characteristics and performance detection of bottlenecks; determining the basis for network optimisation;
- Determining the basis for traffic and congestion control strategies;
- Support for network modelling;
- Network administration;
- Load distribution and capacity planning;
- Billing calculations; and
- Understanding of network behaviour.

Network-wide measurements usually focus on the reasons mentioned above and require consideration about:

- User-oriented versus network internal measurements;
- Network-wide versus spatial-restricted measurements;
- Simultaneous multiple nodes versus sequential single-node measurements;
- Long-time or short-time measurements; and

- Continuous versus sample-oriented monitoring.

It has to be understood that network-wide measurements can cause a problem because of:

- Distributed and shared measurement facilities;
- The transmission of the gathered information creates additional network load; and
- Measurement administration and synchronisation problems.

4.10.3 Analysis of Structure, Applications and Protocols

A set of parameters, with regard to the objects of the network to be measured, is necessary:

- User oriented or host oriented traffic;
- Server-to-server traffic;
- Terminal/PC to Server traffic which is based on user behaviour; and
- Time measurements of delay, response, connect, disconnect times.

4.11 Protocol Modelling of Proprietary Mechanism

As indicated in the previous chapter, the mapping of the Frame Relay packets into the Fixedpackets creates an overhead. This increases the volume of data on the PVC links by adding data overhead. The additional data added by this protocol include the following:

- Control bits for framing, delimiting the protocol packets;
- Control bits for sub-framing, to delimit the data from each switch input;
- Addressing to indicate the link from which data originates; and

- Error detection, to allow re-transmission of corrupted data.

In discussing the modelling of the access network the focus is placed on the Fixedpackets and their overheads in the PM Frame Relay network. In order to evaluate the increase in LAN traffic due to the Fixedpacket overhead some simple equations are developed to estimate the increase, based on the rules of the protocol.

4.11.1 Representation of the Fixedpacket Overhead

In order to account for the increase in Fixedpacket network traffic due to the overhead some equations have been derived which can be applied to the traffic figures to adjust the message-length and arrival-rate probability distributions accordingly.

The total additional protocol characters added to the original user traffic profile are based on the actual values of length and arrival rate over the range of values they take. Some simple integer arithmetic is performed to calculate how many data overhead bytes would be produced for each possible arriving message length for all the users connected to the switch. The basic model is then adjusted to account for the increase in traffic.

4.11.2 Calculation of Fixedpacket Overhead

In [Gav93] a method for the calculation of packet overhead is shown. The method is used in the simulation (Chapter 7) for the calculation of the overheads. The definition is as follows:

Define a function to account for additional bytes on Fixedpackets, relating it to the result of the modulo (mod) function.

K is the variable that can only take on the value 0 or 1.

$K = 0$, if the MOD function has a non-zero remainder.

$K = 1$, if the MOD function has a zero remainder.

In order to write an expression defining the total number of characters leaving the switch output, the basic input arrival rate must be known and the number of additional bytes created by the protocol must be calculated. To simplify the analysis, consider only a single input with arrival rate λ and frame length σ data bits. Where $\sigma = 1/\mu$ and μ represents the frame service rate in frame/bit.

The total overhead bytes per second is calculated as follows:

1. Define N to be the total number of bytes, on average, offered by the switch input for insertion into a protocol envelope per second.

$N = \text{total number of data bytes} + \text{number of bytes due Fixedpacket headers},$

$\text{total data bytes per second} = \text{arrival rate} * \text{frame length} = \lambda * \sigma$

$$N = \sigma \lambda + [(\sigma \text{ DIV } P) + K] h$$

Where h is the Fixedpacket header, and P the Fixedpacket length.

The overhead bytes due to Fixedpacket header are caused by each arrival of frames being assigned to one or more Fixedpackets. Therefore, a minimum of one overhead per arriving frame is required. The number of Fixedpackets per message is defined by

the frame length divided (using the integer dividing function DIV) by the frame size plus one extra frame to carry the remainder of bytes not exactly filling a frame (hence the K-term).

The overhead due to the Fixedpacket header = $[(\sigma \text{ DIV } P) + K] h$

2. Define the total number of protocol frames per second to be N.

N = total number of bytes including Fixedpacket headers divided by the frame size.

The K function means that the partially filled frame is accounted for.

$$N = [\sigma \text{ DIV } (F-H)] + K$$

3. Let Δ be total character overhead due to the protocol,

$$\Delta = \Lambda [(\sigma \text{ DIV } P) + K] h + N H$$

The frame formation time at the destination depends very much on the traffic delay of the Fixedpackets in the Frame Relay network. The overhead is based on the bytes per frame formation time basis and is then multiplied by the number of frames per second to give protocol overhead per second.

4.12 Analysing TCP/IP

In order to implement an equivalent to TCP's congestion control, it is necessary to understand the effects of packet loss on the overall performance of a TCP connection.

Background packet loss affects the capacity C of a TCP connection by the following formula [Flo91] [OM96]:

$$C = 1.22 * \Delta / RTT * \sqrt{loss} \quad [1]$$

where Δ is the packet size being used on the connection; RTT is the round trip time; and the loss is the loss rate being experienced by the connection. The following gives a simple derivation for this formula.

Consider a TCP connection with a particular roundtrip time and packet size. Further consider a model where the network drops a packet from that connection when the connection's congestion size increases to W packets. Assume that the congestion window is then cut in half, then is increased by one packet per roundtrip time until it reaches W packets again, at which point the network again drops a packet and the steady-state model continues as before. For such a model, to derive a formula for the steady state drop rate as a function of the roundtrip time, packet size, and average capacity the following is assumed.

The TCP connection has Δ bytes/packet, and a roundtrip time of RTT seconds. Assume that, when a packet is dropped, the TCP connection had a window of W packets, and was sending at an average rate (over that roundtrip time) of

$$S = W * \Delta / RTT \quad [\text{bytes/second}].$$

After the packet is dropped, it takes roughly $W/2$ roundtrip times for the TCP sender's congestion window to build up again until it reaches its old value (and, in steady state, the TCP connection receives another packet drop). Thus, in steady state the TCP connection receives an average capacity of $0.75 * S$ bytes/second, (because the sending rate varies smoothly between $S/2$ and S bytes/second). Restated for capacity = $0.75 S$, the TCP connection receives an average capacity of

$$C = 0.75 * W * \Delta / RTT \quad [\text{bytes/second}] \quad [2]$$

To derive equation 1, note that Loss, the loss rate for that TCP connection, is

$$\text{Loss} = 1 / (W/2 + (W/2 + 1) + \dots + W)$$

$$\text{Loss} \sim 1 / ((3/8) W^2)$$

This gives

$$W \sim \sqrt{8 / (3 \text{ Loss})} \quad [3]$$

Substituting [3] for W in [2] gives Equation [1].

Equation [1] has been verified by simulations in [Floyd97] for loss rates up to 5 per cent. As the loss rate increases, Equation [1] begins to overestimate the capacity received by a TCP connection.

Equation [1] does not apply at all for loss rates of 15 per cent or more, for the following reasons. It can be seen from this model that this formula only applies for TCP connections that achieve a value W of at least four packets when a packet is dropped. For TCP

connections that send only a small number of packets per roundtrip time, losses are dominated by the delays imposed by retransmit timeouts, and not by the dynamics of this steady-state model. Thus, this model clearly does not apply for $W < 4$.

4.13 Markov Model

Two types of traffic models are used in the simulation. The first type is a source with packet trains generated by a two state Markov Modulated Poisson Process (MMPP). The second model uses the TCP/IP protocol to study the interaction of end-to end protocols on the backbone.

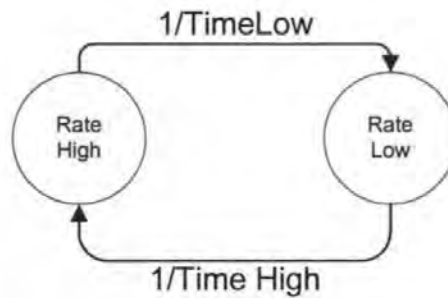


Figure 4.3: Parameters for a two-state MMPP source

The two-state MMPP source can be characterised by four metrics:

- $\sigma_1 = 1/\text{Time Low}$
- $\sigma_2 = 1/\text{Time High}$
- $\lambda_1 = \text{Rate Low}$
- $\lambda_2 = \text{Rate High}$

The MMPP parameters are given for modelling the packet trains by end-stations. A two-state MMPP process is a double stochastic process where the rate of the generation of packets is determined by the state of a continuous-time Markov chain. The two state model has a Rate High to model bursts lasting Time High seconds, and Rate Low to model a low intensity period for Time Low seconds. The transition rates between the two states are the inverse of the Time High and Time Low parameters, as can be seen in Figure 4.3. In both states the state occupancy times are negative exponentially distributed, with the inverse of λ_1 and λ_2 as the corresponding mean inter-arrival time. Now the mean transmission rate for this source is:

$$\lambda = \frac{\lambda_1 * \sigma_2 + \lambda_2 * \sigma_1}{\sigma_1 + \sigma_2} \quad [\text{bits/s}]$$

The MMPP parameters will be represented in the next section by

MMPP = TimeLow / TimeHigh / RateLow / RateHigh. The times are in seconds and the rates in kbit/s. The simulations have exactly the same source throughput versus time characteristics to compare the results with each other.

4.14 Granularity

An important aspect of this research is granularity, or the extent of the communicating entities. Potential granularities include traffic by application, end user, host, network number, backbone node, domain, department, company, and many more. These granularities do not necessarily have an inherent order, as a single user or application might use several hosts or even several network numbers. One example of flow granularity of interest derives from [Ray93], which identifies different traffic profiles, on X.25 by business sectors. He found for example that with the travel industry the traffic is

defined as “a small series of short burst of bursts” by the user and then “a repeat of large bursts of data from the remote host to display some holiday options”. However the traffic profile of the motor industry was defined by “long repetitive bursts of data from user to host to request car part orders”.

4.15 Flow Definition

The four aspects of the flow model are

- Directionality;
- One sided or two sided;
- Granularity; and
- Functional layer.

The directionality was chosen to investigate unidirectional flows. For the second point flows in single flows were examined. The granularities were PVCs of companies, departments, specific WWW applications, network numbers. The last aspect was laid in the network layer and in some aspects the transport layer. The information used was the start and endpoints of a session.

Chapter 5: Analysis Results of Obtained Data

5.1 Network Profiling

One limitation, which exists in profiling, is the availability of detailed statistics over a long period of time. This is usually associated with memory restrictions and performance problems, as gathering statistics is a processor-intensive procedure.

The statistical gathering for this research was divided into two stages. In the first stage profiling took place at the University of Plymouth and at AT&T in Redditch. In both locations several LAN segments were chosen where the technical environment would be considered similar. The LAN traffic of both locations was then used to create traffic profiles.

In the second stage, the Frame Relay Network of AT&T was used for traffic profiling. This network is spread over different locations of the UK and Europe. Capacity decisions and the choosing of the CIR for the Frame Relay network depends on knowledge of the network behaviour. The profiles of the Frame Relay are classified in this chapter and have then been used by the Traffic Estimation Process (TEP).

5.1.1 Utilisation at University of Plymouth

At the beginning of the observation the following utilisation levels at the University shown in Figure 5.1, Figure 5.2, Figure 5.3 and Figure 5.4 were observed (see Appendix C, Sections C.2 and C.3, for numerical details and further examples of night and day utilisation levels). Usually the utilisation started to rise at around 9 a.m. At that time the average utilisation was between 25 per cent and 30 per cent. But for shorter intervals the utilisation levels were much higher and could rise up to 70 per cent and then further to 80 per cent in the busiest second. The graphs indicate the main usage of the network between

9 a.m. when the library opens, till around 10 p.m. when the computer pools are closed. However, it can be seen that during the holiday period, the utilisation levels drop earlier. This indicates that the library and the computer pools close at 5 p.m. It can also be noted that the utilisation during the day is lower than during term times.

The average utilisation level by night is around 2-5 per cent. This indicates low activity on the network. The only traffic which can be seen in Figure 5.2, is between two servers that are polling different computers and backup data.

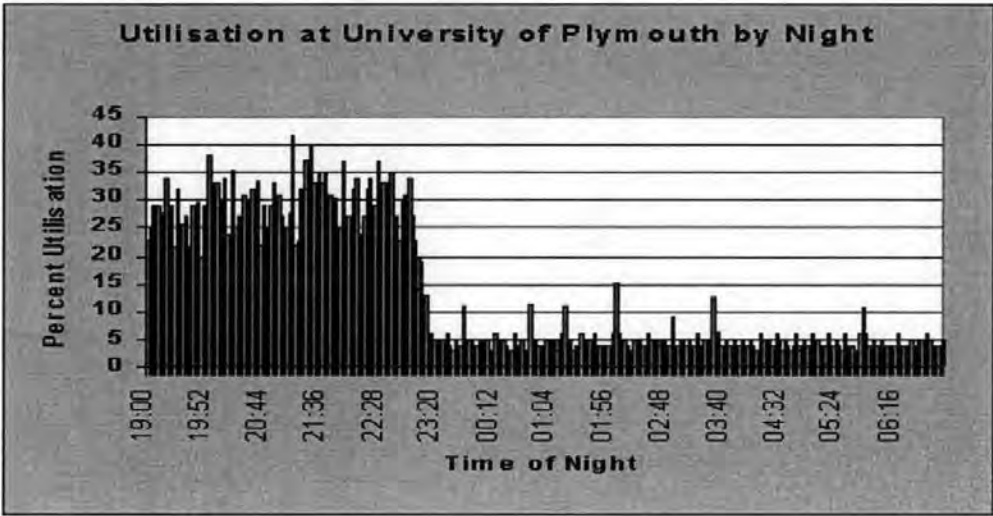


Figure 5.1: Utilisation at University of Plymouth by Night

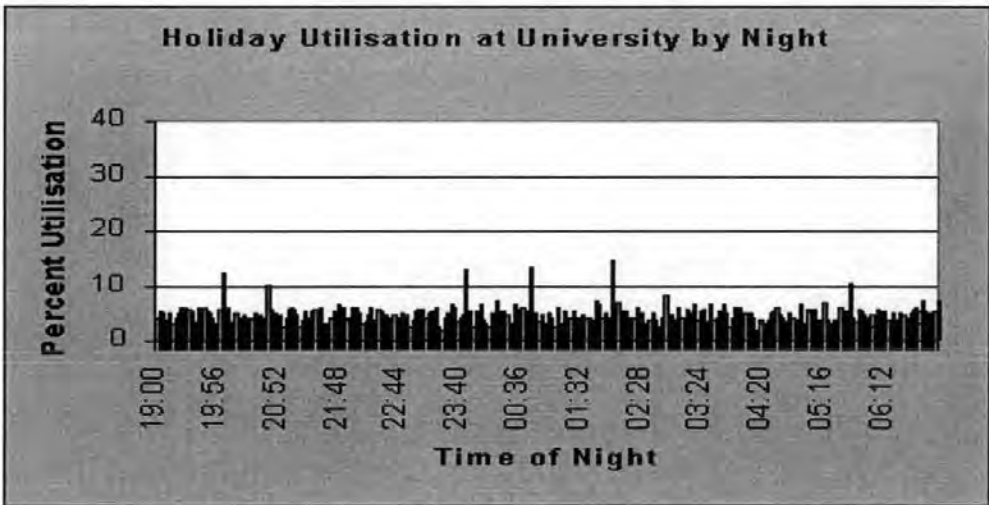


Figure 5.2: Utilisation at University of Plymouth by Night during Holidays

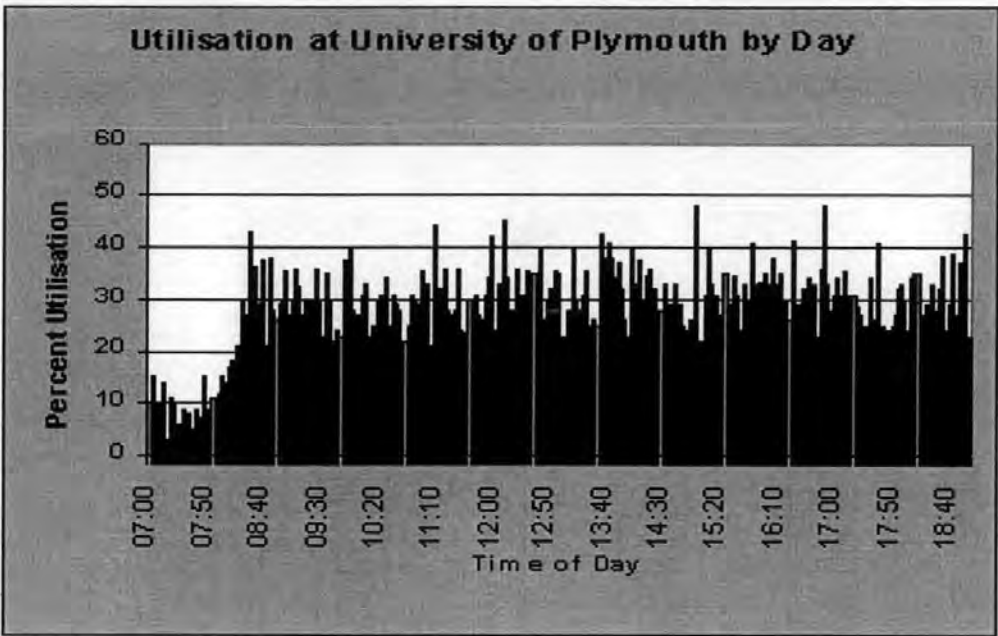


Figure 5.3: Utilisation at University of Plymouth by Day

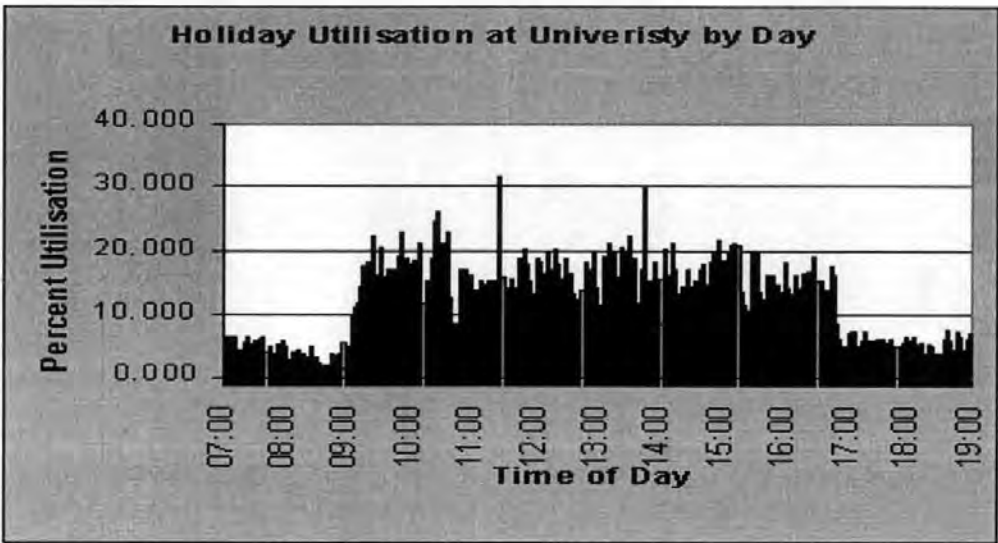


Figure 5.4: Utilisation at University of Plymouth by Day during Holidays

5.1.2 Utilisation at AT&T

The utilisation levels of the segments at AT&T are shown in Figure 5.5 and Figure 5.6 (see Appendix C, Sections C.2 and C.3 for numerical details on further examples). The traffic started to rise at around 8.30 a.m. indicating the beginning of the “working day” for most of the employees in this department. The average utilisation over the day was

between 15 per cent and 22 per cent, but for shorter intervals the utilisation levels could rise from 70 per cent up to 75 per cent. Figure 5.5 and Figure 5.6 indicate the main usage of the network between 8.30 a.m. and 5 p.m., when most of the people leave work. By night the average utilisation level was much lower. However, a few major peaks by night show considerably high utilisation between 15 per cent and 20 per cent. The reasons for these peaks by night are batch files are written for mirroring and backups. These peaks occur at exactly the same time every night.

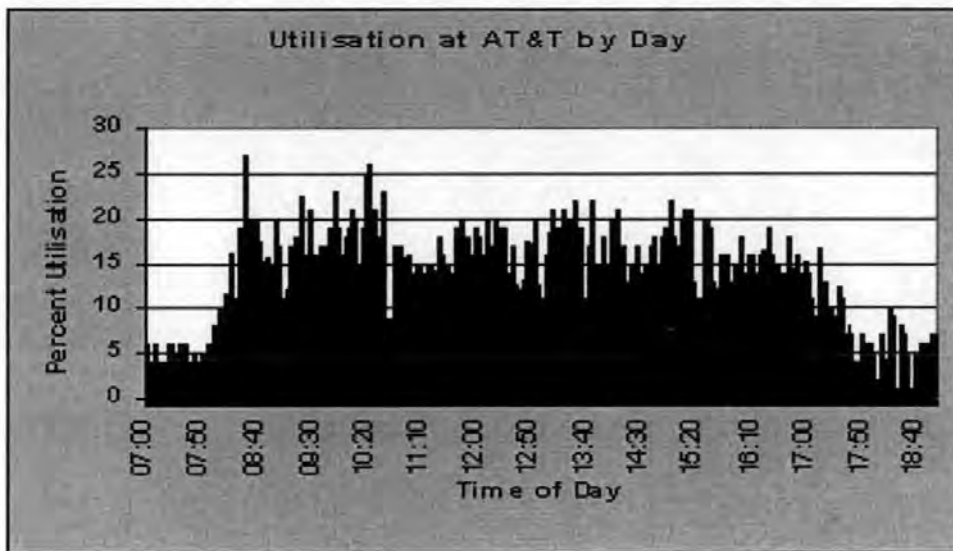


Figure 5.5: Utilisation at AT&T by Day

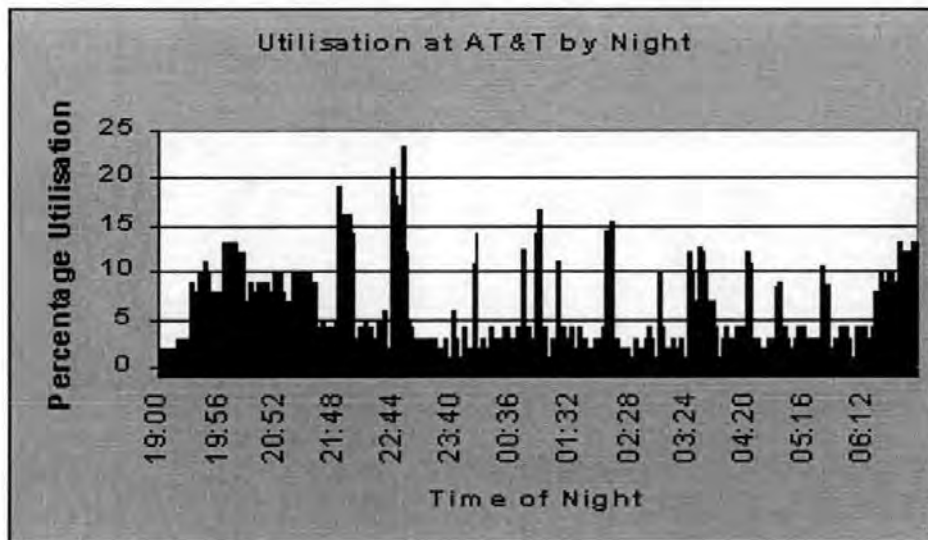


Figure 5.6: Utilisation at AT&T by Night

5.1.3 Packet Size Distribution at the University of Plymouth

Aaron Kershenbaum stated in [Kersh93] that "usually messages are packetised, dependent on the network protocol requirements, and each packet is transmitted separately and independently. Therefore the delay analysis should be done in terms of the lengths of the packets, not the messages". The project takes this aspect into consideration and some of the observations were made on the packet size distribution as well as on message length distribution.

[Kersh93] further stated that "packetisation not only affects the average message length, it also affects the message length distribution". This was found to be true in cases where an overhead to the packets was added.

The measurements obtained in the research show a correlation between the packet length distribution and the protocols used on the network. A further aspect was noted. Applications on these protocols leave a recognisable "fingerprint". Every application that is used over the network can therefore be recognised under certain conditions.

The research undertaken involved looking at 500 million packets at different times of the day during a period of several months.

The knowledge of packet length distribution is very important for modelling and simulating the Ethernet and interconnection of LANs over WANs. With detailed knowledge it is easier to predict the capacity and delay characteristics of the network.

The packet size distributions found at the University of Plymouth have shown an interesting aspect. Two major protocols were in use. The Internet Protocol (IP) and the IPX protocol also known as Ping Protocol (PP).

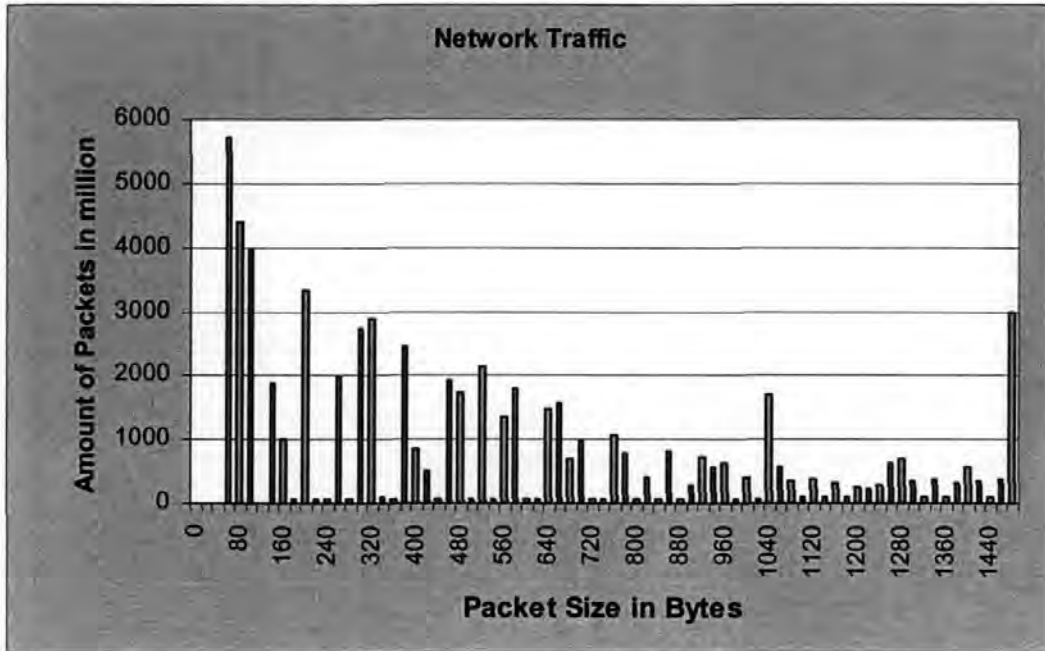


Figure 5.7: Packet Size Distribution at the University of Plymouth - IP

The distribution of packet size used by the applications of the IP protocol is shown in Figure 5.7. The majority of packets are 64 bytes and 1500 bytes long. These short packets are used to send control messages between the source and destination computer. The long packets are a sign of larger file transfers. However, it can be noted that there is a large number of other packet sizes represented. The reason for this could be due to the use of different applications like World Wide Web (WWW), File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP) on top of TCP/IP. The information accessed with these applications vary from very small messages (e.g. email messages) to large messages (e.g. graphics accessed via WWW). During the access the applications have to use the received data and acknowledge different window sizes for the source.

The IPX protocol however, shows a bimodal distribution in Figure 5.8, the major peaks are at 64 bytes and 500 bytes with hardly any other packet sizes. The explanation for this is that the protocol is used for client server operation to download Microsoft Windows applications and backup operations. The result is that they are cut into lots of considerably large packets when downloaded from the server and sent to the clients.

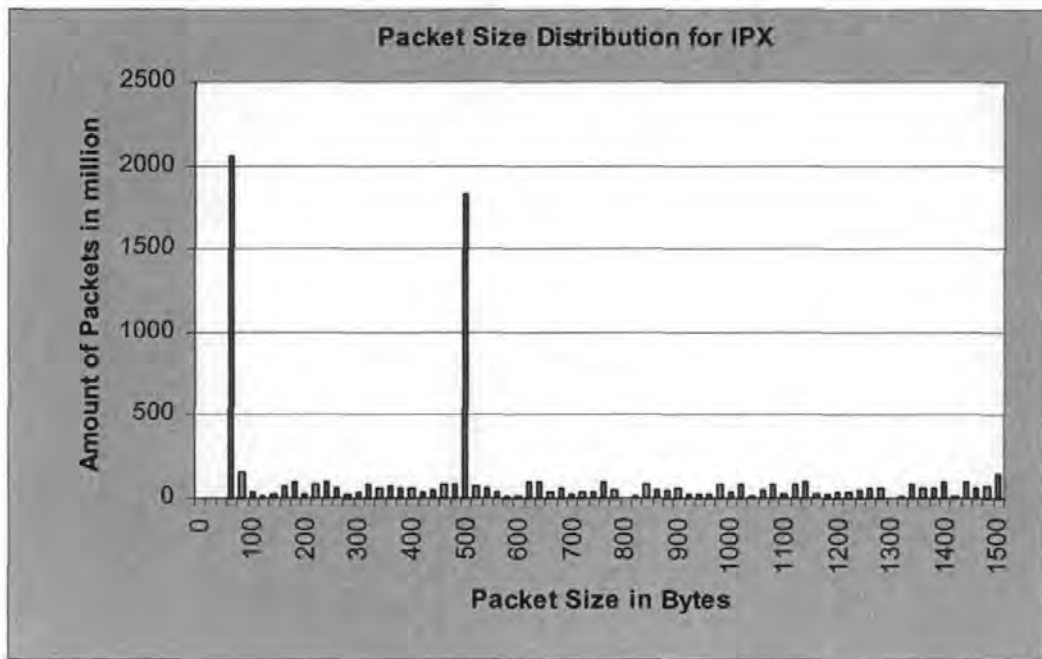


Figure 5.8: Packet Size Distribution at University of Plymouth - IPX

As this is an Ethernet network, which allows a maximum packet size of 1500 bytes of user data, these packets can be regarded as relatively small. This could be a function of the fundamental nature of the IPX protocol or, could be set-up by the network manager in that system.

5.1.4 Packet Size Distribution at AT&T

As shown in Figure 5.9 the shape of the distribution is similar to the bimodal distribution of the IPX protocol. The major difference, however, is that the peaks at AT&T's network

were at 64 bytes and 1500 bytes. The monitored network uses a client server environment with LAN Manager from Microsoft. The use of the LAN Manager protocol is very similar to the use of IPX.

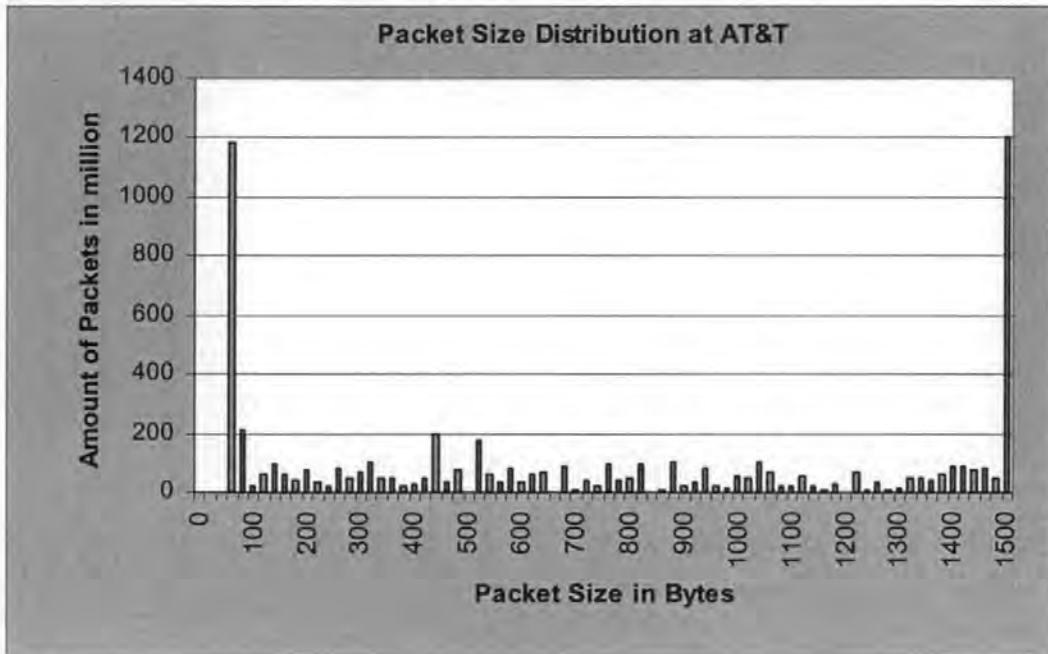


Figure 5.9: Packet Size Distribution at AT&T

5.1.5 Conclusion

Analysis by Grout [Grout89] has shown that the message length of one AT&T network closely follows a negative exponential distribution. In comparison to this is the observation of the packet size distribution found as one result of this project. The packet size distribution varied very much on the type of protocol used for different purposes, not the network itself. The multi-purpose TCP/IP protocol has shown a negative exponential distribution, whereby the LAN Manager and the IPX protocols were very similar and have shown bimodal distributions. The only difference between LAN Manager and IPX is in the second peak, which can be a reflection of the protocols themselves or the set-up by the network managers. However, all three protocols can be used on the same PVC at the same

time. The results will therefore be used in later chapters for the methodology and simulations.

5.2 Daily Traffic of Frame Relay Network

The utilisation has been taken over five working days in a week from 6 a.m. to 8 p.m. Figure 5.10 shows the overall utilisation of the Frame Relay network in Europe. It can be seen that these periods are showing a relative constant traffic profile and are very similar to the profiles found in networking and Tele-traffic literature. The busiest hours are between 9 a.m. and 4 p.m. Before and after these hours, the overall traffic decreases rapidly and is not very important for the planning process. Including these unimportant hours into the planning could even give a false reading of the network and lead to a misunderstanding of traffic loads.

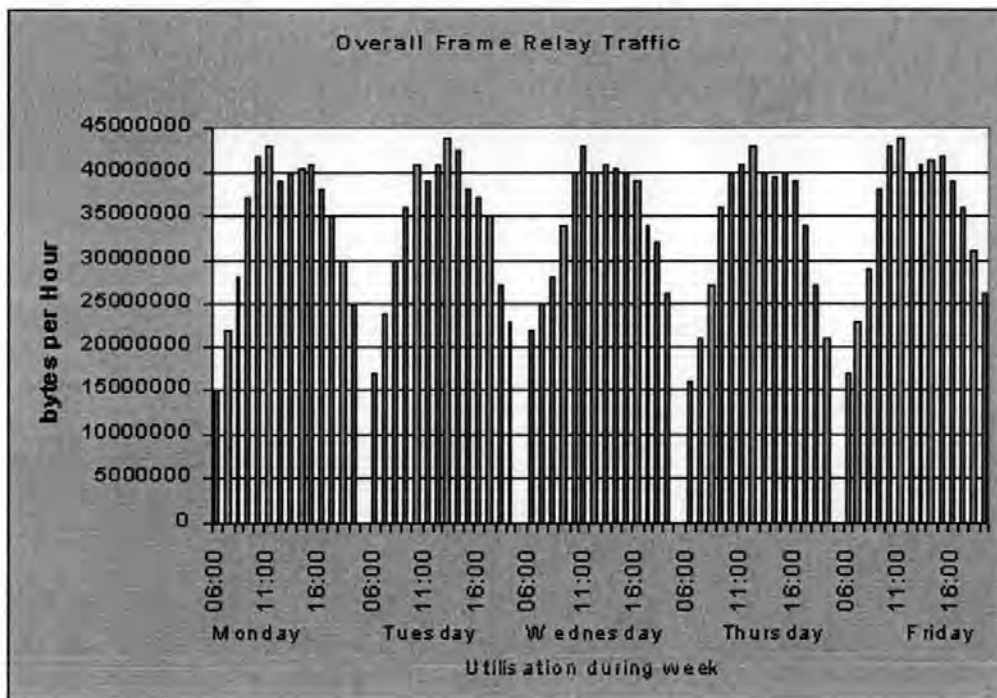


Figure 5.10: Overall Frame Relay Traffic

5.2.1 Individual PVC Profiling

It is important to break up traffic into components. The network provider does not know what exactly is sent over his routes, but knowing how individual PVCs are behaving is crucial.

Different types of traffic are emerging and for network providers it is important to characterise individual PVCs, regardless of transport protocol or application. Having seen the daily overall traffic of the whole network, it is important to see profiles of similar PVCs. The similarity from the perspective of the network provider is only seen in the booked CIR. Only PVCs with the same CIR should be compared directly.

Figure 5.11 and Figure 5.12 show examples of two different PVC dynamic utilisation. The PVCs show hourly figures (from 6 a.m. to 7 p.m.) for 5 consecutive days. It can be seen that the “busy” hours are usually between 9 a.m. to 4 p.m. There are a few peaks which are much greater than the busy period, but these peaks are not crucial as they change at every PVC and would comply with the theory of statistical multiplexing. There are no special concerns regarding peaks occurring from time to time, as there is only an interest in the trended traffic.

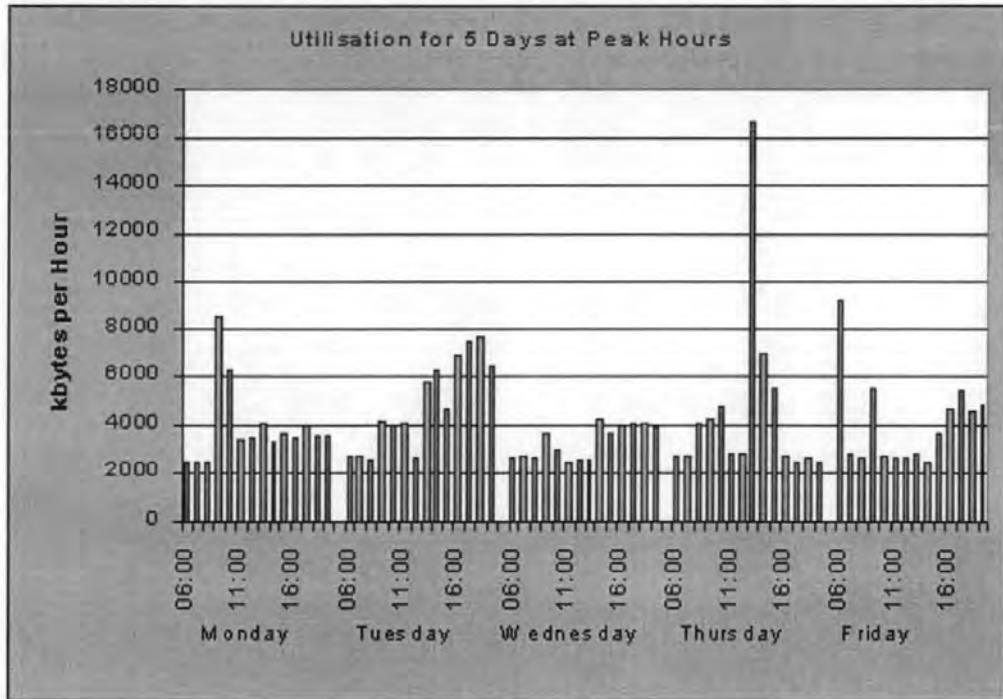


Figure 5.11: Utilisation for 5 Days at Peak Hours

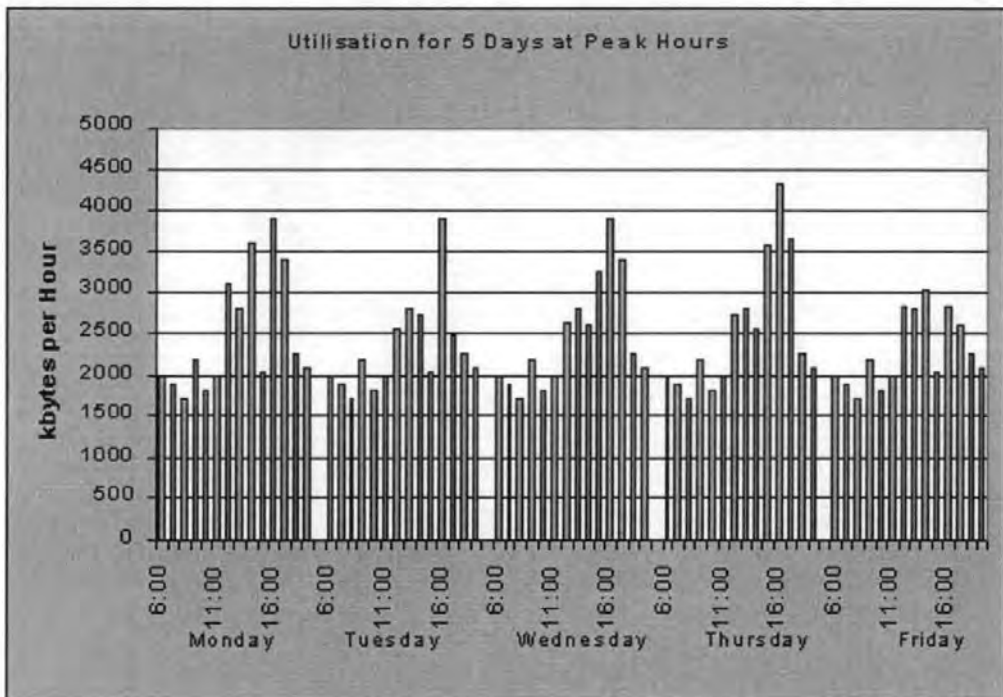


Figure 5.12: Utilisation for 5 Days at Peak Hours

5.2.2 Distribution

It can be seen in Figure 5.13 how the dynamic utilisation is distributed. To show it graphically the dynamic utilisation of all PVCs with a CIR of 256 kbit/s has been selected and they have been sorted in descending order. The distribution of PVCs on this shows that 20 per cent of the PVCs create 80 per cent of traffic above the CIR. This not only applies to 256 kbit/s PVCs but also for every other PVC size. This means that the distribution is independent of the actual CIR, but more related to the fact that there is a problem in the methodology to chose CIR values. The capacity allocation for the whole network based on actual dynamic utilisation is less than that required by the existing rule. Naturally, preferred routes have to be adjusted to reflect the required individual PVC capacity allocation.

The research has shown that the CIR levels are generally set much higher than the busy-hour actual utilisation for a great number of PVCs. While a few PVCs are using more capacity than allocated, most PVCs do not use the set CIR level. Using existing utilisation values for each PVC to design the network forecast trunk utilisation in line with recent actuals would occur. The network would then be more balanced, enabling the trunk to be loaded by low utilised and high utilised PVCs. At the moment, certain trunks are running at nearly 100 per cent dynamic utilisation causing packet losses and re-transmitting of frames and messages due to wrong overbooking. The model and the static load on a trunk appear to be correct. However, the use of the general overbooking rules does not take into account the imprecise setting of overbooking.

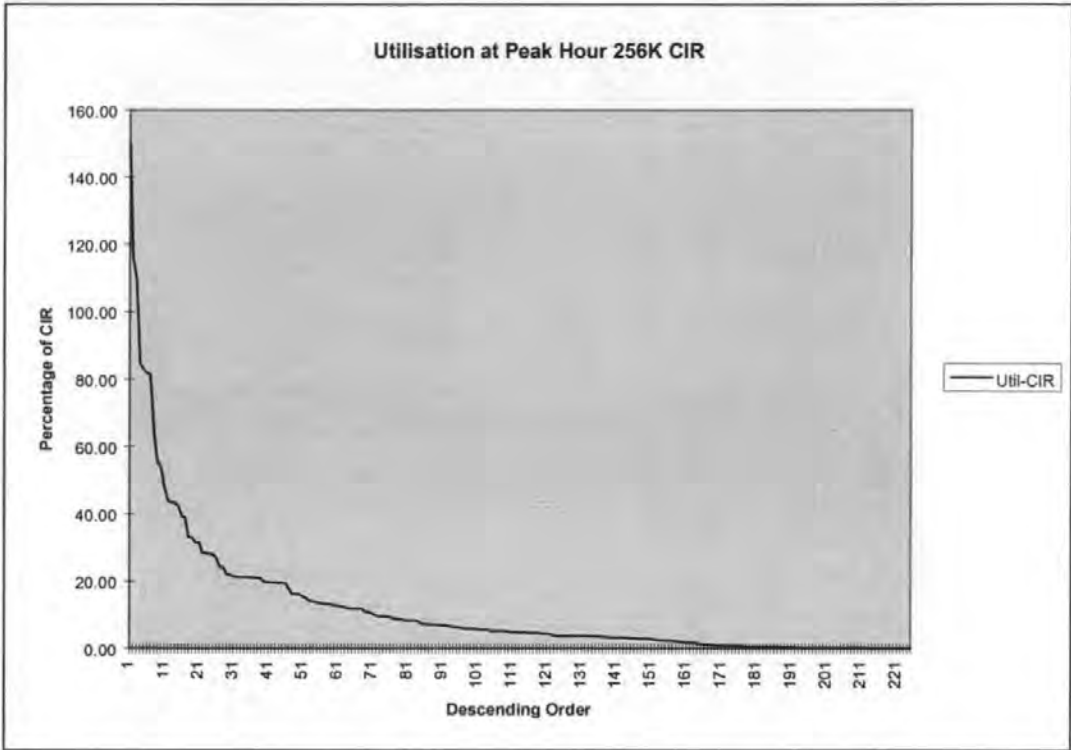


Figure 5.13: Utilisation at Peak Hour 256 kbit/s CIR

5.2.3 Variability

It is assumed that by using real dynamic utilisation, the model will be very accurate. However, if the dynamic utilisation readings put into the model are changing too often and furthermore are too high or too low, the model will not represent the network correctly. It is therefore important to have a certain continuity of dynamic utilisation levels over the measured weeks. As filling the trunks completely is not attempted, the utilisation of each trunk in the network at peak hours can be measured. This is carried out to see how close to the predicted utilisation they are and adjust accordingly the dynamic utilisation of each PVC being fed into the model.

PVCs, which change their utilisation at peak hours dramatically in an upwards or downwards direction from one month to the next, are very difficult to categorise. By monitoring these changes, actions can be taken very soon after they occur. In Figure 5.14

is shown the change in dynamic utilisation over the period of a month. The results in the graph are for the month of May 1997.

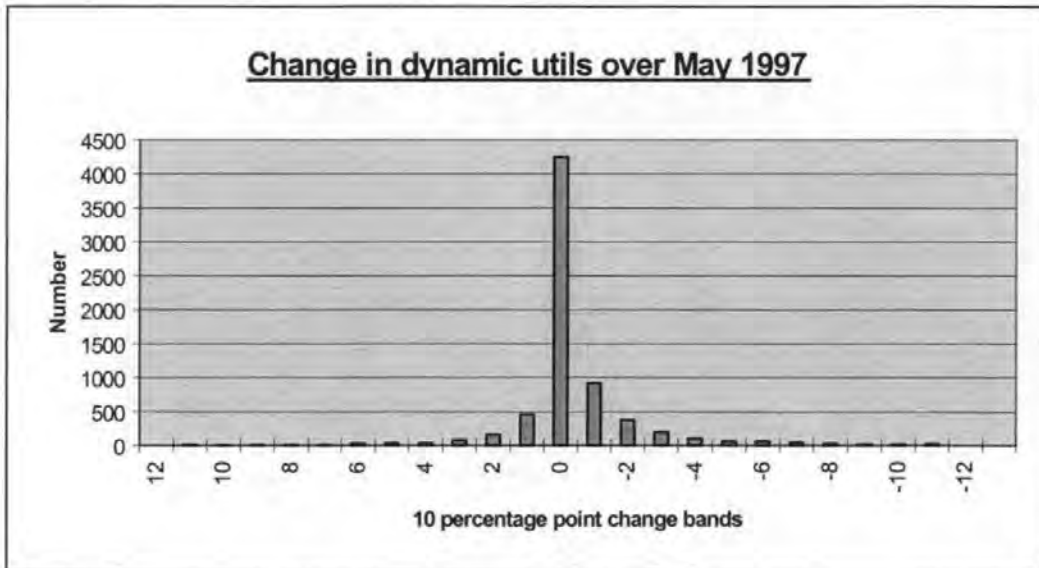


Figure 5.14: Change in Dynamic Utilisation over May 1997

The graph shows how stable the average dynamic %UTIL is in practice. Nearly 90 per cent of all PVCs change by less than 10 per cent, therefore giving a good level of confidence and hence predictability.

The investigation has discovered that PVCs' peak utilisation tend to grow with a steady linear rate, interrupted by a sudden jump. This "step" upwards can be explained by the introduction of new software or the use of new servers and services by the users. In the case of new Internet connections, some PVCs jump by 100 per cent or more over one week, and stay at this level. This has to be recognised, as these PVCs need special treatment and should be watched more closely. Averaging the busy period in a rolling window and not recognising this jump could result in a big error and in capacity limitations. Major contributors to these traffic jumps are the Intranets and the use of

WWW browsers. It is difficult to predict these steps, but from experience on some of these large networks, it can be seen that the appearance of WWW servers creates these sudden increases of traffic. Providing the step changes occur to a small number of PVCs in a month, their extra capacity needs can be met from the contingency provided by the allocation to the rest of the PVCs on a trunk. Experience indicates that mixing traffic from many users on a trunk reflects these individual step changes into a general overall expansion of use. Monitoring and revising utilisation weekly is expected to adequately accommodate these movements.

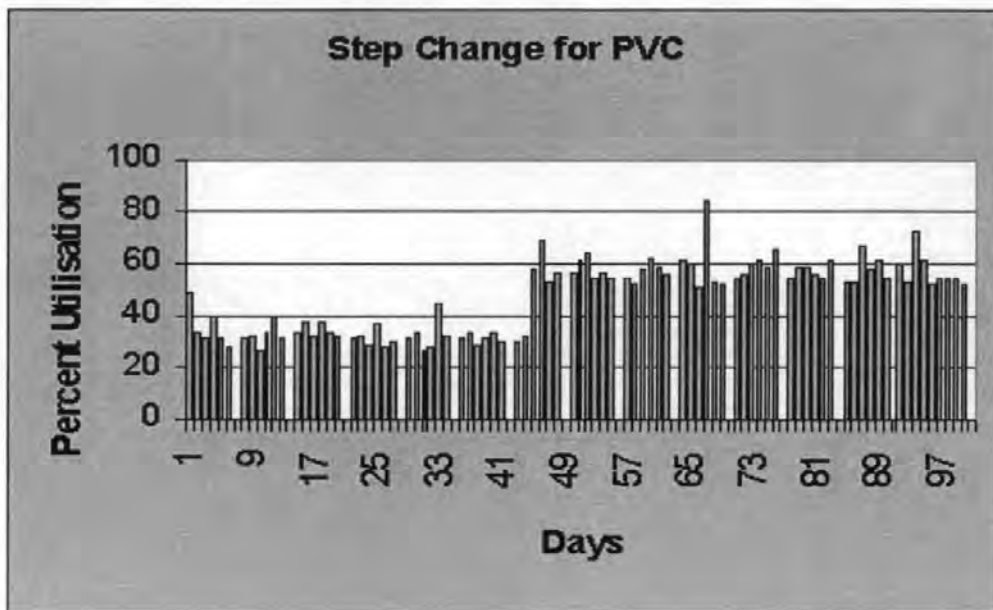


Figure 5.15: Illustration of a Step Change for a Single PVC

5.2.4 Traffic Patterns

Paxson [Paxs91] and Caceres et al [Case91] made the observation that "approximately 72 per cent of the traffic was between systems connected by the same Ethernet, and the rest was between a system on the Ethernet and some other system connected with the Ethernet

through a WAN". This observation is widely used today as a rule of thumb with the LAN/WAN traffic division to be roughly an 80/20 or 70/30 split.

The results obtained for the present research were not different. Around 70 per cent of the traffic observed was LAN traffic using various protocols which was flooding over the whole network across different Ethernet segments. Around 30 per cent of the traffic was found to be utilising the router to the outgoing WAN. Dividing the traffic into different protocols, it could be seen that around 95 per cent of the LAN traffic was created by the IPX protocol. The rest was mainly created by TCP/IP. The majority of the WAN traffic however was created by TCP/IP traffic.

5.2.5 Locality

Related to network rateability are metrics of traffic locality, which reflect the geographic non-uniformity of traffic distribution. Previous studies have established the existence of network traffic locality, in particular short-term traffic locality in specific network environments for selected granularities of network traffic flow [Pax95]. Some studies also find evidence for locality even in networks of wider geographic scope [Ray93]. Locality refers to the concentration of traffic among a small subset of possible network addresses. Taking advantage of locality can have great potential benefit, especially in large-scale infrastructures which exhibit substantial gaps in traffic volume between the heavy and light flow.

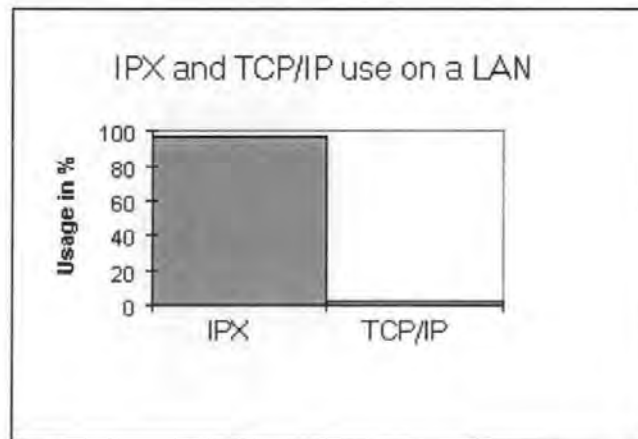


Figure 5.16: IPX to TCP/IP ratio used purely on a LAN

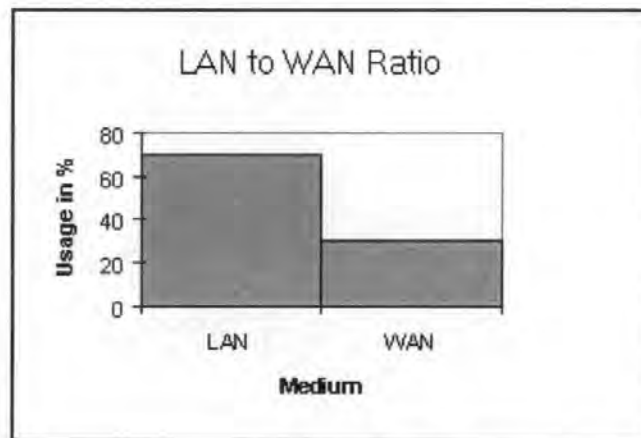


Figure 5.17: LAN/WAN Ratio

Another interesting observation was that most of the traffic involved two IPX servers, as can be seen in Figure 5.16. These servers are used by most PC clients on the main University campus, sending or receiving around 60 per cent of the existing traffic on this segment. It can also be seen in Figure 5.17 that the ratio between LAN and WAN use is 7:3.

In Figure 5.18 it is shown that the UNIX hosts took up approximately 10 per cent of the traffic. The PCs, which most of the time act as Novell clients, sent or received 25 per cent.

"Others" indicate router and destination nodes, which are not physically on the segment, but were involved in the traffic patterns.

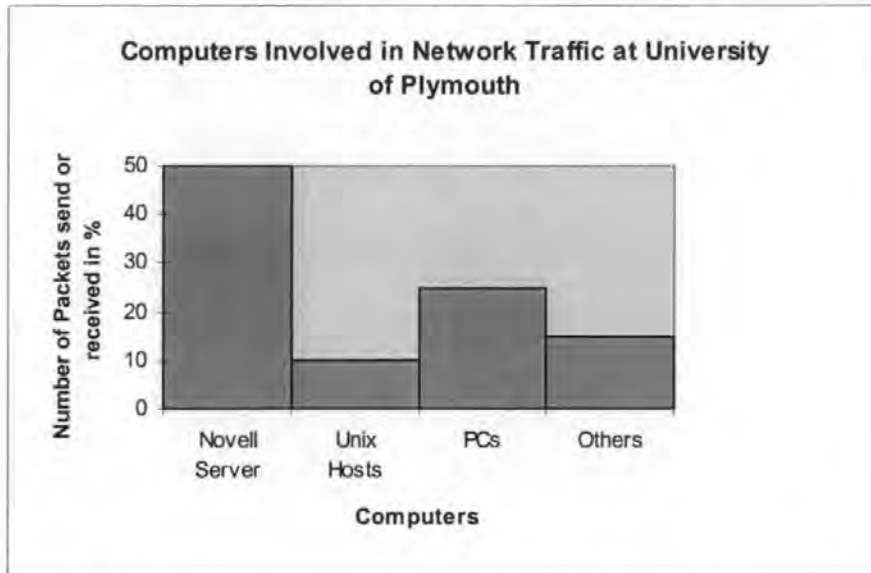


Figure 5.18: Ratio of Computers involved in Traffic at the University of Plymouth

5.3 Application Signatures

The hypothesis is that applications leave a signature when used over networks. Data that are sent from the original source and travel to the destination utilise the network in a very similar manner if no strong congestion is experienced. Should the network experience congestion, the unavoidable frame losses will alter the signature of the application and change the statistical property of the data. However, certain properties remain the same, even after a network disruption and congestion. This is due to the protocols used and the parameters set in the environment of the traffic source. It is thought that these findings could help in the understanding of the traffic characterisation and the processes involved.

5.3.1 Introduction

Data from user equipment are broken into multiple frames and sent according to the protocol used. This is carried out in a specified manner over the network. Traditional

network theories imply that the packet size distribution is exponential and often work on average packet size distribution [Ker95].

One of the original profiling models was inspired by the objective of characterising network traffic locally. Locality describes the phenomenon of a distribution of traffic to, from, or among a selected few sites [Chiu92]. Network locality presents an obstacle to characterising packet arrival processes as Poisson.

Previous attempts to characterise Internet arrival processes have concentrated on traffic by component, e.g., telnet and ftp. Caceres et al [Cas91] provide evidence that characteristics of an instantiation of a specific TCP application do not depend on the environment, but that characteristics of the conversation arrival process itself do depend on the environment. They state that they were “unable to form a realistic and network-independent model of conversation arrivals, since the arrival parameters depend on geographic site, day of week, time of day, and possibly other factors”. [Pax95] provides further evidence that traffic patterns vary greatly, both over time and more so from site-to-site, not only in traffic cross-section but also in connection characteristics.

Paxson and Floyd used fifteen wide-area traces to investigate the extent to which TCP arrival processes (session and connection arrivals, file Transfer Protocol (FTP) connection arrivals within FTP sessions, and telnet packet arrivals) are Poisson. They find that user-initiated TCP session arrivals, e.g., remote login and file transfer, reasonably reflect Poisson processes with fixed hourly rates, but other connection arrivals are less convincingly Poisson. Furthermore, they find that modelling telnet packet arrivals as exponential inaccurately reflects telnet burstiness. Finally, they determine that FTP

connection arrivals within ftp sessions come bunched into “connection bursts”, the largest of which are so large that they completely dominate ftp traffic [Pax95].

5.3.2 Traffic Patterns

The knowledge of various network properties is very important for modelling and simulating networks. With such detailed knowledge it is easier to predict the capacity requirements and delay characteristics of the network.

The packet size distribution is one of the characteristics of protocols. In the process an investigation of the properties of packet size distribution, inter-arrival time of packets and packet trains as well as the predictability of traffic patterns of individual applications were carried out.

The research focuses on the most popular protocol, the Internet Protocol Suite (TCP/IP). The general characteristic of a building block is that it would identify a sequence of packets travelling from the source to the destination, which is identifiable and can assume a particular type of service or program. These packets are all-identifiable and belong to the same application. The application can be either a WWW page or any other repeatable network download activity.

Breaking traffic up into components and analysing them is inevitable. Advanced technologies like ATM require quality of service (QoS) parameters for the application. Examining the distribution of conversation inter-arrival times by application at various sites is also relevant, as applications may differ by site. The arrival characteristics of traffic on a regional network may differ from the same application used over a WAN. This

is due to time-sensitivity of the protocol, its congestion control algorithms and the various reasons for delay. These are propagation delay and congestion delay.

However, it is felt to be more and more important to characterise the aggregate arrival process in relation to transport protocol and application. This approach will be increasingly relevant as different types of Internet traffic proliferate, decreasing the proportion of traffic carried by traditional protocols. The model in [Jain86] provides parameters with only five TCP applications (FTP, remote login, net news transport protocol, simple mail transfer protocol, and telnet), that represented a major proportion of Internet traffic. A larger part of the data on the Internet, at the present time, results from WWW traffic.

The measurements indicate that at least in the environments that were studied, current IP traffic consists of more short-transaction type traffic rather than longer-term flows. The short packets and short flows together shed doubt on a strategy of optimising for long flows that are in fact the minority case. However, it is noted that many new applications may change this characteristic of Internet environments as they introduce traffic flows with different behaviour. In particular, real-time continuous media flows tend to exhibit greater duration and flow volume.

5.3.3 Visual Study

The aim in this research is to show the validity of these claims for a ‘data fingerprint’. This will be done via visual study of graphs and simple statistics of real life applications. To do this, different applications downloaded over a LAN have been used. The reason why a LAN is chosen over a WAN is purely philosophical. As the traffic usually

originates from a source connected to a LAN, the original form of the traffic profile can be found very “close” to the source. This point of view was adopted and leaves room for later discussions in further research.

All graphs in Figure 5.19, Figure 5.20 and Figure 5.21 represent the same WWW page, which has been downloaded under various traffic conditions on the LAN. The utilisation on the LAN varied from 5 per cent, 20 per cent, and 30 per cent. It should be noted that a number of applications and WWW pages have also been measured and have shown the same effects as the one represented below.

In the following, it can be seen how similar Figure 5.19, Figure 5.20 and Figure 5.21 look. Even with a big utilisation difference, no major difference in the graphical outlook can be found. The short utilisation variations obviously have little or no effect on the fragmentation of the packets.

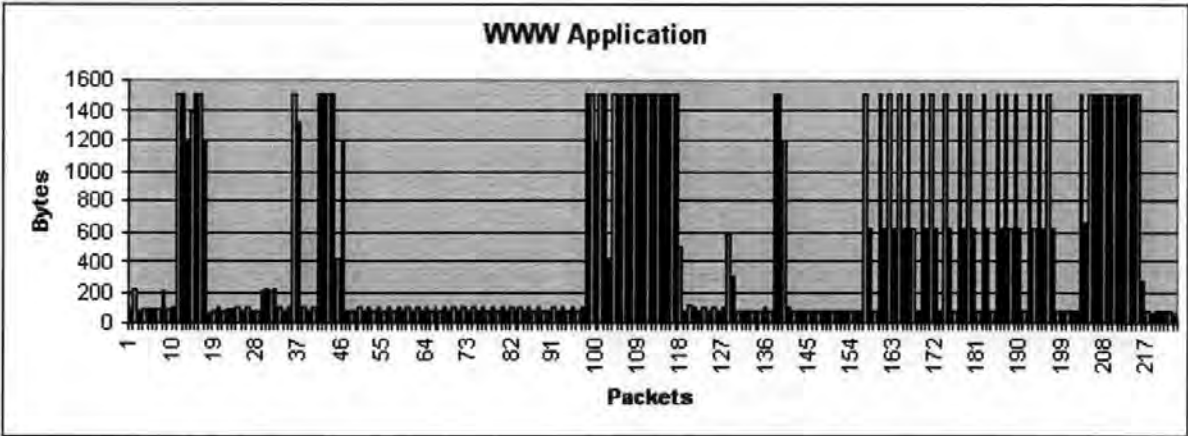


Figure 5.19: Measurement at 5 per cent utilisation

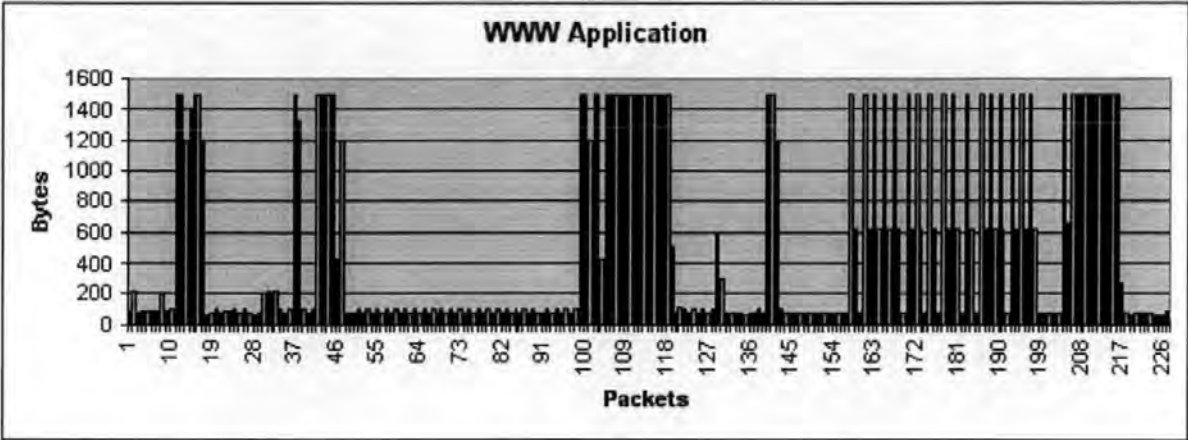


Figure 5.20: Measurement at 20 per cent utilisation

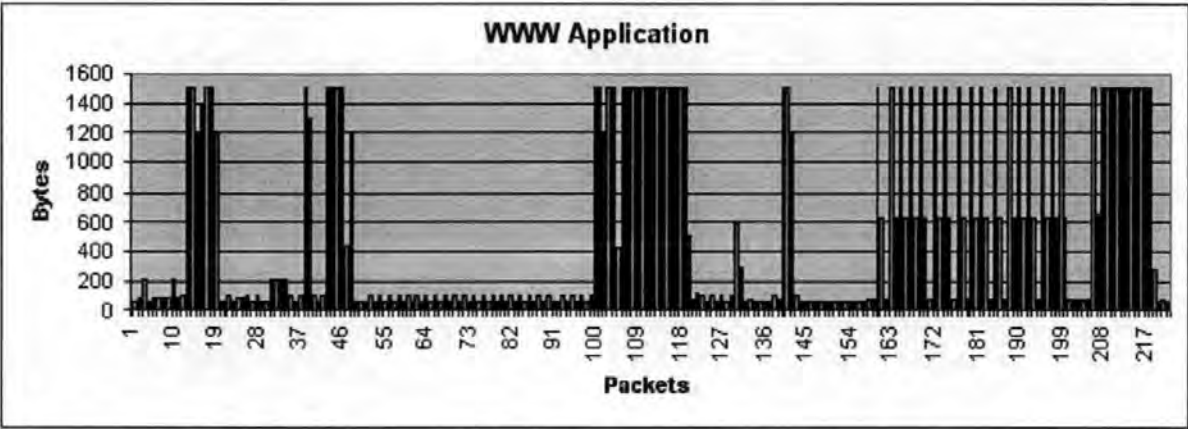


Figure 5.21: Measurement at 30 per cent utilisation

5.3.4 Packet Size Distribution

The packet size distributions are shown in Figure 5.22, Figure 5.23 and Figure 5.24. It can be seen that the packet size distributions are very similar. It was found that the number of packets with the same length was very much the same in all traces. This can be seen as one property of an application signature. A large number of packet traces were conducted and a few are displayed in this thesis.

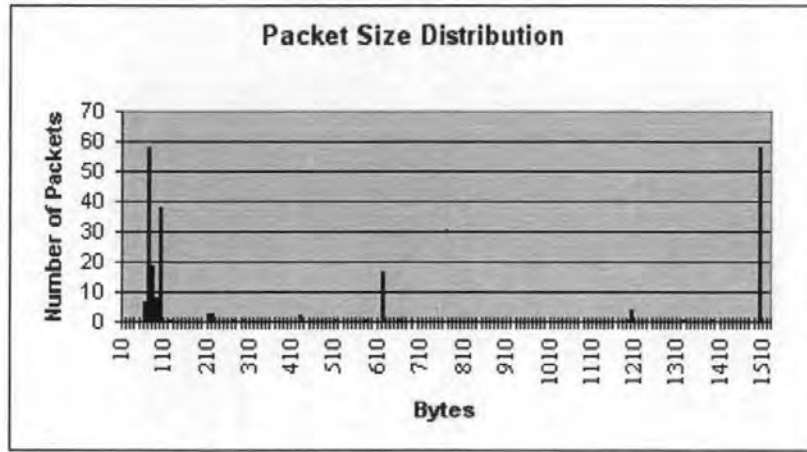


Figure 5.22: Packet size distribution at 5 per cent utilisation

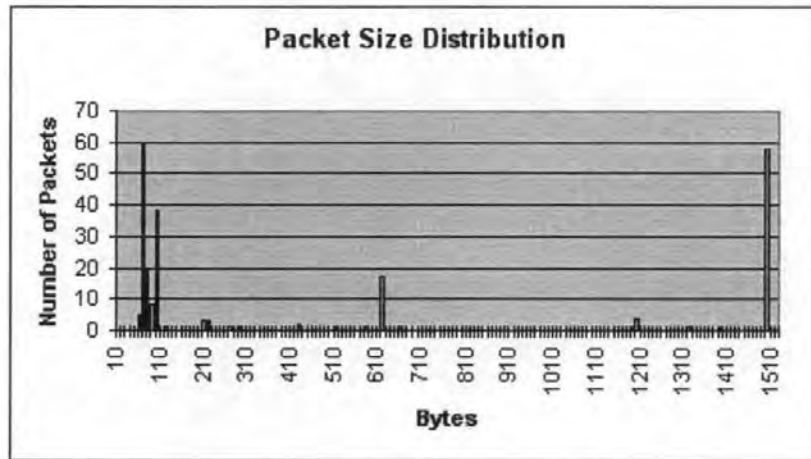


Figure 5.23: Packet size distribution at 20 per cent utilisation

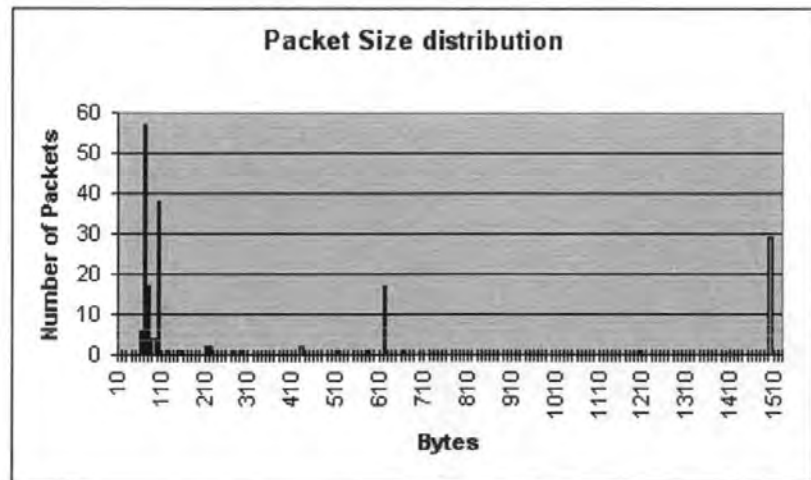


Figure 5.24: Packet size distribution at 30 per cent utilisation

5.3.5 Responder / Originator Ratio

Using this model for traffic profiles for prediction of traffic, the relationship between the distributions of the originating source and responder must be investigated. It would often be desirable to know how many responder bytes to expect given a particular application. Previous research of telnet session has shown a ratio of 20:1 between the bytes generated by the computer in a remote login session and those generated by the user. The research differs in the way that every application seems to have a distinct ratio, which is hardly changing. This ratio can vary by application from 5:1 to 40:1. In any case however, the same application created the same ratio in all trials. An example of an application profiled over several days at different times can be seen in Table 5. 1 and more ratios can be found in Appendix C, Sections C.4.1 to C.4.7.

Application profiled over a period of 2 weeks				
Date	Time	Packets Sent	Packets Received	Ratio
26.1.1998	10:17	719570	102357	7.03 :1
26.1.1998	12:33	763756	98361	7.76 :1
26.1.1998	13:41	623192	94832	6.57 :1
26.1.1998	15:16	727534	92163	7.89 :1
26.1.1998	15:58	604304	94140	6.42 :1
26.1.1998	16:39	728338	97459	7.47 :1
27.1.1998	10:17	614109	97004	6.33 :1
27.1.1998	12:33	759930	99643	7.63 :1
27.1.1998	13:41	768349	100140	7.67 :1
27.1.1998	15:16	758530	102058	7.43 :1
27.1.1998	15:58	668596	104713	6.39 :1
27.1.1998	16:39	780997	102780	7.60 :1
28.1.1998	10:17	681992	100396	6.79 :1
28.1.1998	12:33	759842	98079	7.75 :1
28.1.1998	13:41	641788	100430	6.39 :1
28.1.1998	15:16	751411	97756	7.69 :1
28.1.1998	15:58	750841	97158	7.73 :1
28.1.1998	16:39	750669	99912	7.51 :1
29.1.1998	10:17	685841	101925	6.73 :1
29.1.1998	12:33	722731	98661	7.33 :1
29.1.1998	13:41	641757	98865	6.49 :1
29.1.1998	15:16	769216	100293	7.67 :1
29.1.1998	15:58	592159	97477	6.07 :1
29.1.1998	16:39	685894	94101	7.29 :1
30.1.1998	10:17	681043	92317	7.38 :1
30.1.1998	12:33	656829	89047	7.38 :1
30.1.1998	13:41	546144	89467	6.10 :1
30.1.1998	15:16	661234	91856	7.20 :1
30.1.1998	15:58	636657	93316	6.82 :1
30.1.1998	16:39	604081	91550	6.60 :1
02.2.1998	10:17	646682	92960	6.96 :1
02.2.1998	12:33	591378	90951	6.50 :1
02.2.1998	13:41	569339	87495	6.51 :1
02.2.1998	15:16	517919	84000	6.17 :1
02.2.1998	15:58	524376	84252	6.22 :1
02.2.1998	16:39	558793	86376	6.47 :1
03.2.1998	10:17	565233	88522	6.39 :1
03.2.1998	12:33	580602	87145	6.66 :1
03.2.1998	13:41	554185	85318	6.50 :1
03.2.1998	15:16	536411	87133	6.16 :1
03.2.1998	15:58	613352	87539	7.01 :1
03.2.1998	16:39	605827	90811	6.67 :1
04.2.1998	10:17	535207	87362	6.13 :1
04.2.1998	12:33	606833	89756	6.76 :1
04.2.1998	13:41	621109	89509	6.94 :1
04.2.1998	15:16	614669	92121	6.67 :1
04.2.1998	15:58	581299	95843	6.07 :1
04.2.1998	16:39	633312	94296	6.72 :1
05.2.1998	10:17	593901	92482	6.42 :1
05.2.1998	12:33	581537	93932	6.19 :1
05.2.1998	13:41	629597	90713	6.94 :1

05.2.1998	15:16	626286	90781	6.90 :1
05.2.1998	15:58	545204	87797	6.21 :1
05.2.1998	16:39	566540	88490	6.40 :1

Table 5. 1: Responder to Originator Ratio

5.3.6 Extension of Packet Train Model

The packet train model in Chapter 4, Section 4.6, describes and defines a few parameters.

For the application signature two of these parameters are important:

1. Protocol modelling
2. Predicting the likelihood of the next packet destined for the same target

The research findings in this chapter show a new parameter, which is very interesting and could be used for signatures and traffic prediction in congestion-control mechanisms. It is the probabilities of packet sizes inside a packet train. Depending on the profiled object, the probability of the next packet being the same size as the first packet of the train is relatively large. If, for example, a 1500 byte packet was detected, the probability of another packet with the same size is high. This can be viewed as an extension of packet train model.

The probabilities change depending on the application or service used. One example of a traced application was as follows: Out of 1080 packets around 900 pairs of packets had the same size. The rest of the traces showed around 90 pairs where the first packet was larger than the second, and 90 pairs of packets where the first was smaller than the second. Around 90 per cent of the time the second packet was as large as the first one.

The same method can be used for the relation of the first packet to the third to establish the ratio between occurrences longer than two packets. In Figure 5.25 and Figure 5.26 two packet trains from one application can be seen. The trains flow from right to left, and the similarity can be seen.

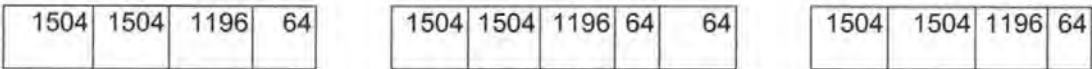


Figure 5.25: Packet train one

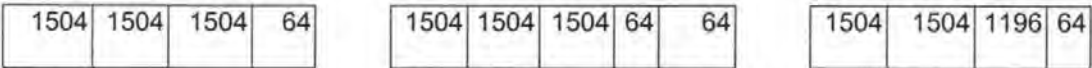


Figure 5.26: Packet train two

When simple statistical analysis is applied and the autocorrelation is tested, the following results can be seen in Table 5.2, Figure 5.27 and Appendix C, Section C.9.

Lag	Correlation
1	0.757351
2	0.615631
3	0.544482
4	0.5296
5	0.49611
6	0.473742
7	0.414315

Table 5.2: Autocorrelation for packet sizes inside packet train

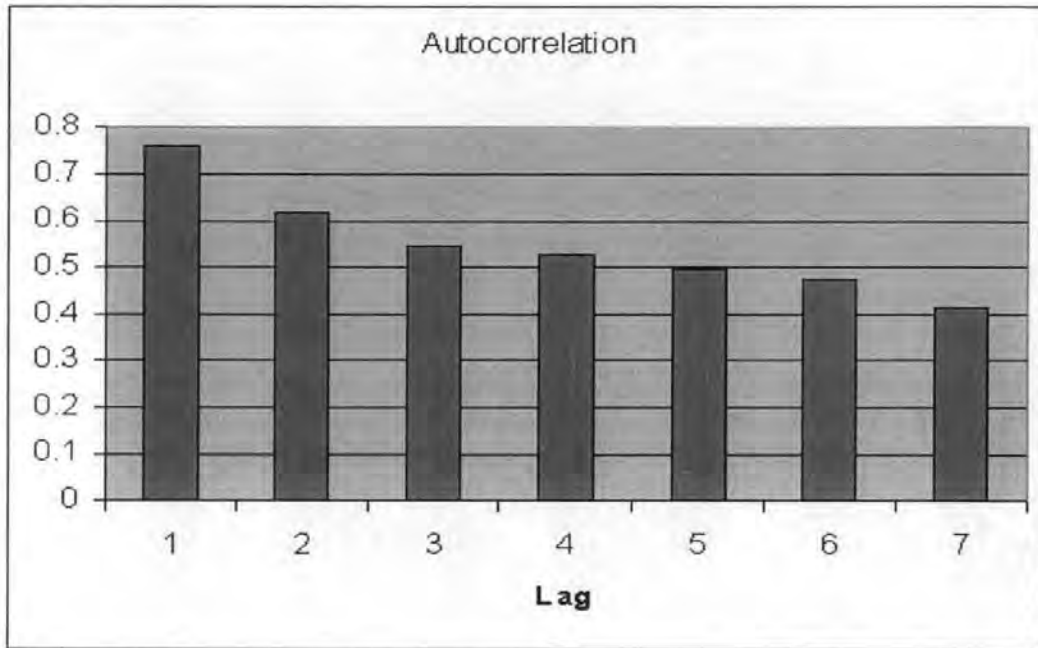


Figure 5.27: Autocorrelation for packet sizes inside a packet train

The analysis that was performed on a series of applications was to look into the packets and see their packet size. The time intervals between the packets form a time series. The autocorrelation function (ACF) shows the relationship of the packets t_i and t_{i-k} and is expressed as lag k .

The autocorrelation function always lies between -1 and 1 . A negative ACF implies a reverse relationship. A positive autocorrelation function shows a direct relationship. A zero autocorrelation indicates independence. As can be seen from the graph in Figure 5.27 the autocorrelation for the first lags is relative high. This would therefore imply direct dependence. The conclusion therefore is that the packet train model for packet size distribution can be used for prediction, and will be used to create a packet train profiler later in Chapter 7.

5.3.7 Summary

A number of graphs and statistics have been presented, which confirm the hypothesis that applications have certain characteristics. This behaviour varies from application to application, but finds very similar characteristics when one application is used at different times or by different users. It has been found that in general the graphical and simple statistical presentations are very useful to get a feeling for this complex subject.

The essence of the argument presented here is that while traffic cannot be modelled exactly in a statistical sense, it can still provide evidence of simple statistical properties with good approximations for the use in empirical models. Furthermore, these models can be used in simulations to look at other aspects of WANs such as congestion control.

5.4 Traffic Characteristics of WWW Sessions

The growth of the World Wide Web (WWW) and its use is causing a dramatic impact on networks. The use of the older protocols and applications are being replaced for WWW usage, by quasi-standard interfaces like Internet Explorer from Microsoft and Netscape's Navigator. The following examines whether existing models of traffic theory can be used for WWW session arrivals. The need to understand the underlying statistical properties of the distribution of WWW sessions, relative to time over a corporate network, influences the design and modelling of WANs. Such information can be important since the behaviour of the WAN and the corporate network will depend on the performance and QoS requirements of the network.

There are two distinct but equally important components to any session / time distribution:

- The session duration distribution, which is the distribution of the difference between the initiation and the end of a transaction of a session.
- The inter-arrival time distribution, which is the distribution of the difference between the connection time between two consecutive sessions.

The following parameters will be analysed and displayed in the following section:

- session duration for groups of users, and
- inter-arrival times of users for whole corporate network

5.4.1 Flow Inter-Arrivals

The focus in this section is on the distribution of inter-arrival times between flows. Several studies have found evidence that Internet packet arrivals do not seem to exhibit Poisson behaviour [Man91] [Cas91 / Che89] [Muk88] [Kun94]. In contrast, as will become apparent, the Poisson model does seem to adequately characterise Internet flow arrivals as defined by certain parameters.

Previous attempts to characterise Internet arrival processes have concentrated on traffic by component, e.g., telnet. [Cas91] provide evidence that characteristics of an instantiation of a specific TCP application do not depend on the environment, but that characteristics of the conversation arrival process do. [Cas91] admit that they were “unable to form a realistic and network-independent model of conversation arrivals, since the arrival parameters depend on geographic site, day of week, time of day, and possibly other factors”.

[Paxn95] provides further evidence that traffic patterns vary greatly, both over time and more so from site-to-site, not only in traffic cross-section, but also in connection characteristics. Paxson and Floyd use fifteen wide-area traces to investigate the extent to which TCP arrival processes (session and connection arrivals, connection arrivals within File Transfer Protocol (FTP) sessions, and Telnet packet arrivals) are Poisson. They find that user-initiated TCP session arrivals, e.g., remote login (rlogin) and file transfer, reasonably reflect Poisson processes with fixed hourly rates, but other connection arrivals are less convincingly Poisson. Furthermore, they find that modelling Telnet packet arrivals as exponential inaccurately reflects telnet burstiness. Finally they determine that connection arrivals within FTP sessions come bunched into “connection bursts”.

Breaking traffic up into components such as the above studies is helpful when the dominant application in the traffic cross-section at a given site overshadows many characteristics of overall traffic measurements. Therefore, characterising the dominant application is very close to characterising the overall workload. Examining the distribution of conversation inter-arrival times by application at various sites is also relevant. Applications may differ by site, e.g., the arrival characteristics of network news transfer protocol (nntp) traffic on a regional network may differ from those on a larger backbone, due to how the nntp protocol distributes news.

However, it is felt that it is important to characterise the aggregate arrival process, in regard to application and user groups. This approach will be increasingly relevant as different types of Internet traffic proliferate, decreasing the proportion of traffic carried by traditional protocols.

5.4.2 Measuring Conditions

To investigate every WWW activity on a corporate LAN, a network analyser was installed and recorded all sessions. For each transaction a record can be set up containing the starting and ending times of a WWW session, with the originating address and destination address. These records can then be analysed to provide a more detailed profile of session distribution for individual addresses, for groups of addresses and for a complete corporate LAN. As it can be imagined, it would need a few hundred pages to display all the records. Therefore, only the analysed data is displayed in the following sections.

The theory indicates that a sequence of events recorded with regard to the time at which they occurred are taken as independent and will give rise to a negative exponential probability distribution. In the study of traffic theory and telephone networks two events are often examined. They can be either the start and finish times of a call or the start times of two successive calls. Either the duration times or the inter-arrival times of these sessions can then be represented as having negative exponential distributions. If the inter-arrival times follow the negative exponential distribution described, then the distribution will follow the Poisson distribution.

For this reason, phenomena such as sessions within a communications network found to have negative exponential distributions of inter-arrival time and duration are commonly referred to as having Poisson characteristics or obey the Poisson model. The widely presumed independence between sessions has meant that considerable sections of the theoretical work carried out to date, with regard to data networks, have been based on the Poisson assumption.

5.4.3 Empirical Results

The negative exponential and related Poisson distributions provide typical examples of theoretical distributions. Even in instances where the real network may be modelled in this way, there will always be areas in which observed data will deviate from the idealised form. Variations like these are inevitable in a situation such as the AT&T network in which session patterns seem to change from one moment to the next, from day to day and from week to week. A slight lack of precision must be taken into account when testing for distributional fit – perfect exponential curves and straight-line logarithms are unlikely to be found.

The following Figure 5.28 and Figure 5.29 show the session duration for individual groups. These show a typical session duration distribution for a specific LAN segment. The two examples represent the administration staff of AT&T for two different days. More data is available in Appendix C, Sections C.5.1 to C.5.3. For any recorded network session, the number of sessions will of course be finite while the theory states an infinite number. It is therefore unlikely for the inter-arrival times and duration to have occurred with exactly the frequency predicted by the Poisson model. However, the amount of data traced should be enough to give an impression of the distributions. In a perfect negative exponential distribution, sessions of any positive length, no matter how long, occur with non-zero frequency and no two inter-arrival times can occur with the same frequency. Other network factors peculiar to the session of interest will demonstrate the inaccuracy of the real world examples. There may be, for example, at a particular time, a good reason why sessions within a certain range of lengths occur more often than others. This is especially true for short measurements or extremely long sessions. Often sessions with just a few bytes are sent, which represent polling of equipment or time dependent updates of

WWW pages. Especially automated WWW sessions, which access the same sides and poll for new information, show this occurrence, as one accessed WWW page only changes its traffic profile when strong congestion occurs.

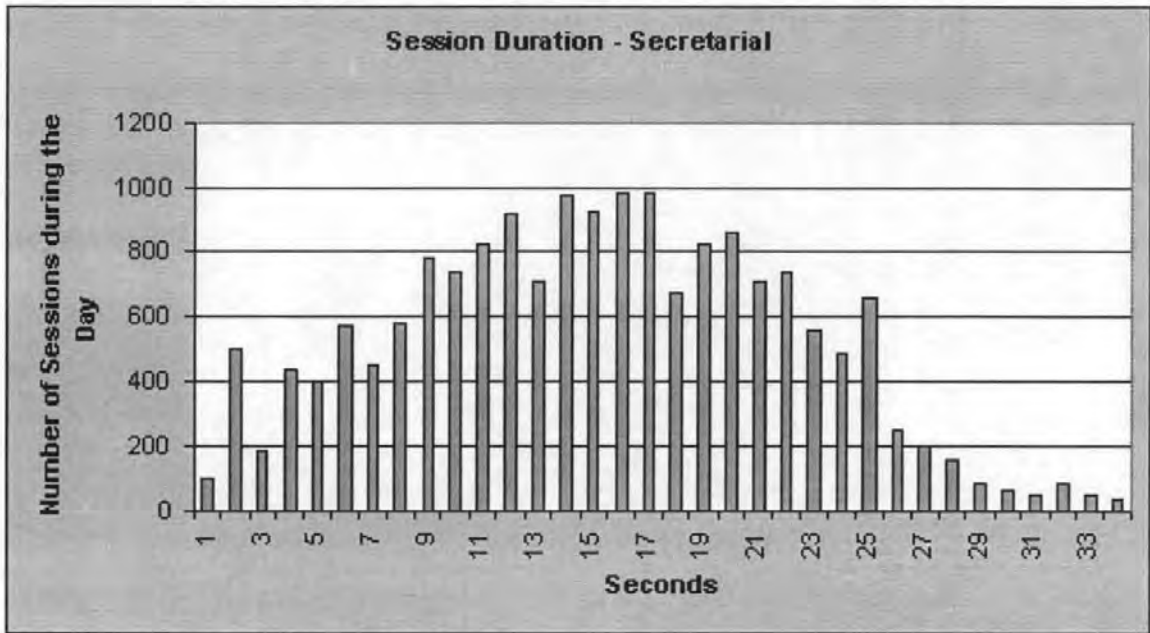


Figure 5.28: Session duration for the Administration LAN segment during day 1

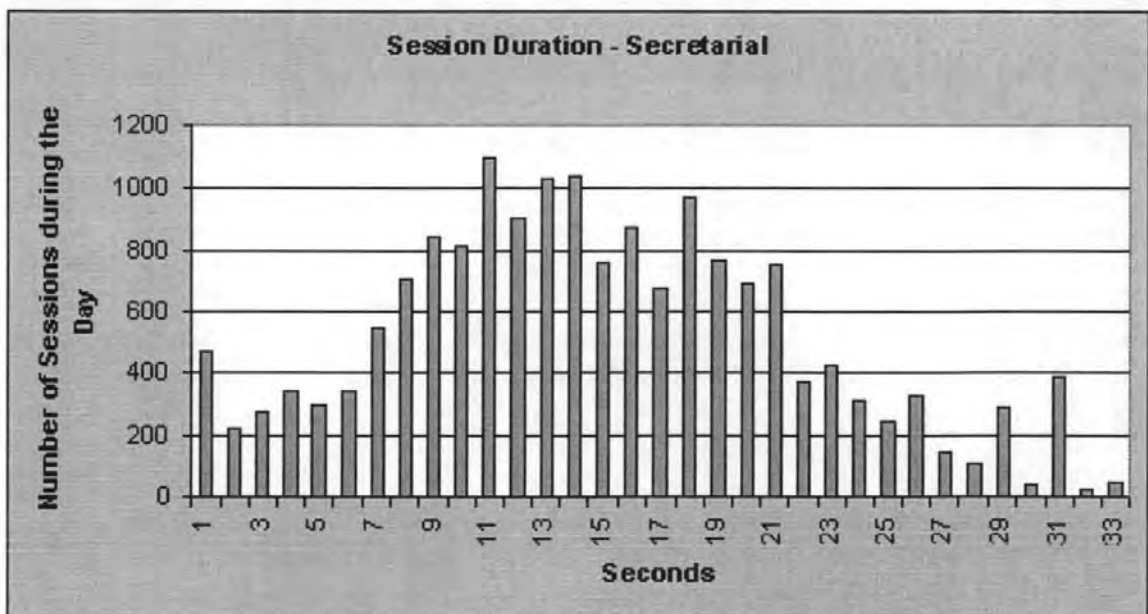


Figure 5.29: Session duration for the Administration LAN segment during day 2

It is interesting to note that most, but not all, user groups seem to have a distinctive “signature” when establishing a session. Usually all users connect to an initial WWW page, which is set up as the home-page on the internet browser. From there on users tend to go to either a location which they usually use, or a search engine, which acts as an Internet directory.

Typical results for user inter-arrival time frequencies are given in the Figure 5.30 and Figure 5.31, displayed in standard and logarithmic form. The approximation for the negative exponential frequency distributions was noticeably better for larger samples than for small samples, presumably since individual peculiarities are more likely to affect experiments with fewer measurements and group behaviour mentioned earlier. The sampling period was conducted for over 1 month and more than 1 million log files have been analysed. In general, the fit of the distribution was good in the majority of cases for which the sample was at least large enough.

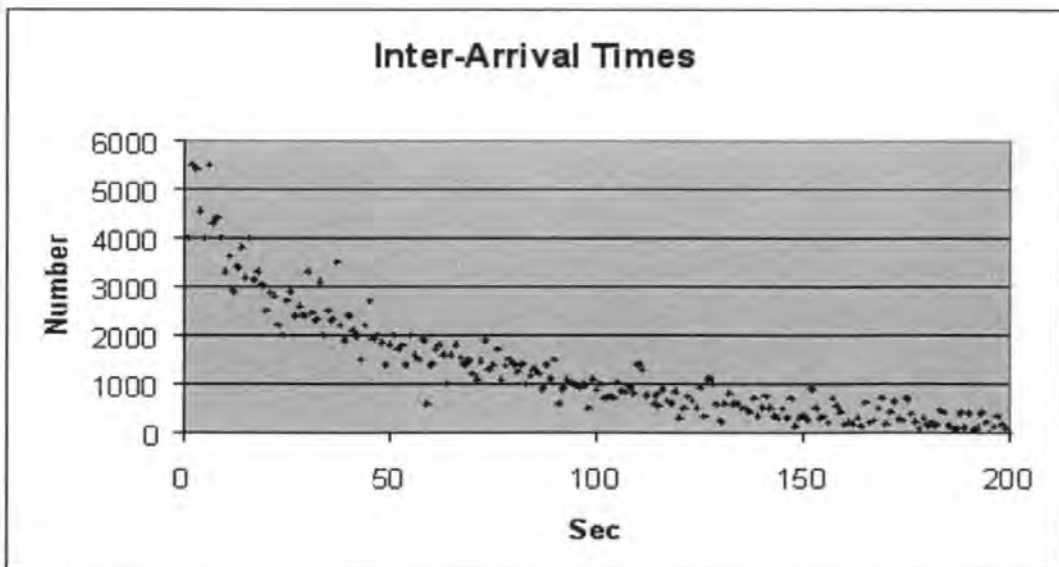


Figure 5.30: Inter-Arrival time of users

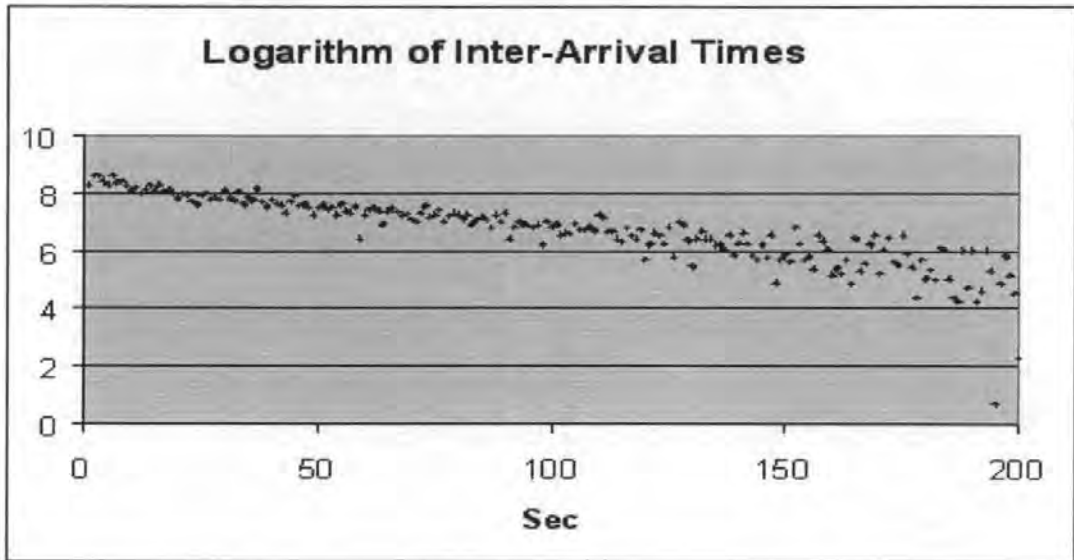


Figure 5.31: Logarithm of inter-arrival times

5.4.4 Conclusions

The conclusions that can be drawn from the study are as follows:

- There is indeed a pattern to the sessions made by specific WWW groups on a LAN. This pattern consists of user groups that tend to access the same WWW pages over and over again, e.g. stock quotes or news. This would imply that user sessions are not totally independent in their choice of WWW sites and this specific user behaviour creates unique probability distributions. Usually these distributions are not exponential and can vary greatly. Maybe the reason for this is that user groups often have similar interests found in their jobs or hobbies.
- If the whole corporate network is analysed the specific distributions of the individual groups tend to merge and produce a negative exponential distribution, as many addresses and users tend to use many different locations.

- In some cases it is possible to quantify the distributions and it would appear that the distributions remained relatively constant over the period of the study. However, it was also observed, that some departments or user groups do not have a distinct distribution at all

The measurements indicate that current IP traffic still consists more of short-transaction traffic rather than longer-term flows. It is noted that many new applications may change this characteristic of Internet environments, as they introduce traffic flows with different behaviour. Particular real-time continuous flows, which tend to show greater duration and flow volume are increasingly popular applications on the Internet. Such a shift in the flow profile will change existing models of the current datagram Internet, which usually consist of many bursty sources.

Chapter 6: Network Planning System

6.1 Introduction

Frame Relay policy considerations and network analysis must begin to interact in a way not previously recognised or implemented by network providers. One way of using the full potentials of this technology is by over-subscription of capacity. To be compliant with the commonly used terms in Frame Relay, the name Overbooking Factors (OBF) will be used. By overbooking the available capacity the provider takes advantage of the fact that not all PVCs use their full CIR at the same time. PVCs, which are silent, provide busy PVCs with their capacity and consequently a sharing of trunk resources.

One of the aims of this project was to develop a methodology, which allows a more accurate design of Frame Relay networks by using traffic profiles and other parameters of every individual PVC and utilise trunk resources in the best possible way. The hypothesis is that individual OBF can be set to gain both advantage in the service provided to the user as well as economy to the network provider. The basic idea is to change the nominal booking factors from a static value only dependent on CIR, to individual overbooking of PVCs dependent on various factors, e.g., recent actual usage and importance of customer. PVCs with a high overbooking will be mixed with PVCs with a low overbooking. This was not done previously, as all PVCs with the same CIR were treated equally regardless of their risk parameters.

In the face of the current evolution of global information infrastructure, which is expanding both in complexity and sophistication of applications, measurements and experience offer evidence to support the hypothesis that overbooking should be the result of the whole information base available to the network planners, not just the size of a CIR.

The results in this chapter have been obtained over a period of 12 months from the AT&T Frame Relay network in Europe. In data networks it is important to plan for the busy period, which is defined as the period of the day utilising the network most and with the greatest risk of losing data.

The introduction of new overbooking rules will improve customer service, increase reliability of the network and thereby reduce costs. Changing the overbooking factors is a crucial component and must therefore be implemented in the existing planning process. To demonstrate the theory the results will show how

1. to balance the Frame Relay network; and
2. utilise trunks to a specific threshold by using traffic profiles.

6.2 Estimating Network Traffic

The planning and the installation of a new network require knowledge about the network demands. One possibility often used for small networks and LANs is “trial and error”. The problem with this method is that at such a time when a high level of utilisation is noticed, the network has to be subdivided into different segments through the use of local bridges or routers. Implementing a WAN in this way would be very costly and can disturb the network users. If the utilisation level is not recognised soon enough this could result in a slowing down of the whole operational process in a company. To provide a better level of performance to network users, it is important to understand how to estimate network traffic. Therefore the results of a traffic estimation process can be used to determine if any changes should be made to the LAN or WAN.

6.3 Perception

As in most business sectors, the customers quickly identify the lack of service or the quality of the products. The source of the problem is usually found quickly by the staff or manufacturer. The perception is for example that either the products are available or sold out, the service identified as good or bad. This attitude is reflected in every day life and therefore in the users' behaviour on a network. When congestion on a network occurs, the user hardly blames himself for it, even if he is the source and reason for it. The typical perceptions are that it is the network providers' fault, or the network is either too slow or too small, etc. Customers rarely blame themselves for choosing an inappropriate CIR or QoS, which is too small or not appropriate for the required service. Also, mostly the customers are not only the network managers, but also the people inside the organisations who do not understand network technologies and their related problems. This is a point, which has to be understood by network service providers and why the sizing of the CIR, QoS and other requirements cannot be left to customers alone.

PVCs with the same access environment and "User Class" can show very different behaviours and traffic patterns. This is due to the subjective assessment of the managers, slight differences in the technical facts and cultural/organisational differences of the company. Human error and the ability to judge the traffic volumes, duration of traffic, etc., play a very important role. To apply conclusions from the traffic patterns of one PVC to another PVC can be fatal for the performance of a PVC and should be made very carefully. However, there are similarities, which will be shown in this chapter that gives the network provider a better understanding of traffic profiles and the customers.

6.4 Rule of Thumb

The methodology is based on a “rule of thumb” technique as bursts and utilisation levels vary according to user levels, time of day, seasonality and introduction of new applications and service levels expectations to the users. The method differs from the existing AT&T design rules [ATT], which treats all PVCs the same way, independent of their real utilisation.

Most notably, not all PVCs can be treated in the same way, as they use different protocols and applications and, therefore, show a different traffic profile. It is not assumed that the dedicated capacity rate stays within its limits, but rather grounds the definition on real utilisation and peak levels of the individual PVCs. A principal objective is finding the balance and granularity of reconfiguring intervals. This approach can address some fundamental Proprietary Company Frame Relay opportunities, including performance requirements of switches and quality of service levels, e.g. packet losses.

6.5 The Network Traffic Estimation Process

Assuming there is no access to monitoring equipment with which to analyse an existing LAN or the network is in a planning stage, a reasonable estimate of this traffic can be made by considering the functions each network user performs.

To facilitate the traffic estimation process a number of network users can be placed together into a “User Class” category. To estimate the traffic for future users from the same class, the results can be multiplied by the number of workstations grouped into the specific “User Class” to obtain an estimation of network traffic for a similar group of network users. This process can be repeated for different user classes and added together.

The approach was used in [Held92], a methodology developed to estimate network traffic on a local network. A traffic estimation worksheet was used to predict the average and peak traffic that could arise on a typical Ethernet or Token-Ring network.

A similar approach to the “traffic estimation process” (TEP) is used by the wide area network service provider AT&T and other network providers to estimate capacity requirements for PVCs when interconnecting LANs via Frame Relay. This approach requires the user (usually represented by the network manager of the company) to have a good idea about the traffic patterns of the company. The wide gaps in correctness between the estimates for the CIR, however, show that the estimated traffic of users differs very much from the real capacity requirements. Network providers here face a dilemma in “believing” the customer and relying on their judgement. When setting up the required link the network providers can often only hope that the capacity requirement will fit the estimated customer profile. Usually the problems start to occur after the trial period, when the network users start to adapt to the new service and the traffic profile changes.

6.6 Definition of Overbooking

The existing design rules and the technical parameters are described here in relation to some Frame Relay equipment. The Proprietary switches used by AT&T have a parameter, which allows the setting of overbooking factors. This parameter is called the Per Cent Utilisation Factor (%UTIL) and indicates what percentage of the CIR will actually be reserved. This factor is the inverse of the overbooking factor.

The switch determines

$$S_T \geq \sum_i CIR_i * \%UTIL_i * B_{iT}$$

for all trunks T where S_T is the trunk speed

$$B_{iT} = \begin{cases} 1 & \text{if PVC } i \text{ is carried on Trunk } T \\ 0 & \text{otherwise} \end{cases}$$

At the start of Frame Relay it was understood that some network providers had Overbooking Factors (OBF) of 10:1, where it was assumed that users only used 10 per cent of their actual CIR. With time, however, this OBF gradually decreased, as users started to use more of their available resources and Internet and its applications like the World Wide Web (WWW), and other TCP/IP connections started to increase.

One extreme would be an overbooking of 1:1, and in some cases it is certainly required, dependent on the QoS. As a general case it is, however, uneconomical, as the price for this service is much greater and time division multiplexing technologies become more appropriate. Experience shows that an overall overbooking of around 3:1 is a practical compromise for network providers that satisfies most QoS requirements of the users. The question of the overbooking threshold is an economic question, and is driven by the profit margin from the trunk cost, maintenance cost etc. and the asking price from the customer.

To determine the CIR is a difficult process and involves a good knowledge of the local traffic. The network manager has to plan for the expected traffic, keeping in mind that at

very busy times he does not have the same throughput and availability of capacity above the CIR for bursty traffic. Depending on the use of protocols, this can have major influence on the choice of the CIR. Time-sensitive protocols like SNA need more consideration than TCP/IP applications.

Very often the choice of CIR is not influenced by the traffic volume itself, but by the budget a company is able to afford. In cases where the CIR is chosen too low and the general OBF is applied, planners have two inaccurate parameters. In such cases the measured dynamic UTIL is sometimes much larger than 100 per cent. This not only causes congestion, data loss, time-outs of protocols and poor throughput but also is a financial loss to the network provider.

However, the goal of finding the optimum network design is most likely not achievable, as the traffic sources have a level of uncertainty in their profile. Reynolds' approach [Reyn93] settles therefore for a "near optimum design", in which he states that "slight deviations from the optimum conditions are of little importance" and of no great impact.

Examples for actual existing design rules could be as the following:

The %UTIL for PVCs are set at

- 10 per cent for 16kbit/sec
- 33 per cent for all other CIR.

Network planners think that PVCs with higher CIR have a larger impact on the network than PVCs with lower CIR. PVCs with higher CIR are therefore more conservatively

overbooked. This means that all PVCs with a CIR of 16 kbit/sec will be overbooked by 10:1 and all other PVCs by 3:1, regardless of their traffic profile.

This categorising should enable the network provider to maintain a required quality of service. It is assumed that the lower loaded PVCs will allow the higher loaded PVCs to use the available capacity on a trunk, as not all users are using their PVCs at the same time. This assumption is supported by looking at the overall dynamic utilisation of a network. The existing simple overbooking rule results in some trunks being overloaded whilst others are relatively lightly loaded at peak hours.

Hence practice shows that certain trunks tend to have more frame losses than others do. As the network uses fixed routing and all parameters are set manually, it is, therefore, only able to react to these overloading problems by congestion control.

Usual Design Rules

Most network providers have design rules, which apply for all PVCs of a certain size, regardless of the individual requirements. The following design rules for the %UTIL for PVCs are currently used by many WAN providers:

CIR	%UTIL	CIR	%UTIL
4	64	64	24
8	32	96	32
12	21	128	32
16	32	192	32
24	21	256	32
32	24	512	32
48	26	768	32
56	22	1024	32

Figure 6. 1: Usual overbooking table

6.7 The New Methodology

As mentioned before many of the problems of overbooking arise due to misjudgement or ignorance of risk factors other than the CIR. The new methodology also covers these factors and merges them with the Traffic Estimation Process (TEP). It is important to understand that new network design methods will have to use information collected by the whole company. Departments, which were regarded as unimportant to network design, can have important information for the new methodology. This new way of thinking is significant as relationships between departments play a major role for information exchange. The statistical figures of a PVC are not the only property for network design. The influencing factors are represented as parameters and can be divided in 5 categories:

- Asset;
- Threat;
- Uncertainty;
- Strategy; and
- Loss.

Many of the parameters cannot be known by network managers or capacity planners alone. The design of a system that meets user input and requirements, requires information put in by different types of staff and departments such as:

- Network Designer;
- Finance Department;
- Field Engineer;
- Marketing / Public Relations;
- Customer; and
- many other information sources.

The methodology is based upon certain basic principles, shown in Figure 6.2, that are defined as follows and can be translated by the general policy of the network provider into the system.

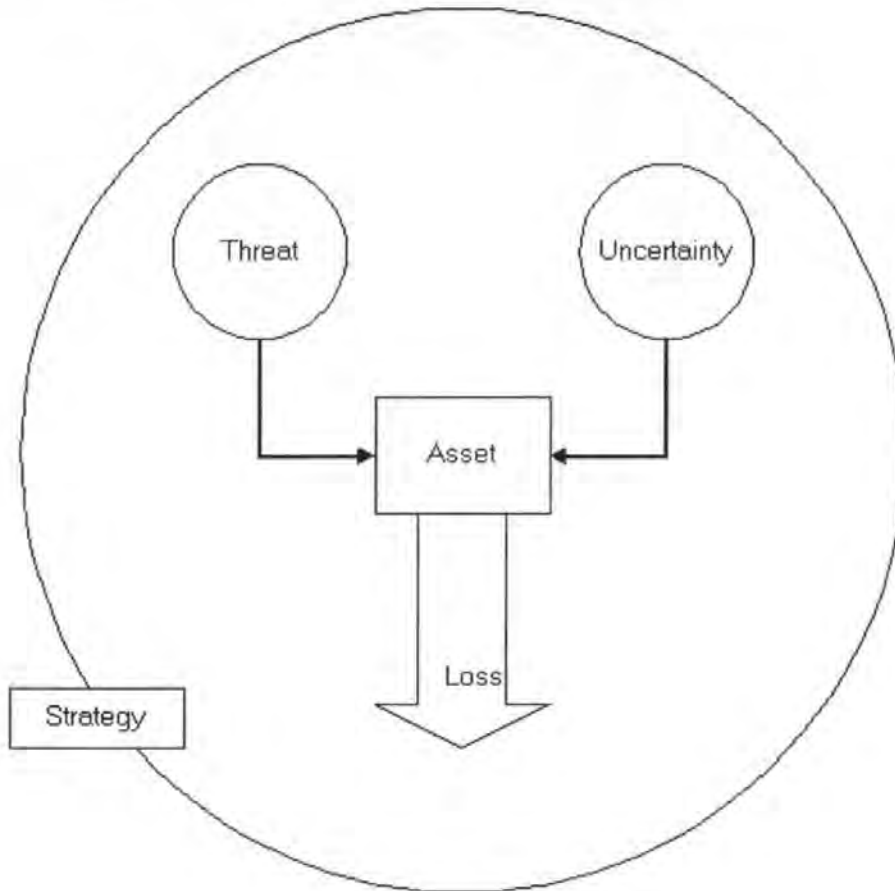


Figure 6.2: Design Methodology

Depending on the functioning of the network provider a loss of QoS has to be expected. In addition, an uncertainty factor influences the assets to make overbooking even more difficult. The threat and the uncertainty are the two factors, which influences the operation of overbooking. The correctness of the overbooking itself can only be verified by the amount of frame losses, delay and throughput.

6.8 Development of the Basics

When looking at the individual PVC performance, it can often be seen how proprietary congestion algorithm reacts to bursty traffic. Sizing a PVC too small or overbooking it too

high, can push traffic to bottlenecks or access points, rather than solving the problems of congestion.

The main reasons for this can be:

- Sizing the CIR incorrectly
- Overbooking is too high
- Permanent overload on the trunk

Pushing traffic to the edges of operation does not solve the problem of congestion for the end-to-end connection; it just moves the problem from the network backbone to the “branches”.

One way of improving the performance of design systems is to include knowledge about the impact of different parameters and the decision taken in generating the design solution. Embedding knowledge of the design and the strategy into the methodology makes it possible to address some of the above points. For example, by the description of the parameters and following through the flow of the methodology a specific overbooking factor can be proposed for each PVC. This overbooking factor is then not only dependent on the specific traffic profile, but also on the environment surrounding it. The design problem is further complicated by the distributed information inside a company. To complicate matters still further they are often dispersed in different time zones and over many countries.

6.8.1 Asset Identification and Valuation

An asset can be defined as:

Any resource or item of information of value to an organisation, which if compromised in some manner would impact on the network, and would result in some form of loss.

This is concerned with identifying the assets of the network environment, e.g. ports, switches, trunks, etc. These assets are analysed to produce a prioritised list according to the criticality to the organisation. This prioritisation is based upon the loss of the network asset to the organisation, for example:

- vital PVC - loss will be critical to the customer and the business
- important PVC - loss to customer would cause severe disruption
- regular PVC - loss creates inconvenience but has no major impact

The network provider faces ever increasing problems in designing a network depending on the requirements of organisational policies and the dynamic response to the QoS requirements and changes of the customers. Some of the major reasons for these problems are the inability of the existing design methods to access requirements not directly related to QoS parameters but still strongly influencing the behaviour of a network or PVC. Therefore parameters must be included in the design process which

- Identify and react to the changes occurring in the task or environment
- Communicate and feedback the implications of a change in the parameters to the designer or network management system
- Identify an acceptable plan once the implications of a change have been calculated.

6.8.2 Threat Assessment

A threat can be defined as:

A potential action manifested by an act, or some technical influence that could result in a loss. In order to determine the potential impact to the PVC, the threats are classified as:

- Size of a trunk, port, PVC
- Overbooking Factor
- Choice of correct protocols

6.8.3 Uncertainty Factor

Assessment of uncertainty in an environment with varying qualities of service requires effective categories of transmission, reflecting the required level of service. Examples of such levels may include information retrieval, real-time video, file transfer etc. These classifications of traffic will not only include the prioritisation of packets in queuing but also prioritised discarding in congestion-control mechanisms. These discarding mechanisms are directly connected to the uncertainty level in a network service. The problem especially arises in networks, which have a “best effort” QoS. Here the assessment is even more important as the allocation of capacity and resources is prior to the service and not in real time. Even when problems are recognised, then upgrading of a service takes time and should be considered in the planning.

High-level goals often qualify if not define the relationship between network design and network policy. The critical point in the evolution of global information infrastructure is that network analysis must include and interact with company policies and is not an

isolated task of pure mathematical design for least cost, least hops etc. Uncertainty also includes the behaviour of a PVC and the stage in a life cycle, which a PVC has achieved.

6.8.3.1 Life Cycle

Any new technology that is being implemented causes an impact on the existing organisation and technology. The impact can occur top down and affect the organisation as a whole. Certain issues like disruption of existing services should therefore be considered in the context of the staff and the organisation. It is important to ensure that the introduction of any new PVC or network does not hinder the existing systems and the life cycle of the PVC.

6.8.3.2 The Trial Phase

This occurs when the PVC is initially set up but does not usually support any crucial function of the company. Technical staff are trying possible scenarios with the technology and experiments with the new set-up.

When a new customer switches services e.g. from X.25 to Frame Relay and maybe to a new network supplier, then usually trial periods are arranged between the customer and the network provider. Trial periods can span a period of 3 months and are assessed afterwards. The service agreement between the parties is usually drawn up before the trial period and should be reviewed to accommodate for the results of the trial period. However, the measurements and research show a lack of understanding from both sides in the difference of the trial period and a PVC which is alive. The following points are often not assessed and can lead to problems:

- When migrating from one service to a new one, the migration period is not fully utilising the new service as both services are in use and share the capacity and system requirements.
- Often new computer systems are going to be upgraded in conjunction with the new service. However the upgrading is not fully completed and the systems are not used in the trial period.
- Financial restrictions put pressure on network managers and result in a “somehow we are going to manage” approach by the customers.
- Customers do not understand the new service or perceive the contract in a different way. For example, CIR and QoS can be interpreted by people in regard to reading different articles, books or publications of standardisation bodies, and not in relation to the specific technical specifications of the network or service provider.

6.8.3.3 The Introduction Phase

This stage is reached when the contract is signed and the technology is initially introduced to some departments. Usually users have not discovered the full potential of the network and are not utilising it to full potential. At this stage the PVC should be monitored closely by the capacity planner.

6.8.3.4 Growth Phase

This stage is reached when existing systems are used over the PVC and affect most departments involved with tasks going over this specific PVC. Users in the company have discovered the full potential of the technology and are using it without restrictions. Here the system can be monitored by a network management system, but the capacity planner should still check on the PVC from time to time.

6.8.3.5 Maturity Phase

This is the stage when the PVC is used to its fullest and has a relative predictable behaviour. A large percentage of PVCs will be in this phase most of the time. When a PVC is in this stage the monitoring can be automated to take utilisation levels and perform the necessary calculations. In this stage there are nearly no maintenance costs for capacity planning. The capacity planner does not have to check on the individual PVCs till “a set threshold for utilisation” is reached. When a set utilisation level is reached then the planner has to decide if it was a one-off effect or if the PVC is going into another phase.

6.8.3.6 Upgrade Phase

This stage usually happens in the maturity phase and shows a sudden jump or reduction of utilisation levels. It happens when new computer systems and services are introduced in the organisation. To recognise this level the network management system which is monitoring the PVC can be set to threshold levels to indicate when an unusual utilisation occurs and then indicate a usual behaviour to the capacity planner.

Any of these stages do not reflect the utilisation level ratio to the CIR of the PVC. These stages merely reflect the users' behaviour in a life cycle of a PVC and its service.

6.8.4 Strategy

The strategy is very important in the influence on design decisions. If the strategy is not known, then miscalculations for trunk usage, hops needed, etc will happen and the result becomes meaningless.

The first task is to focus on setting strategies for re-routing and overbooking of PVCs on the backbone. Especially, the low utilisation relative to the MIR, seen on most of the PVCs used in the backbone, has impact on the dimensioning strategy that can be chosen. But the methodology is also about dimensioning the connection for LAN interconnect services; also end-user performance issues and protocols are regarded. The backbone needs to be optimally fitted to the behaviour of these PVCs, which gives a good reason for studying the behaviour of these protocols. The chapter will be concluded with a description of the most important parameters and a proposal for how they can be used.

6.8.4.1 Re-routing

Re-routing is a procedure that needs to be followed when a trunk is not available, due to some incident. All PVC's that were active on this particular trunk need to be re-routed over other trunks. From the experience at AT&T and other network providers with a large backbone it is known that the probability of one trunk going down in the busy hours is not negligible. So capacity has to be reserved as a backup for this event. The fact that two or more trunks on one node are down is considered as a catastrophe and will not occur very often. The cost usually does not make up for the extra safety.

The re-routing procedure of the PM system is that first PVC's with the highest CIR are re-routed and then the smaller ones. The reason for this is that it is most difficult to fit in the re-routed PVCs with the largest CIR. Now if planners want to be able to re-route the traffic of one trunk, they need to find the nodes with the lowest number of trunks connected, because these nodes have the least capacity for re-routing. In this specific backbone this number is between 2 and 5. The nodes with two trunks connected have one active and one spare trunk, so they have full back-up capacity and would not be regarded in building a strategy for re-routing. Other nodes have three or more active trunks connected.

When one trunk goes down at a node, network providers still want to guarantee (at least) the CIR in all circumstances. Imagine that a node has n trunks connected and that $p\%$ of the trunk capacity is configured as CIR. To ensure that when a trunk goes down all the CIR can be delivered, the following equation exists:

$$p = 100 * (n-1)/n \quad \text{for } n \geq 2$$

This implies that the total amount of capacity actually reserved for all PVC's on a trunk (ΣCIR) may never exceed $2/3$ of the full trunk capacity, if a minimum of three trunks ($n=3$) is connected to one node. In practice most planners will assume that the ΣCIR on a trunk is 70 per cent, as most of the nodes have more than three trunks connected. When no trunk is down in the network, the re-routing capacity will be used as Extended Information Rate (EIR) for active PVCs.

The choice of setting the maximum CIR at 70 per cent implies the following:

If the maximum CIR is 70 per cent of trunk capacity this implies that the re-routing capacity is 30 per cent of total trunk capacity. The largest CIR that can be re-routed on a trunk is 480 kbit/s (30 per cent of 1.60 Mbit/s), meaning that the largest CIR that can be configured for a PVC is 480 kbit/s.

6.8.4.2 Overbooking risks

So the re-routing strategy is to reserve 70 per cent of total trunk-capacity for Σ CIR. In this section it will be assessed what this strategy means for the overbooking risks. In general there are three ways of overbooking:

- trunk based overbooking;
- port based overbooking; and
- time of day overbooking.

6.8.4.3 Overbooking on a Trunk

With trunk-based overbooking the fact is used that the utilisation of PVC's is often much lower than the configured MIR, so that the total configured MIR on a trunk can exceed the trunk capacity. If this is the case the reserved capacity for the CIR can be tuned to the same level as the real utilisation of the MIR. Reasons for under-utilisation of the MIR can be due to over-estimation of capacity needs by the customer, inefficient end-protocols that are not able to fill the reserved capacity or half-duplex end-user applications. If the decision to overbook a trunk is made, the configured sum of the CIR on this trunk exceeds the total available trunk capacity.

As seen from the previous section the maximum CIR was set at 70 per cent of total trunk capacity, so that planners are able to re-route the CIR in the current network configuration. But as this method is deriving a baseline dimensioning strategy, it is also important to guarantee the re-routing of the MIR's when a trunk is down. On the other hand the network providers want to sell as much trunk capacity as possible. Now when all trunks are available the absolute upper bound of data that has to be guaranteed on a trunk is the Σ MIR. So if planners choose the sum of MIR equal to 100 per cent of trunk capacity, the trunks can handle all PVC's requesting capacity at the same instant, when all trunks are available. Also, when a trunk goes down, the network is still able to re-route the traffic of the trunk that has gone down, because 30 per cent of the trunk capacity is free for re-routing.

So the baseline-dimensioning rule for a trunk is that the maximum sum MIR equals 100 per cent of trunk capacity. This allows all the guaranteed data to be transported (sent at MIR) when all trunks are available, even when all end-users burst at the same time. Also re-routing is still possible, as this is done based on the CIR. An implication is that the %util of all the PVC's is set at 70 per cent.

It is obvious that Fixedpackets will be lost in the backbone when a trunk goes down and all PVC's are requesting the Σ MIR at the same instant. But as PVCs only use a fraction of their MIR on average, it can be assumed that the average utilisation of the MIR will be lower than 100 per cent. Now if planners want to guarantee the re-routing of the average MIR when one trunk is down, the following condition has to be met:

The average MIR has to fit in the backbone when one trunk is down. A condition that has to be met is that the mean utilisation of the MIR on a trunk never rises above 70 per cent. If this condition is not met the loss of MIR data is structural for as long as the trunk is down.

Summarising, the chosen dimensioning strategy allows for re-routing traffic based on the CIR. When all trunks are available the backbone can support bursts of all end-users requesting their full MIR at the same time instant. When a trunk is down and all end-users burst together, MIR data will be lost. But with this configuration the network is still able to transport the average utilisation of the MIR, as there is no structural overload of MIR on the backbone when a trunk is down and the mean utilisation of the MIR is below 70 per cent.

This is a very conservative dimensioning rule (a baseline) but it applies to the starting points and conditions of this research. Overbooking on a trunk can be done if the configured utilisation of the MIR (the %Util parameter) can be tuned to the real utilisation of the MIR.

6.8.4.4 Overbooking on a Port

Port-based overbooking means that the sum of the MIR is bigger than the Access Rate (AR) at a port. So when overbooking on port, it is assumed that the probability of all customers on this port requesting their full MIR at the same instant to be negligible. Port-based overbooking can only be done per customer, because at this moment for each port only one router can be connected, so the traffic of one customer will be multiplexed on its

own port. The definition that will be used for port-based overbooking is that the guaranteed capacity sum of the MIR on a port is larger than the AR of the port.

For LAN interconnect the AR of a port is a given value, as it is chosen by the customer most of the times. After the choice has been made for the AR, AT&T derives the MIR that is associated with it. In general, there are three ways of configuring a port, which are displayed in Figure 6.3:.

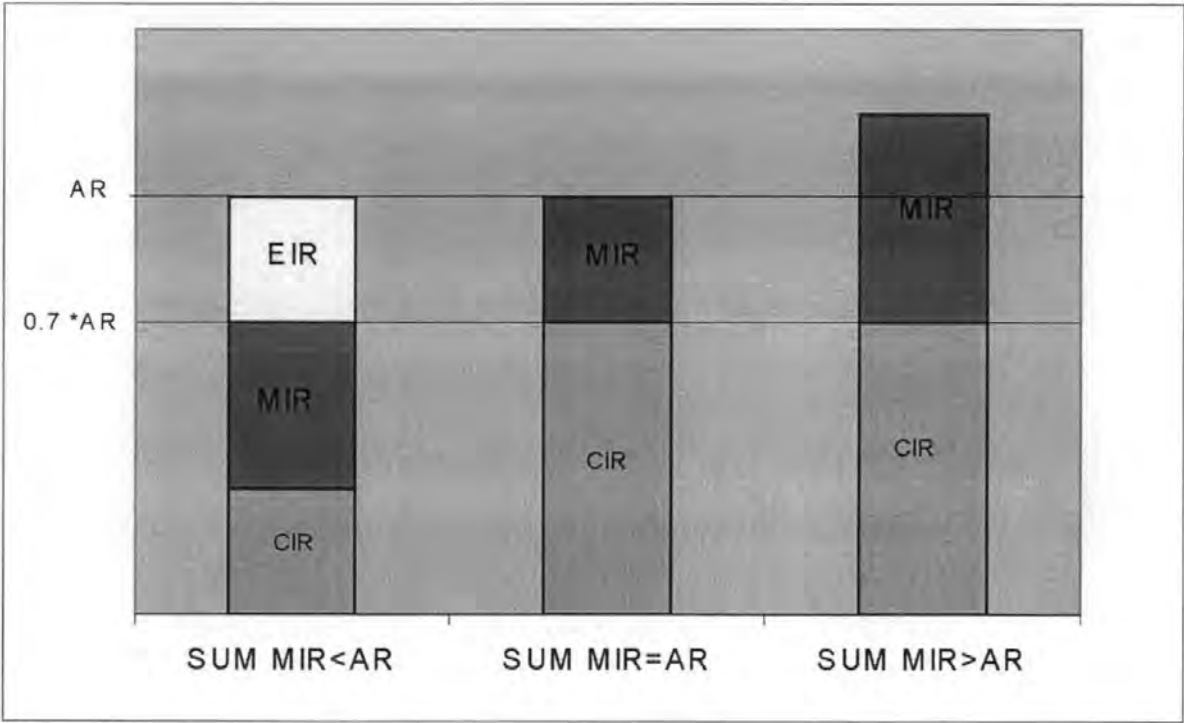


Figure 6.3: Three Strategies for Setting Σ MIR Relative to AR

Type 1 Port: Sum of MIR is smaller than Access Rate (Σ MIR < AR)

In this way there can be some risk for the service provider. If the PVCs get fully utilised the network designers have to tune up the %util to 100 per cent. In this case the guaranteed MIR has to be fully reserved on the trunks for the customer, thus conflicting with the assumption that the utilisation is 70 per cent. When a lot of the PVCs get fully utilised on

one trunk, the total utilisation of the MIR rises above 70 per cent, endangering re-routing abilities. Seen from the current measurements though, it is very unlikely that all PVC's get a utilisation of 100 per cent. So when these PVCs are well spread over the network, there will not be any negative effects of some PVCs having a utilisation close to 100 per cent, as designers can balance them with PVCs with very low utilisation.

An advantage for the customer is that some EIR is free at their port, so that when EIR is available on the trunk, this PVC can certainly use it. This is not dependent on the silence periods of other PVCs on this port. A problem though is that due to relative low MIR these PVC's get a low share in free EIR (see section 6.8.4.6 below for explanation). Another problem is that due to the relative low MIR, the rise-time from MIR to PIR is large. Remember that when capacity is available PM adds 10 per cent of the MIR to the Credit Manager Rate (CMR) every 40 ms, so in short the port characteristics are:

- relative low share of EIR on trunk
- availability of EIR on port not dependent on silence of other PVC's on this port
- high rise-time due to relative low MIR.

Type 2 Port: Sum of MIR is equal to Access Rate ($\Sigma \text{MIR} = \text{AR}$)

This strategy brings the same risk of the possibility that the utilisation has to be set at 100 per cent for some PVCs, although this is very unlikely. Compared to the previous option, a medium share of EIR is expected, due to higher MIR. The availability of EIR is dependent on the silence of other PVCs and cannot be guaranteed if these PVC's do not share the same trunks or if the network is not load balanced. In short the port characteristics are:

- Medium share of EIR on trunk;
- Availability of EIR on port is dependent on silence of other PVCs on this port; and
- Medium rise-time due to relative medium MIR.

Type 3 Port: Sum of MIR is larger than Access Rate ($\Sigma\text{MIR} > \text{AR}$)

The reasons for choosing this strategy can be that the customer expects his PVCs to be highly interactive and bursty, so that the PVCs provide and use each other's MIR as EIR directly at the port. Another advantage to the customer is shorter rise-time from MIR to PIR because of relative high MIR and a relative high share in EIR. The characteristics are:

- the relative high share of EIR on trunk;
- the availability of EIR on port is dependent on Silence of other PVCs; and
- the low rise-time due to relative high MIR.

Remember that the strategy for the trunk was to have $\Sigma\text{MIR} \leq \text{trunk capacity}$. But the ports of type three will never be able to get ΣMIR into the network, so reserving the full ΣMIR on the trunk would be unnecessary if this port would be used as a standard for the network.

A port of type one can get its EIR whenever it is available on a trunk, but it has a low share in it due to the relative low share in MIR. This port does not have a natural guard against sending more than ΣMIR into the network at some time instants; thus it is also conflicting with the assumptions made for the trunk. The second option now fits perfectly to an earlier made assumption for the trunk stating that it must be able to deliver the ΣMIR when it is requested at some time instant. It is assumed though that the average utilisation

of the ports and trunks are lower than 70 per cent, and therefore 70 per cent of the Σ MIR is reserved.

If the sum of the MIRs at a port is equal to or larger than the Access Rate, only EIR can be gained when one or more PVC(s) on this port are silent. The problem is that if the silent PVC(s) and the active PVCs are not using the same trunks (which is likely), in some circumstances it is not guaranteed that this PVC gets its EIR.

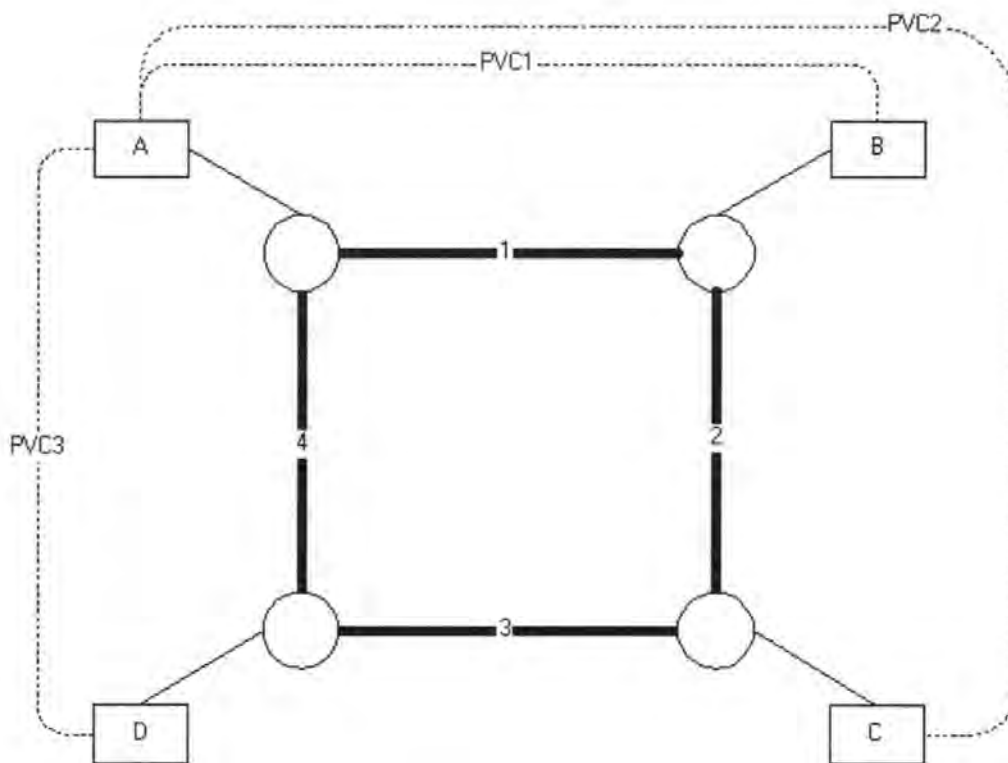


Figure 6.4: Routing problem

An example of a small network is shown in Figure 6.4. Looking at the three PVCs connected at site A. From site A PVC1 is routed to site B via trunk1. PVC2 uses trunk1 and trunk2 to connect to site C, while PVC3 uses trunk4. Now assume that PVC3 is not used very much, thus leaving free EIR at port A. Let us also assume that trunk1 is heavily loaded, so that PVC1 and PVC2 would desperately want the available capacity of PVC3.

But PVC1 and PVC2 cannot use the available capacity as EIR, because they do not use trunk4 where this capacity is available. This calls for a strategy that heavy-use PVCs share trunks with lightly-used PVCs when they are configured on the same port. This strategy will be referred to as load balancing.

6.8.4.5 Overbooking on Time of Day

When overbooking on time of day, network providers sell the same capacity in the backbone to more than one customer, assuming that they will not use the capacity in the same time windows. Notice from chapter 5 that LAN interconnect at night only uses a fraction of the capacity that it uses during daytime.

Overbooking on time of day is interesting, especially when it is realised that the LAN interconnect service is almost unused at night-time. The management system could be run in the afternoon to establish PVCs for the night and run a job again in the morning to remove them. This can only be done for a small amount of PVCs, so this will not become common practice.

Another strategy is to use PVCs with $MIR=2.4$ kbit/s, the smallest MIR possible. For these PVCs it is very hard to raise their CMR above MIR (2.4 kbit/s) during daytime, but at night these PVCs should be able to get full capacity up to Access Rate, as the network is not used very much at night. It should be realised that the normal LAN interconnect users still have access to network-capacity at night, so that EIR cannot be guaranteed to a PVC with a very small MIR, unless the utilisation at night stays as low as it is.

6.8.4.6 EIR Guarantees For LAN Inter-Connection

It is assumed that the tuning is done in the way described in the previous section, meaning that $\Sigma \text{MIR} < 100$ per cent of trunk capacity and that %utilisation is used so that $\Sigma \text{CIR} < 70$ per cent of trunk capacity. Re-routing is still possible, because this is carried out based on CIR, not on MIR. With this strategy some guarantees about the EIR availability can be given.

First it should be noticed that the sum of EIR is zero when the sum of MIR equals the total capacity. If there is EIR available on a trunk due to the fact that not every PVC will use up its MIR continuously during the day, the share of a certain PVC in the total of EIR is dependent on its share in the total of MIR. So when a PVC_i has a MIR_i its share in the available EIR will be at minimum:

$$\text{EIR}_{\min} = \text{MIR}_i / (\Sigma \text{MIR}_i) \quad (1)$$

Formula 1 shows that absolute guarantees about EIR cannot be given, but a relative minimum share can be guaranteed when EIR is available.

It will be only in rare occasions that all the end users are using up their MIRs constantly at the same time. To have an equal distribution of free EIR over the network it will become important that trunks are monitored to see if the utilisation of the MIRs are distributed equally over the network. If not, then unfairness has been created, because the end-users routed over an under-utilised route will have more free EIR than other end-users routed over an over-utilised route.

The above implies that load balancing of the MIRs in the network is necessary to distribute the EIR in a fair way among end-users.

Now if it is assumed that the average utilisation of the Σ MIR on a trunk is less than 70 per cent (the maximum that still allows for re-routing the average MIR when a trunk is down), then the appropriate EIR guarantee can be seen in Figure 6.5.

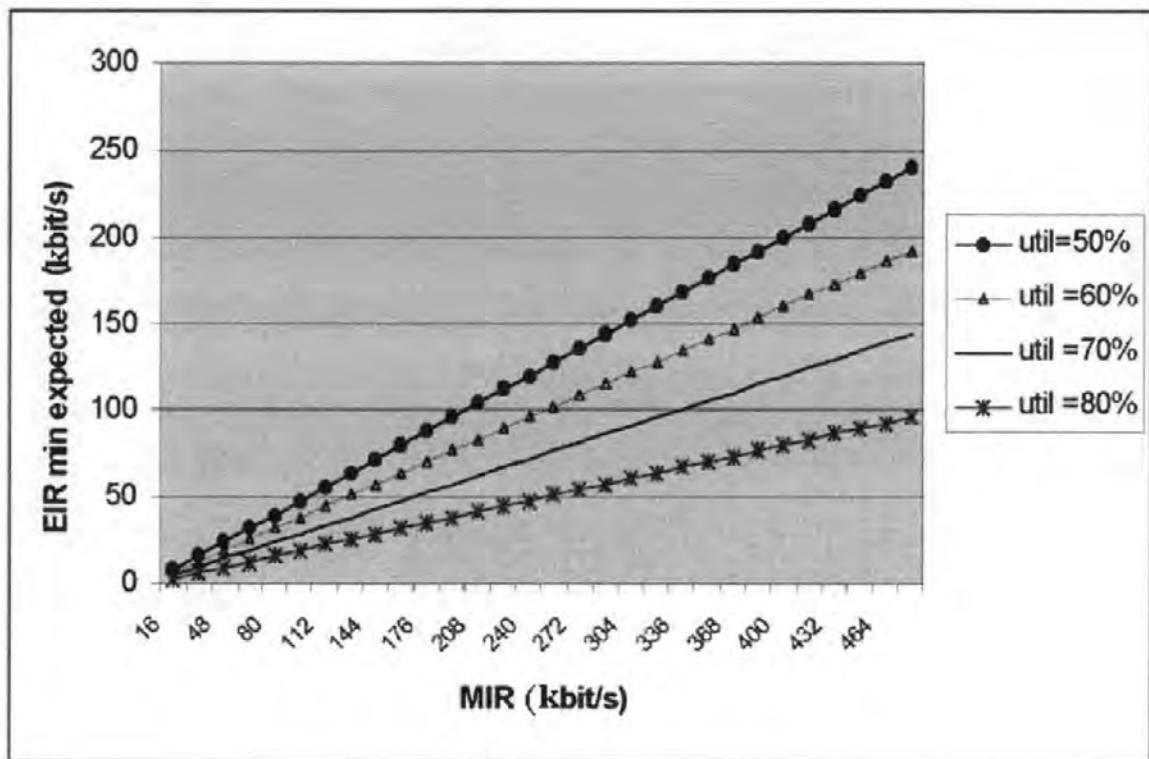


Figure 6.5: Minimum average expected EIR

In Figure 6.5 the configured MIR has been plotted against the expected amount of EIR. Formula 1 was used to compute the relative share in EIR based on the share in MIR. The absolute EIR capacity available was set according to the average utilisation of the MIR on this trunk.

For example, if the capacity of the trunk is 1.6 Mbit/s and the average utilisation of the MIR of the trunk is 70 per cent, this leaves 30 per cent of trunk capacity on average free for EIR, which is 480 kbit/s. The sum of MIR is 1.6 Mbit/s which gives for a certain PVC with a MIR of 200 kbit/s a relative share of 12.5 per cent in EIR according to formula 1. This makes 60 kbit/s of average EIR availability, as can be seen in Figure 6.5.

The average measured utilisation of the MIR of these PVCs is used as a parameter, which can be different than the configured utilisation of 70 per cent. It is expected the instantaneous utilisation of the Σ MIR to be 100 per cent at some time instants, but the utilisation over a longer time windows (one hour) must not to be higher than the configured 70 per cent) because this is a dimensioning rule. This allows one to expect a certain minimum average availability of EIR on large time windows.

Using the dimensioning rules described earlier, then the expected EIR availability for a certain configured MIR over a time scale longer than an hour is at minimum the value of the 70 per cent line.

When looking at the utilisation pattern of Chapter 5 in Figure 5.11 it can also be expected that the average availability changes with time of day. If these PVCs use the trunk capacity for their MIR up to 70 per cent, an indication can be derived for the improvement of the availability of the EIR, under the condition that no trunks are down. The improvement indication factor is plotted in Figure 6.6.

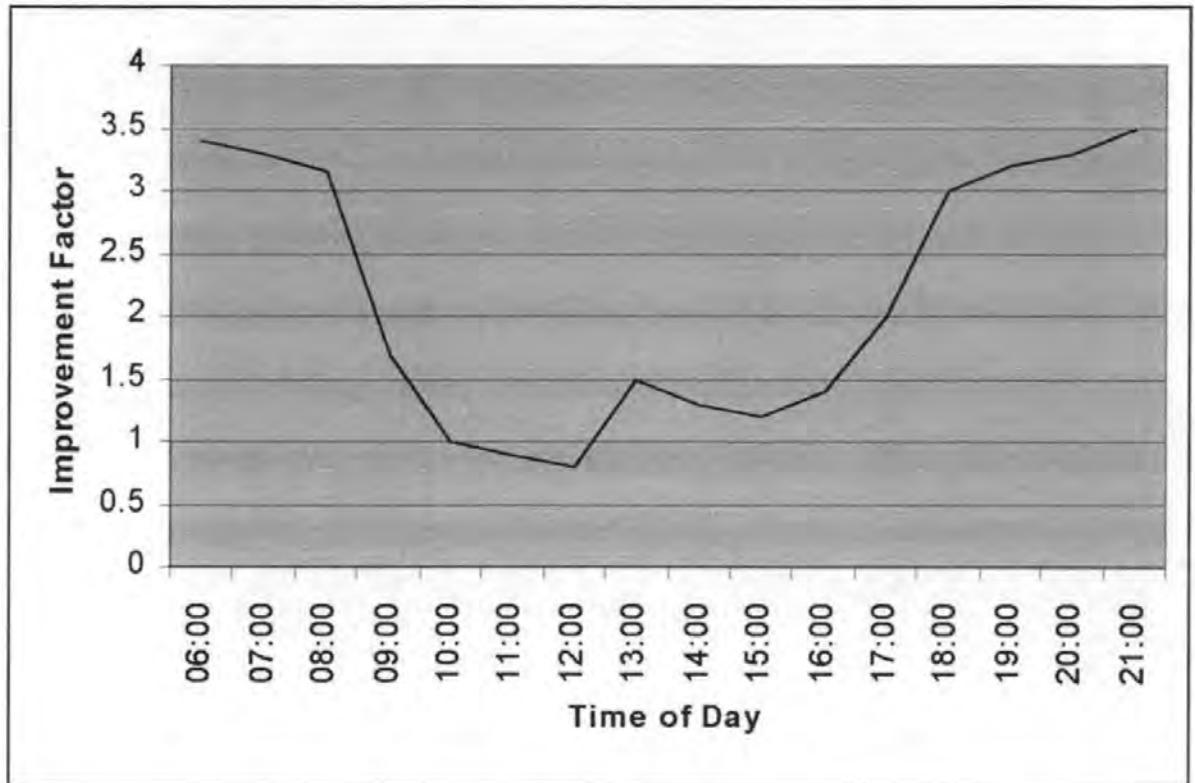


Figure 6.6: An indication for the improvement of EIR availability

6.9 Methodology

Instead of using the general OBF, it is proposed to assess the individual OBF by feeding information from different sources into the model. It should be noted that the method in this thesis is not restricted to the steps shown, and that these are only generic guidelines. Network providers with different needs can adopt the methodology as a generic instrument to modify it for their own purposes. These principles are used in a real situation as follows:

1. Choose the design rule. Here a design rule for the port is chosen as $\Sigma \text{MIR} = \text{Access Rate}$, so that when all PVC's request their MIR they can get it into the network, because the trunks are designed to handle that situation. But as it is expected that the utilisation of the ΣMIR of the trunk to be less than 70 per cent, the average utilisation of the ports

must also be around 70 per cent or less. The re-routing decisions are then based on 70 per cent trunk utilisation.

2. Identifying the busiest day of the week for the whole network in terms of total traffic. It seems that at the middle of the week, or usually between Tuesday and Thursday, the traffic levels are at their highest.
3. The average traffic of the peak hours in terms of total traffic has to be determined. As can be seen on the previous charts, the busy period are between 10:00 and 16:00 and are used for each PVC on Tuesdays, Wednesday or Thursdays.
4. Identify the recording window. These busy day figures are recorded for a rolling window of 13 weeks. The 13 weeks was chosen, as this gives a period where the granularity is small enough for identifying changes and still large enough for the overall trending and it gives a good level of confidence for predictions.
5. These figures then give a trend of the dynamic utilisation for each PVC. For an optimisation of the routes in an existing network, the trended utilisation of the PVC to 2 weeks is fed into the model. Two weeks are chosen to enable the network planning for reaction time, e.g. capacity management.
6. Identify the stage in a life cycle of the PVC. Depending on the stage of the PVC, a premium on the OBF should be added.

7. Classify the importance of the PVC. Regarding the importance of the PVC, a premium should be added to the OBF of the PVC. The importance of the PVC can be dependent on the service, customer, owner etc of the PVC. This parameter can be put in by other departments, as mentioned before.
8. Define the OBF for the PVC. Note that for some of the PVCs the statistics show utilisation levels greater than 100 per cent. This is because they are running over trunks that are not congested. For design and configuration purpose these PVCs are limited to 100 %UTIL as planners want to design the network for a maximum of up to the CIR for each PVC. The reason for setting the PVC to a maximum of 100 %UTIL and not higher, is economic, as any network provider is committed in the contracts to deliver a service set by the CIR. By setting the utilisation at a level higher than 100 per cent the company would be giving capacity away, and therefore losing money. PVCs, which are not used by the customer, or where the dynamic utilisation is low, are set to a minimum level.
9. Evaluate results after a period of time. PVCs, which are permanently loaded at 100 per cent dynamic utilisation or higher, should be upgraded to a larger CIR, as the customer will have a better performance and the profit margin for the network provider will increase. Customer Service and Sales will negotiate revising the CIR with the customer. If QoS parameters are not reached, then the process should be started from step one again to set new OBF for the PVC.

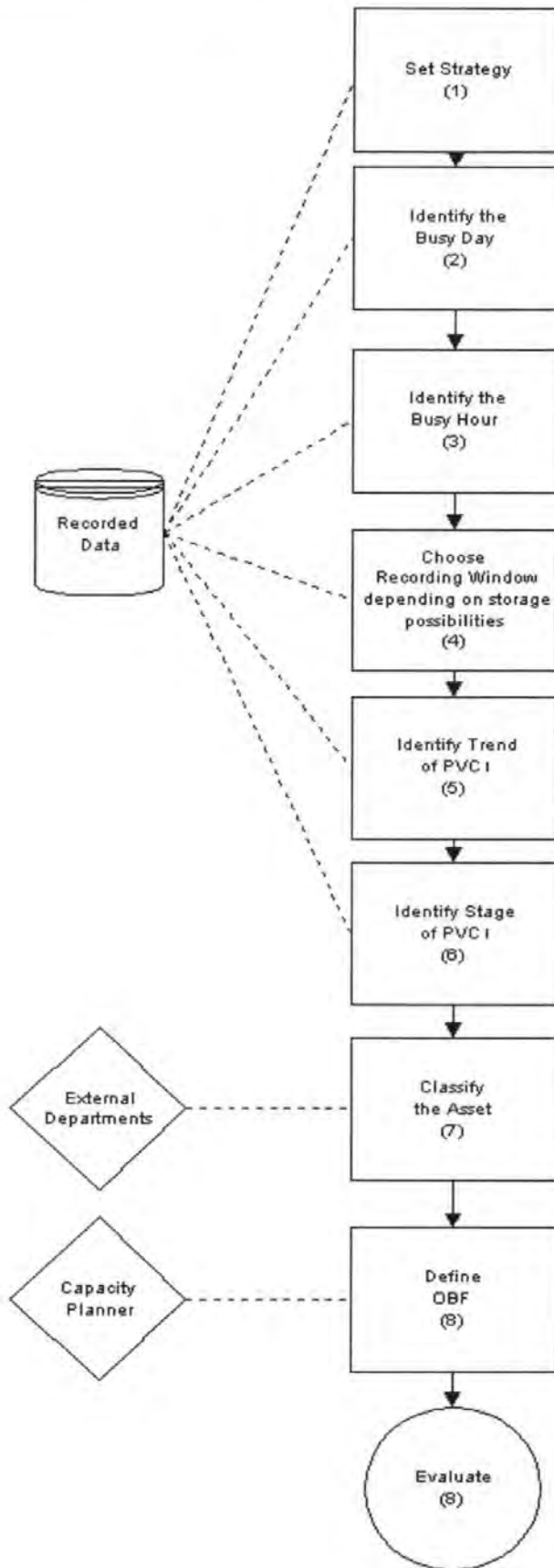


Figure 6.7: Methodology

6.10 Conclusion

A methodology based on traffic profiles and individual PVC traffic behaviour has been proposed. Implementing the methodology delivers a better service and saves potential long-term costs. There is a lot of scope for more investigations into this area, as these ideas can be used and further developed for future technologies. It is important to create traffic libraries and associated statistical models for planning purposes as well as traffic management and billing.

The methodology can be used either for real-time assessment of traffic and network resources or for capacity planning and network design. To use the methodology in real-time a mechanism is needed for the allocation of network resources. One of the possibilities is the emerging Resource Reservation Protocol (RSVP). This is a sophisticated mechanism that specifies its own signalling mechanism for communicating an application's QoS requirements to a router. RSVP has not been widely implemented by application vendors. Although some routers support RSVP, the protocol is not considered mature enough for widespread deployment because of scalability concerns. RSVP imparts a significant processing load on routers and could cause performance degradation.

Implicit QoS is likely to remain more popular than explicit QoS for the foreseeable future. Implicit QoS does not require as much router processing. More important, any explicit QoS technique is a potential management nightmare. Given the chance, end users are likely to configure their software to ask for the best possible service level. Administrators would probably need to establish rules for users and perhaps even configure QoS on a per-user basis. Another possibility is to use application characteristics collected automatically

e.g. robots similar to search engines download WWW pages and save their application signatures for later use.

Chapter 7: New Congestion Control Technique

7.1 Introduction

This chapter addresses the end-to-end traffic management issues in TCP/IP over a proprietary Frame Relay network, which was described in Chapters 2 and 3. The Frame Relay control is applicable to the Frame Relay network, while the TCP flow control extends from end-to-end. The end-to-end performance in terms of TCP throughput and file transfer delay in cases using TCP/IP in the frame Relay network was investigated.

In the beginning of Frame Relay, the general perception tended towards the belief that frames should be quickly discarded in case of congestion to create space on the trunk for other traffic. Meanwhile this idea has changed and various methods emerged covering a wide range of techniques and philosophies, some of them discussed in Chapter 3, ranging from window flow control, slow start, leaky bucket to hop-by-hop credit scheme. Most of the techniques focus on the instant “relief” of congestion, and do not take account of re-transmitting problems and unnecessary delays occurring out of them.

The new idea is to use traffic profiling as a mean to optimise congestion control by predicting arrivals of packets. As Chapter 5 has shown, applications have reoccurring patterns. These patterns have often high probabilities and are used in this Chapter for the development of a new mechanism. Especially the extended packet train information is used in the mechanism for decision making.

To show the effects on a real life network, a proprietary mechanism was chosen. The belief is that the improved mechanism could react more flexibly in times of congestion and therefore improve overall performance and throughput through a more intelligent discarding mechanism. The packet discard strategy, fragmentation of frames, and buffer

management of this mechanism as well as the effects of lost frames were investigated. Using own packet traces and previous work on the packet trains, this model was the foundation for an improved mechanism, which should react in a more flexible way to traffic patterns. To show the re-sending effects of lost frames and the existing problems of this scheme a simulation of the congestion control mechanism was developed.

The effects of frame losses inside the Frame Relay network will be shown to be dramatic at times. Data have to be re-sent from the original source and travel the whole distance again, utilising resources and “wasting” capacity on the route. The main aim is to optimise PM to avoid a re-sending of frames and obtain a fair discarding of frames. If the network experiences congestion, the unavoidable frame losses should be handled in a better manner. It seems that the main reasons for frame drops are unintelligent frames discarding mechanisms related to buffer utilisation levels and their thresholds.

The simulation results show that the new control mechanism performs comparably to the existing congestion control mechanism and even outperforms it. It will be shown that it is not sufficient to have a loss-less Frame Relay network from the end-to-end performance point of view, but also the consequences of pushing the traffic to the edges. The results illustrate the necessity to have a congestion handling mechanism that can couple the PM with TCP feedback control loops. A mechanism is proposed that makes use of the PM feedback information and the edge-device congestion state to make packet dropping decisions at the entry point of the network. Using packet train information at the access point and carrying this information within the Fixedpackets algorithm, the end-to-end performance in throughput and delay are improved while using Frame Relay network technology.

7.2 Existing Models

To show the improvements over existing systems the following two models are shown: In the first case TCP/IP over Frame Relay is considered, and in the second model the interaction between TCP/IP and the existing proprietary mechanism (PM) is shown. This leads then to the new mechanism and its approach.

In a TCP/IP over Frame Relay network environment, a Frame Relay network typically interconnects LANs. The focus is on the end-to-end traffic management issues in TCP/IP over Frame Relay network. The end-to-end performance in terms of TCP throughput and file transfer delay in the Frame Relay network has been investigated. This chapter discusses the issue of trade-off between the buffer requirement at the Frame Relay edge device (e.g., Ethernet-Frame-Relay switch, Frame Relay router interface) versus Frame Relay switches inside the Frame Relay network with a proprietary congestion control mechanism. In such cases the proprietary flow control may simply push the congestion to the access devices of Frame Relay network. Even if the Frame Relay network is kept free of congestion by using PM, the end-to-end performance perceived by the application may not necessarily be better. Since many of today's data applications use TCP flow control protocol, the benefits of proprietary flow control as a network technology can be questioned, arguing that TCP/IP is equally effective and much less complex. The problem, however, lies in the accounting mechanisms and fairness of the TCP/IP when used over Frame Relay without guarding it by other mechanisms.

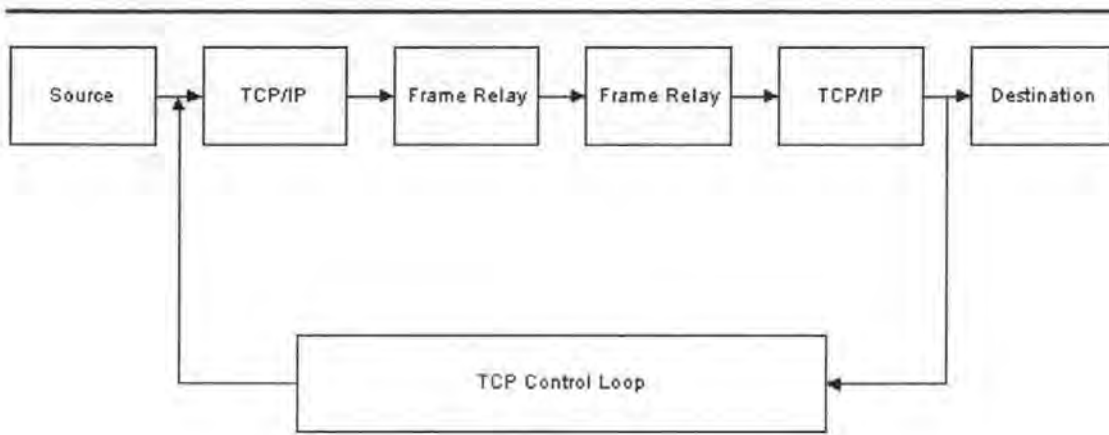


Figure 7.1 Frame Relay Network

In these cases, source and destination are interconnected through TCP/IP over Frame Relay networks. In Figure 7.1 the connection uses no congestion-control and in Figure 7.2 the connection uses PM. In the first case, when congestion occurs in Frame Relay networks, Fixedpackets are dropped, resulting in reduction of the TCP congestion window. In the second case, when congestion is detected in Frame Relay Networks, PM becomes effective and forces the Frame Relay port to reduce its transmission rate into the Frame Relay network. If congestion persists, the buffer in the edge device will reach its capacity and start to drop packets, resulting in reduction of the TCP congestion window.

From the performance point of view, the latter involves two control loops: PM and TCP. There are two feedback control protocols, and the interactions or interference between the two may actually degrade the TCP performance.

In the context of feedback congestion control, buffers are used to absorb the transient traffic conditions and steady-state rate oscillations due to the feedback delay. The buffers help to avoid losses in the network and to improve link utilisation as the aggregated traffic rate is below the link capacity. When there is congestion the switch conveys the

information back to the source. The source reduces its rate according to the feedback information. The switch sees a drop in its input rates after a round trip delay. If the buffer size is large enough to store the extra data during this delay, it is possible to avoid losses. When the congestion abates, the switch sends information back to the source allowing it to increase its rate. There is again a delay before the switch sees an increase in its input rates.

Meanwhile the data are drained from the buffer at a higher rate compared with its input rate. The switch continues to see a full link utilisation as long as there is data in the buffer. Since the maximum drain rate is the capacity of the link and the minimum feedback delay is the round trip propagation delay, it is generally believed that a buffer size of one propagation delay product is required to achieve good throughput.

The above argument assumes that the source can reduce its rate when it receives feedback information from the network and that the source will be able to send data at a higher rate when congestion abates in the network. The problem lies in the reaction of the mechanism, before the dropping occurs. In the current system the edge device is not relaying any congestion information to the network. It controls its transmission rate according to the PM feedback information, whereas TCP/IP sources send data packets according to the TCP/IP window control protocol.

TCP/IP uses implicit negative feedback information (i.e., packet loss and time out). When there is congestion in the network, TCP may continue to transmit a full window amount of data, causing packet loss if the buffer size is not greater than the window size. In case of multiple TCP streams sharing the same buffer, it is impossible to size the buffer to be greater than the sum of the window size of all the streams. When there is packet loss, the

TCP window size is reduced multiplicatively and it goes through a recovery phase. In this phase, TCP is constrained to transmit at a lower rate even if the network is free of congestion, causing a decrease in the link utilisation.

When PM is used, it is possible to avoid losses within the Frame Relay network. However, since the host uses TCP/IP, the congestion is merely shifted from the Frame Relay network to the TCP/IP/Frame Relay interface. There may be packet loss in the transmit-port queue and loss of throughput because of that. Moreover, there is a possibility that there will be negative interaction between the two feedback loops. For example, when the TCP/IP window is large, the available rate may be reduced and when the TCP window is small, due to multiple packet losses, the available capacity may be high. When edge buffers are limited, this kind of negative interaction can cause a severe degradation in the throughput. When no closed-loop mechanism is used, packet losses due to buffer overflow or Fixedpacket discard at the Frame Relay switches trigger the reduction of TCP/IP window. In this case, there is only a single TCP feedback loop and no additional buffer requirement at the edge device. This, however, has the disadvantage of uncontrolled capacity sharing and creates unfairness, as discussed in Chapter 3.

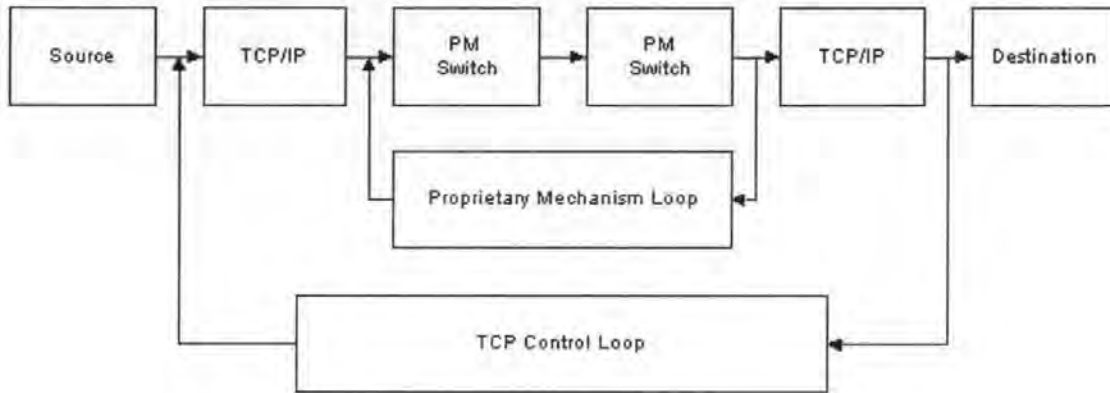


Figure 7.2: PM and TCP/IP Control Loop

The performance metrics considered are basically the goodput at the TCP layer and the delay. The goodput is defined as the rate at which the data are successfully transmitted up to the application layer from the TCP layer. Retransmitted data are counted only once when they get transmitted to the application layer. Packets arriving out of sequence have to wait before all the preceding data are received before they can be transmitted to the application layer.

7.3 New Approach

One phenomenon observed in the traffic traces was that of packet size distribution. In Chapter 5 application signatures were described, which have shown that successive packets have a high probability of the same size. These successive packets also have a tendency to belong to the same train, which can be used for the improvement of congestion control mechanisms.

In the network traffic monitored, it was found that the PC sources either sent or received packets mostly in sessions. This means, that if a source initiated a file transfer, then it

usually did not communicate with other sources at the same time. The traffic therefore is directed towards one destination, if a relative short period of time (within a few seconds) is considered. Most analyses on network modelling assume packets coming from all sources on the network. The probability that two packets will then come from the same source would equal $1/m$, where m is the number of nodes. The theory would indicate that the network modelling assumes a uniform probability of a packet coming from any source on the network. The packets would therefore be equally distributed over a period of time. It also would imply that the packet sizes would be distributed in the same manner. However, if the traffic traces are sorted by packets with the same source and destination, the new findings show that the probability that a packet is followed by another packet with the same size is between 35 and 45 per cent. More than one-third of the packets following a packet from the same source to destination have a similar size. In many cases this number is much higher.

This phenomenon shows that the trains from different source-destination pairs usually do not severely overlap on a LAN, as discussed in Chapter 5 regarding packet train. If the overlap is not too strong, then a packet train can be recognised. Of course, the amount of train overlap depends on the total load on the network. During periods of high utilisation one would expect a higher overlap. During periods of low utilisation, there is lower overlap and a higher probability of packet trains than during very high loads. However, after entering the switch nodes into the WAN, the train characteristics can be lost, as the policing algorithms mix traffic from different queues.

The observations in Chapter 5 have shown that successive packets have a tendency to packet clustering. Given a packet, we can predict with high probability that the next packet

will be the same size as the previous packet. Normally, a network has several thousand nodes. Finding the link on which to forward a packet requires sophisticated table search and hashing procedures at each node. The existence of this characteristic indicates that an improvement in the congestion control mechanism can be obtained by a modified “entry” mechanism of packets in the network. The selective discard can be made by the predictability of further arrival of packets, regulating the impact on performance. Thus, there is a higher probability of having the mechanism deciding on packet discards even before they enter the WAN.

7.3.1 Packet Train Profiler

To use this effect for congestion control mechanisms, a probability table or function for an application has to be used. This mechanism can be seen in Figure 7.3. The Packet Train Profiler (PTP) can be used in conjunction with the information received out of the network about current congestion situations. The PTP is updated every time a new Frame comes into the port. Depending on the size and the time of the Frame entered, and the type of application the frame belongs to, a prediction is made about the arrival of a new Frame. This prediction can then be used inside the network, or at the entry point, to decide how to proceed with tagging, discarding or accepting Frames. If the port does not receive a Frame inside a specified time (this time can be set depending on the technical specifications in the environment or the inter-train time of the packet trains), the PTP is reset and the prediction has to restart when the next Frame arrives.

The efficiency of this mechanism depends upon the correct selection of time outs, which indicate the delay between the packet trains. This time out could also be stored in the packet train profiles so that the new PTP could be set up individually by every packet train

arrival. For example, if the inter-train gap time for application A was chosen to be 10 ms, the PTP would reinitialise if a second Frame is not received within this time, or if the frame size is significantly larger or smaller than the previous one.

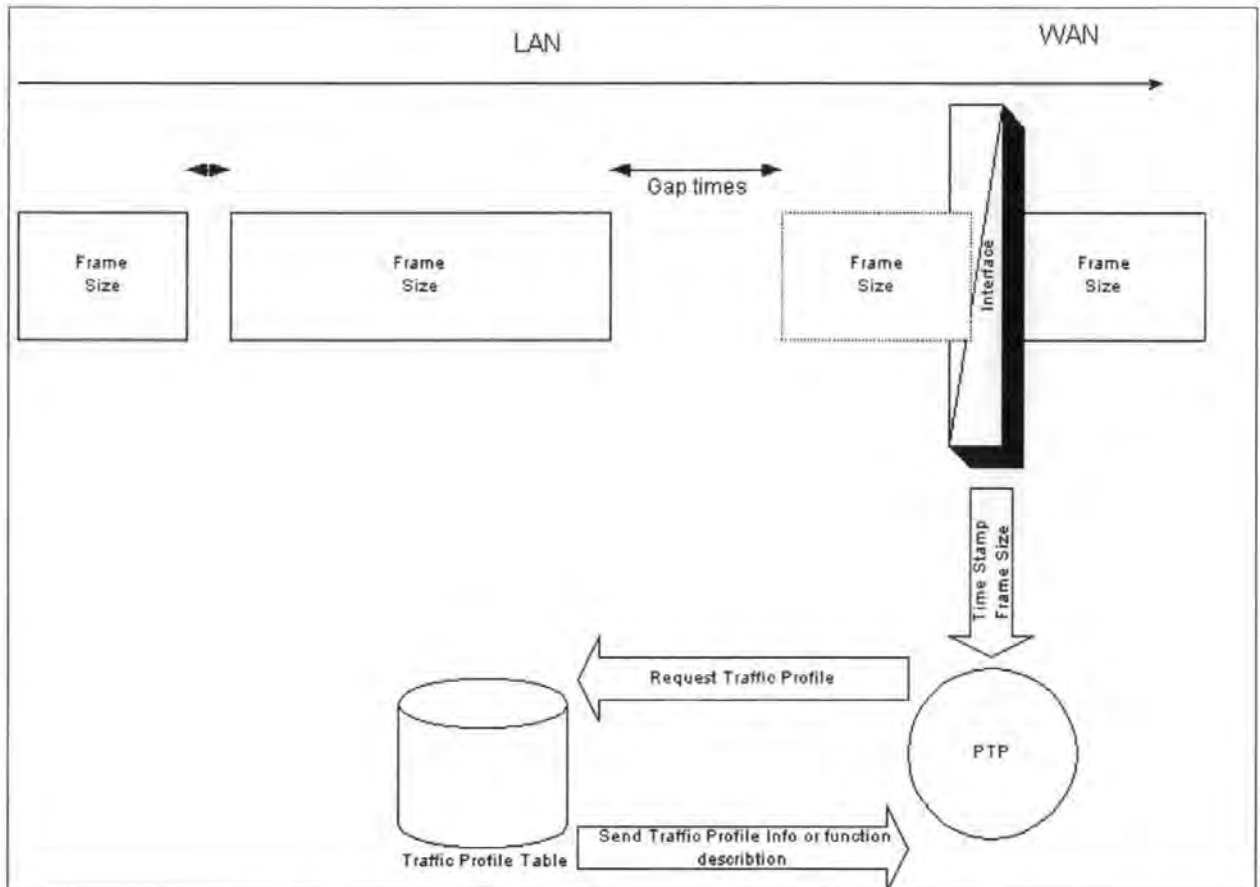


Figure 7.3: Information Flow from Frame traces to PTP

7.3.2 New Mechanism

In Figure 7.4 the new mechanism that indirectly couples PM flow control with TCP window flow control has the objectives of relieving congestion as well as conveying congestion information to the TCP source at the appropriate time. In addition, the algorithm needs to take into consideration how the TCP source can recover from the lost packet without a significant loss of throughput. This is done by combining PTP and the new mechanism. To show the validity of the approach, it is then incorporated with the PM.

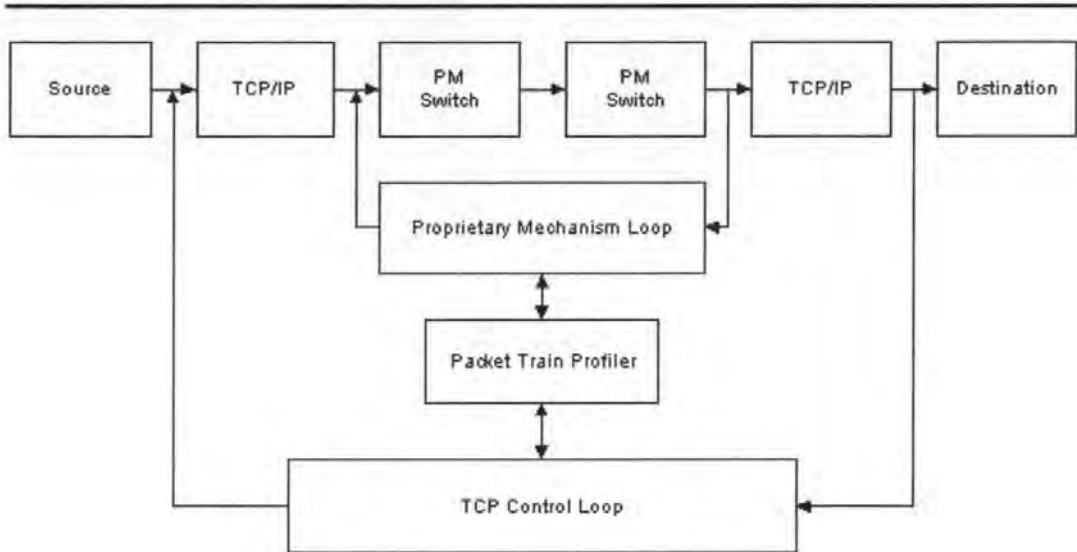


Figure 7.4: Two Flow Controls connected with Packet Train Profiler

Figure 7.5 gives a state-diagram for the algorithm and an explanation of the algorithm follows. The mechanism takes account of the packet train characteristics of an application, deciding which frames should be marked or discarded before entering the network. To enable this additional mechanism, the PTP has to be incorporated with the Credit Manager.

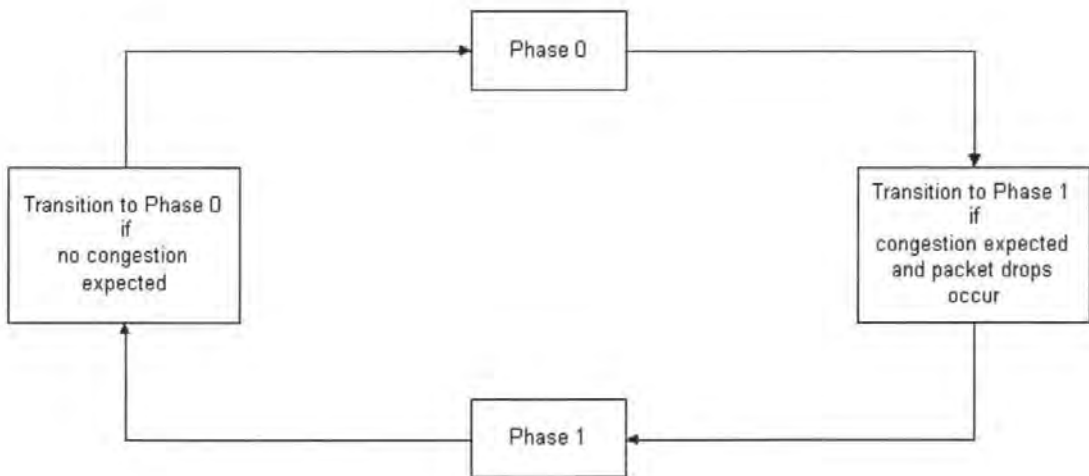


Figure 7.5: Algorithm State Diagram

Variables:

- CMR = Current Available Fixedpacket Rate of the Credit Manager
- CI = Congestion indication in network from receiving RM Fixedpacket.
- QT = Queue Threshold
- PA = Stores the value of the number of packets were anticipated by PTP when a packet is dropped and is then decrement for every packet that is removed from the queue.
- CP = Congestion Phase 1 bit information on the phase of the congestion (0 or 1)

Description: On receipt of the CI Fixedpackets with the feedback information, the CMR value is updated, depending on the PTP and the information received out of the network for the particular PVC. In order to decide on further action, the above algorithm uses this feedback information out of the switch in the network and the anticipated packet arrival derived from the packet train. There are four possibilities:

- a Fixedpacket should be dropped from the queue;
- the rate of inflow Fixedpackets should be reduced;
- the rate of inflow Fixedpackets should remain; or
- the rate of inflow of Fixedpackets should be increased.

This coupling is a critical factor in resolving the issues discussed in earlier sections. Every time a packet is dropped in the network, the algorithm checks with the PTP if more

congestion is expected. The algorithm consists of two phases. The criteria used to affect a packet drop are different in the two phases. When CMR decides that it is reducing the network inflow, this information should only be conveyed to the TCP source if strong congestion is anticipated. If a packet is dropped, the TCP source will automatically reduce its rate, and CMR should not always reduce the network inflow after a packet drop. This enables the new mechanism to be more flexible than the old one, as it is not overreacting in the case of packet drops.

Phase 0: The network is assumed not to be congested in this phase. This is shown through the CMR, which is slowly changing, and the queue length is less than the threshold. There are also no down messages from any switch, which would suggest congestion further in the network. Two possible scenarios can cause a packet drop. If the CMR is constant or slowly changing, the TCP/IP window and hence the input rate to the queue will eventually become large enough to cause the queue length to touch the threshold level. Then it is required to drop a packet to trigger a reduction in the TCP window. In the second scenario, a packet is dropped when there is a drastic increase of congestion anticipated by the PTP and if the queue length is greater than a queue threshold.

These two values combined signify congestion in the network and the algorithm sends an early warning to the TCP/IP source by dropping a packet at the source. The threshold should be set to at least a few packets to ensure the transmission of duplicated acknowledgements, and should not be set too close to the size of queue to ensure the function of congestion avoidance.

In both the above cases the packet is dropped from the queue. This results in early triggering of the congestion control mechanism in TCP. Additionally, the TCP window flow control has the property that the start of the sliding window aligns itself to the dropped packet and stops there till that packet is successfully retransmitted. This reduces the amount of data that the TCP source can pump into a congested network.

Transition to Phase 1: When TCP detects a lost packet, depending on the implementation, it either reduces its window size to one packet, or it reduces the window size to half its current window size. When multiple packets are lost within the same TCP window, different TCP implementations will recover differently.

The first packet that is dropped causes the reduction in the TCP window and hence its average rate. The multiple packet losses within the same TCP window can cause a degradation of throughput and this is not desirable. It is therefore important for TCP/IP that some packets are forwarded from the TCP window even if they are “dead packets”. But a selective drop strategy should allow the recovery algorithm of TCP/IP to recover without the “time out counter” of TCP and therefore reduce throughput degradation. After the first packet is dropped the algorithm makes a transition to Phase 1 and does not drop any packets to cause rate reduction in this phase.

Phase 1: In this phase the algorithm does not drop packets to convey a reduction of the CMR rate. The packets are dropped only when the queue reaches the threshold value at the access point, strong congestion is anticipated and the BD-B queue reaches the threshold and the CP parameter is 0. The packets are then dropped because of the same reasons as described in Phase 0. The difference from Phase 0 is that now a new packet train is

anticipated. When a packet is dropped, the algorithm records the number of packets anticipated in the variable PA. The TCP window size is at least as large as PA when a packet is dropped. Thus, the algorithm tries not to drop any more packets due to rate reduction till PA packets are serviced. In the meantime, however, other PVCs, which are in the Phase 0, will be able to drop packets and therefore reduce congestion on the trunk or switch.

Transition to Phase 0: If the CMR stays at the value that caused the transition to Phase 1, the queue length will decrease after one round trip time and the algorithm can transit to Phase 0. If the CMR decreases further, the algorithm eventually drops another packet when the queue length reaches the queue threshold again, but it does not transit back to Phase 0. The transition to Phase 0 takes place when at least PA packets have been serviced and no strong congestion is experienced.

7.4 Assumptions for the Simulation

In the end-to-end model, the whole network can be considered to be a black box with TCP sources and destinations in the periphery. The only properties of the black box that the TCP modules are sensitive to are the packet loss and the round trip delay through the black box. The round trip delay determines how fast the TCP window can open up to utilise available capacity in the network and also how fast it can react to impending congestion in the network. The packet loss triggers the TCP congestion avoidance and recovery scheme.

The feedback control loop of PM may cause an increase in the over-all round trip delay as it constricts the rate available to each TCP stream. It is generally believed that PM is able to reduce the congestion level within the Frame Relay network. However, it can do so at

the expense of increasing the congestion level at the edge of the network, mainly in the router. From an end-to-end performance point of view it is not clear if this is indeed beneficial. The simulation experiments were performed using a TCP/IP over Frame Relay protocol stack

When cell networks are used to carry application data units that are many cells long, a single lost cell can result in retransmission of the entire application level frame, placing an additional load on the network and magnifying the congestion that led to the cell loss in the first place. Several techniques have been proposed to maintain the integrity of application data units. Early packet discard is one such technique.

Early packet discard is implemented in the output port processor of a switch. Early packet discard keeps track of the passage of frames on selected virtual circuits and if a new frame begins when the occupancy of the link buffer is above a threshold value, it discards the frame. It can be shown that if the buffer size equals the sum of the frame sizes for all the active virtual circuits, early packet discard can achieve 100 per cent utilisation of the output link. However, this does not mean that it achieves the desired goodput. The basic argument, by which this is shown, is as follows. The level of a link queue rising above the discard threshold, and continuing to rise until the beginning of one or more new frames, allows cells to be discarded before entering the buffer. Then the buffer level drops.

PTP can be seen as an extension or replacement to the basic early packet discard technique that seeks to obtain a high utilisation during overload. The idea is to use not only the level of the buffer, but also the probability of arriving frames when making decisions about whether to accept or discard a given frame. The traffic traces play a significant role here.

As Chapter 5 has shown, every application has a unique signature. These signatures can be used for the selection of the packets by the discard mechanism.

To implement PTP, the output port processor of a switch must be modified so that it can track the frames and make decisions regarding frame discarding based on the buffer levels inside the network and the switches and the application signatures available.

7.5 Simulation of PVCs

This part describes the derivation of a baseline for dimensioning a PVC for LAN interconnect services. It is assumed that the backbone is transparent to the end user, therefore a look outside the backbone is necessary to investigate the effects of using TCP/IP. In Chapters 3 and 4 the basics were explained regarding the most used protocol in the backbone. A target of this section is to show the effects in the PVC of increasing the maximum congestion window of TCP/IP on throughput and delay. In this section some initial results are provided to give a general understanding about the performance of the systems. These issues will be discussed in more detail in the later sections, and will address buffer allocations and port utilisation.

7.5.1 Effects of Congestion Window on Throughput and Delay

To study the effects of changing the minimum TCP congestion window, a set-up on throughput and delay has been composed. There is just one PVC configured on a trunk with the following settings:

Trunk: 96 kbit/s

PVC: MIR=48kbit/s; PIR=96kbit/s;

Port: AR.= 96kbit/s

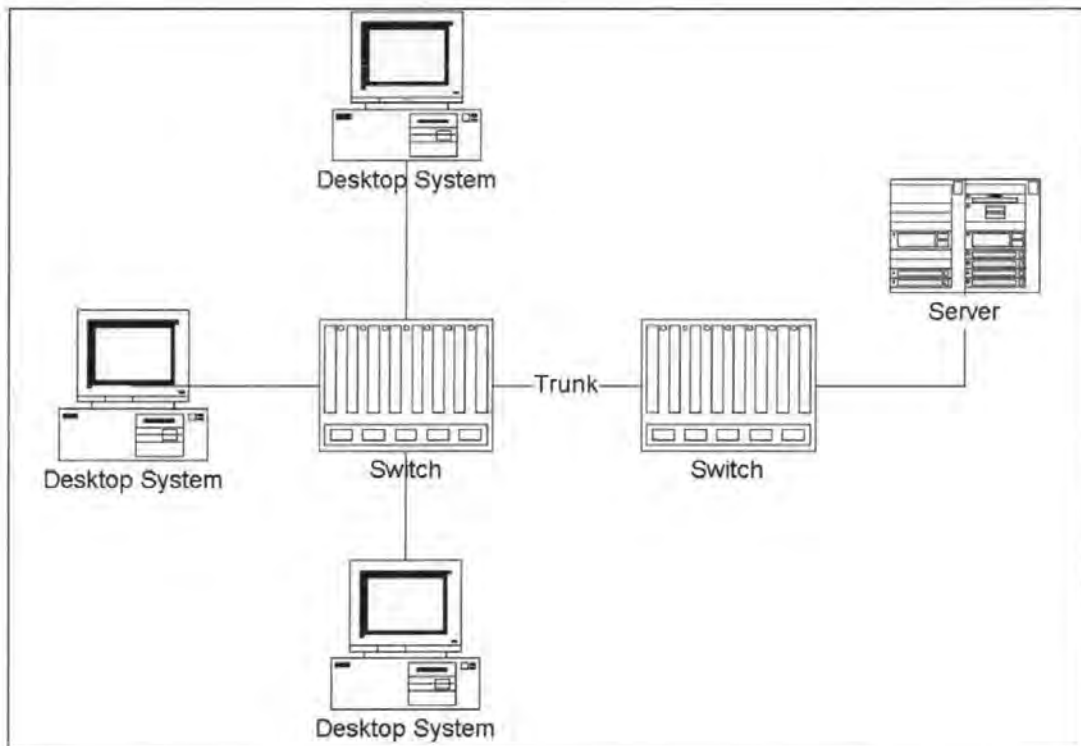


Figure 7.6: Layout for Simulation of PVC

The principal set-up can be seen in Figure 7.6. This picture has been simplified from the real simulation for easier understanding. The standard frame size of a Frame Relay frame in the simulations is 1,488 byte. The end-station generates 1,500 byte LAN-packets of which the 18 byte MAC-overhead is stripped in the router and a 6 byte Frame Relay header is added, which makes the 1,488 byte. The acknowledgement size is chosen at 50-byte size. The set-up implies that the PVC is a channel with capacity AR which can burst up to a level of 96 kbit/s. Buffering can be expected inside the network, and the input rate equals the output rate.

Three end-stations which communicate with one server have been used to simulate the effect that more than one end station is sending data into the network. The throughput and delay of this system have been analysed for the above-mentioned setting and for an

increasing maximum congestion window (1, 2, 4, 8, 16 and 32). Some results are plotted in Figure 7.7.

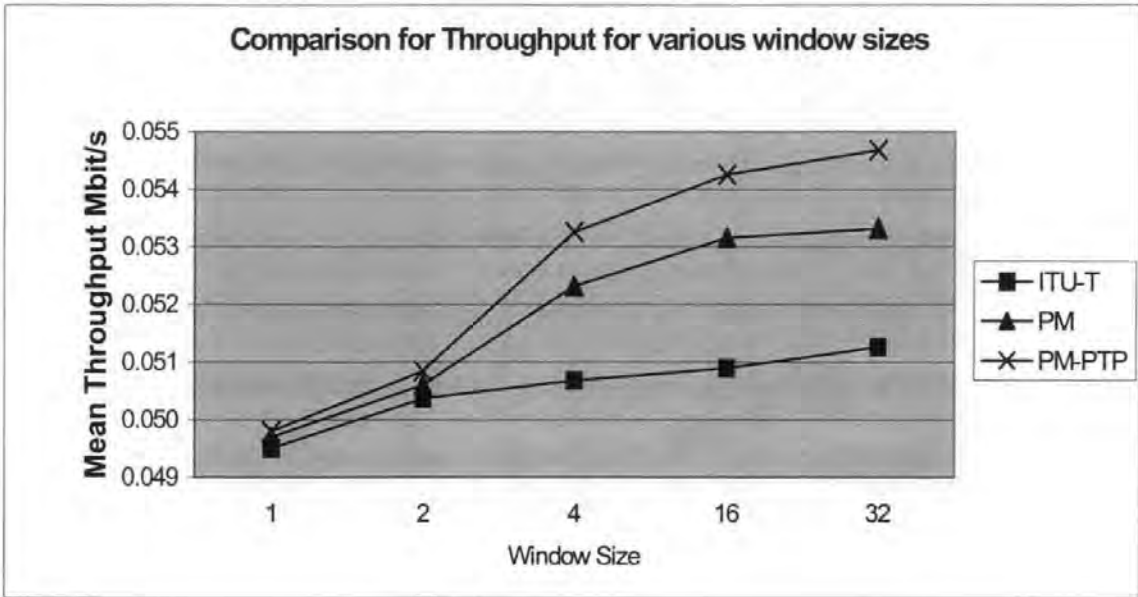


Figure 7.7: Throughput on Increased Window Size

It can be seen that the throughput for all three mechanisms is equal for small windows. This effect can be explained with the following reasoning. For the maximum congestion window set at one, each end station can send one TCP/IP frame per end-to-end Round Trip Delay (RTD) into the network. This delay is mainly made up of the serialisation delays of the Frame Relay frames. The Frame Relay frames are clocked in twice in the sending direction, while the same holds for the acknowledgements of these frames. The expected RTD due to the serialisation are:

$$RTD = 2 * FR_{size} * 8 / B_{channel} + 2 * Ack_{size} * 8 / B_{channel} \quad [s]$$

where FR_{size} is the size of the Frame Relay frame,

Ack_{size} is the size of the acknowledgement, and

$B_{channel}$ is the capacity of the channel

When the channel capacities are known, it is then possible to compute the expected delays due to the serialisation of frames on the access lines. As a maximum congestion window size equal to one is taken, the maximum amount of frames that can be clocked into the network per RTD is three, one for each workstation. This gives an estimation of the capacity consumption by the three workstations.

If the other way around is reasoned by asking what the maximum RTD must be to fill the channel with the given capacity, the calculations are as follows. The three end-stations can each send one frame of 1,488 byte into the network. It takes exactly RTD critical amount of time to transport these frames, which can be computed by dividing the amount of data sent (three frames) by the channel capacity. In this way the channel will always be filled.

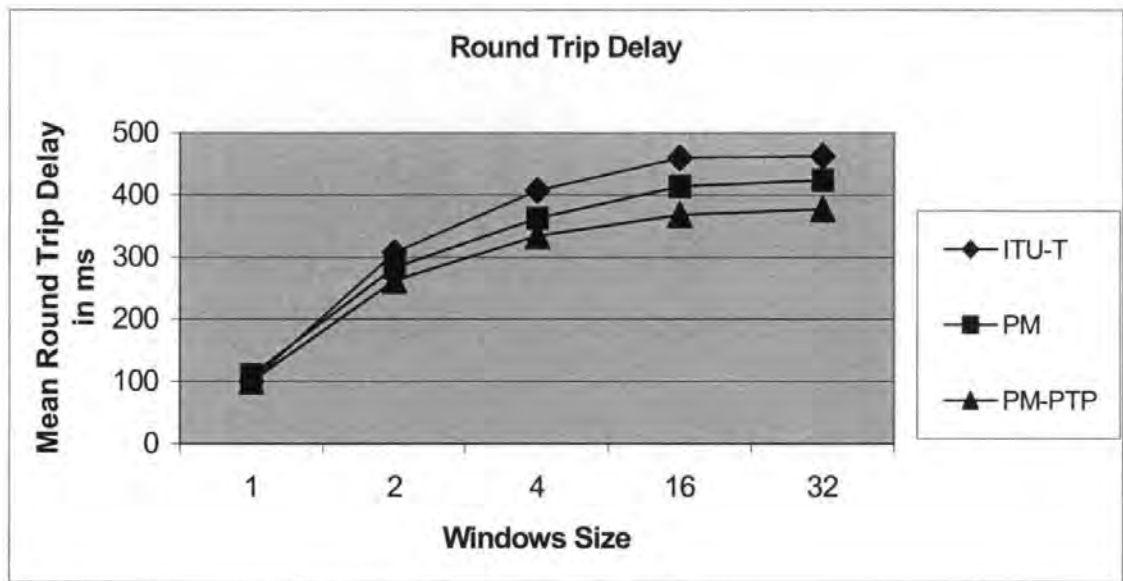


Figure 7.8: Mean Round Trip on Increased Window Size

In Figure 7.8 it can be seen that for the maximum congestion window set at one, the figures for the RTD were very close to the estimate of the inclock delays, meaning that for window 1 the inclocking delays dominate the total Round Trip Delay. Other components of the RTD like LAN delay, router buffer and processing delay, switch buffer and processing delays and the propagation delays of the lines and trunks are less significant.

The results show that when the maximum congestion window increases, the RTD increases fast. This is due to the buffering in the ingress switch, which is situated just before the bottleneck (the channel). When the results for goodput are compared it can be concluded that for this configuration, the increase of the maximum congestion window does not improve the throughput, but makes the results worse.

The short-term queue demand on the gateway is increasing exponentially and opening a window of size W packets, which will require buffer capacity at the bottleneck. If a simple check is done, a maximum congestion window of 8 packets will be taken. This would mean that the total window for three workstations is $3 \times 8 \times 24$ so that in the router queue there would be about 12 packets, which is around 143 kbit when the channel capacity is 64 kbit/s.

To demonstrate the throughput, delay and Fixedpacket loss, some simulation runs have been conducted. The above mentioned set-up for the network was chosen. Also various applications with different response times have been simulated. In the first simulation a response for the application was randomly generated which was between 20ms –30ms. This would represent a typical personal computer with various clocking speed and memory. A typical set of results are shown in Table 7.1 and Appendix C, Section C.1. The

code for the simulation is given in Appendix A and some examples of screen displays during simulations are shown in Appendix B.

Simulation run 1	ITU-T	PM	PM – PTP
Delay in ms	240	244	232
Goodput in kbit/s	66.09	72.62	80.22
Loss at access point in %	8.96	8.99	7.32
Simulation run 2	ITU-T	PM	PM – PTP
Delay in ms	236	232	221
Goodput in kbit/s	66.49	73.86	80.29
Loss at access point in %	9.22	9.39	7.55
Simulation run 3	ITU-T	PM	PM – PTP
Delay in ms	227	257	233
Goodput in kbit/s	67.02	73.56	80.85
Loss at access point in %	9.22	9.40	7.55

Table 7.1: Performance Metrics for Frame Relay, PM and PTP

The set of results with the PM algorithm shows an overall improvement in goodput and loss of packets. The lost packets inside the network are detected with the receipt of duplicate acknowledgements, but the TCP instead of reducing its window size to one packet, merely reduces it to half its current size. This increases the total amount of data that are transmitted by TCP. In case of multiple packet losses the TCP sources are silent for long periods of time when TCP recovers a packet loss through time-out. The increase in throughput in the case of TCP with PM can be mainly attributed to higher amounts of data being transmitted by the TCP source. However, that also increases the number of packets that are dropped uncontrolled at the access points causing an increase in the average file transfer delay.

In the findings it can be seen that PM results in better performance than without proprietary congestion management. While the PM scheme results in less Fixedpacket loss

within the Frame Relay network, it also translates to better end-to-end performance for the TCP connections. However, in some cases the end-to-end performance results in higher frame losses at the access point of the network, due to increased retransmissions and poor interaction between the TCP/IP window and the Credit Manager Rate (CMR). As the input buffers are large, the access port is large enough to hold most of the TCP window in ITU-T Frame Relay. However, a low throughput still exists because of frame losses within the switch. PM, on the other hand, “transfers” the congestion into the edge device, which can handle it with larger buffers. A large buffer could easily reduce frame losses at the access. This does not solve the problem though, since the TCP window dynamics is actually triggered by loss of packets. With PTP the results are more positive, as the goodput is increased and frame losses at the access points of the network are reduced. The delay has also improved slightly. Overall it could be said, that the improvement increases the performance of all PVC connections, as this has a knock-on effect, in reducing congestion and increasing capacity availability for other PVCs. There are better interactions in PTP between the TCP flow control and the Frame Relay flow control mechanism. The only factor that influences the TCP flow control, in the absence of lost packets, is the total round trip time - which in this case slowly increases as the buffers fill up thereby increasing end-to-end delays. With PTP the buffers are better controlled, not only in the access points but also in the network itself.

It is evident from the results that it is insufficient to keep the Frame Relay network free of congestion to provide good end-to-end throughput and delay performance. The approach to solve this problem is to couple the two flow control loops and indicate a problem before a Fixedpacket is lost inside the network. The information received from the network,

along with the local queue sizes, is used to detect congestion early and speed up the TCP slow-down and recovery processes.

7.6 Simulation for Congestion at the Trunk

Firstly, the behaviour of the BD-B queue for bursty traffic characteristics is investigated. Secondly, an investigation will be carried out on whether the trunk performance can be improved by changing the BD-B queue thresholds. All simulations have been designed as in Figure 7.9. The configurations are displayed in the various sections, where the results are discussed.

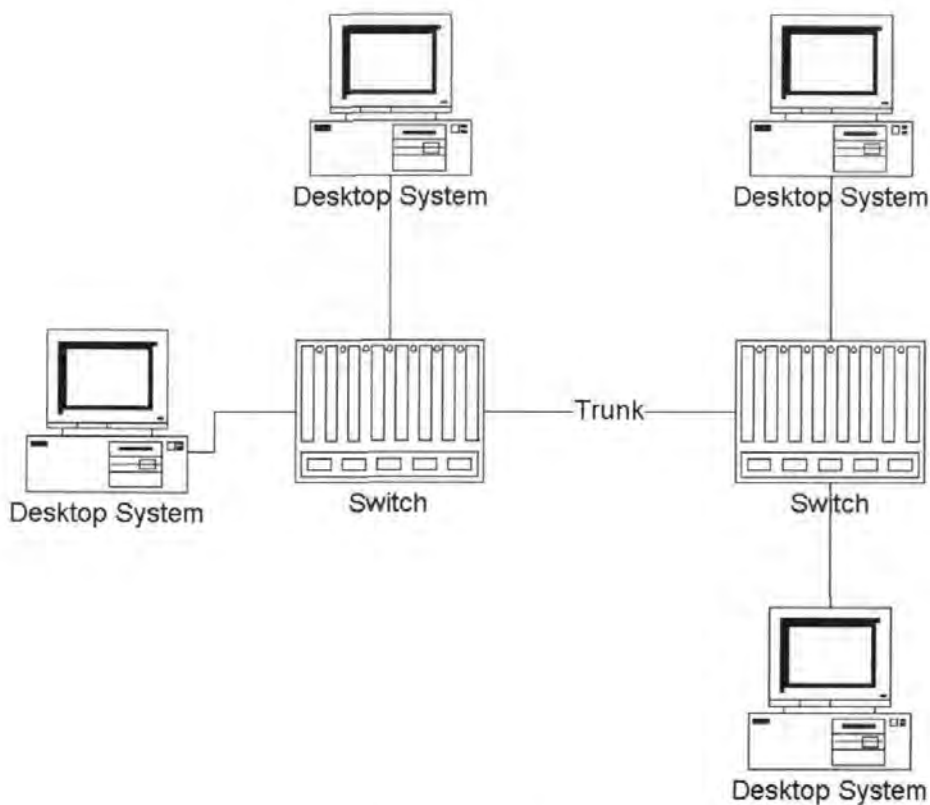


Figure 7.9: Layout for the Simulation

7.6.1 Results of Trunk Overload

To show the behaviour of the BD-B queue for heavy burst loads, a system has been designed as illustrated in Figure 7.9. This picture has been simplified from the real simulation for easier understanding. To show the behaviour of the BD-B queue, the trunk has to be made the bottleneck of the system. To do so the access line and trunk have been configured at 2 Mbit/s. The four TCP/IP (two with MMPP=0.1/0.1/400/600 and two with MMPP=0.1/0.1/300/700) over four PVCs were used. Due to the bursty nature of the traffic, it could be that at some instant one source alone could fill the trunk completely. This is considered normal when a very active TCP/IP source has a large window size. The aggregate mean rate that arrives at the trunk is 1,850 kbit/s, which is more than the 1.6 Mbit/s the trunk can handle. In this simulation the threshold at the BD-B queue was set at 30.

The results are displayed in Figure 7.10 and Figure 7.11. Also more numerical results have been displayed in Appendix C, Sections C.8.1 and C.8.2

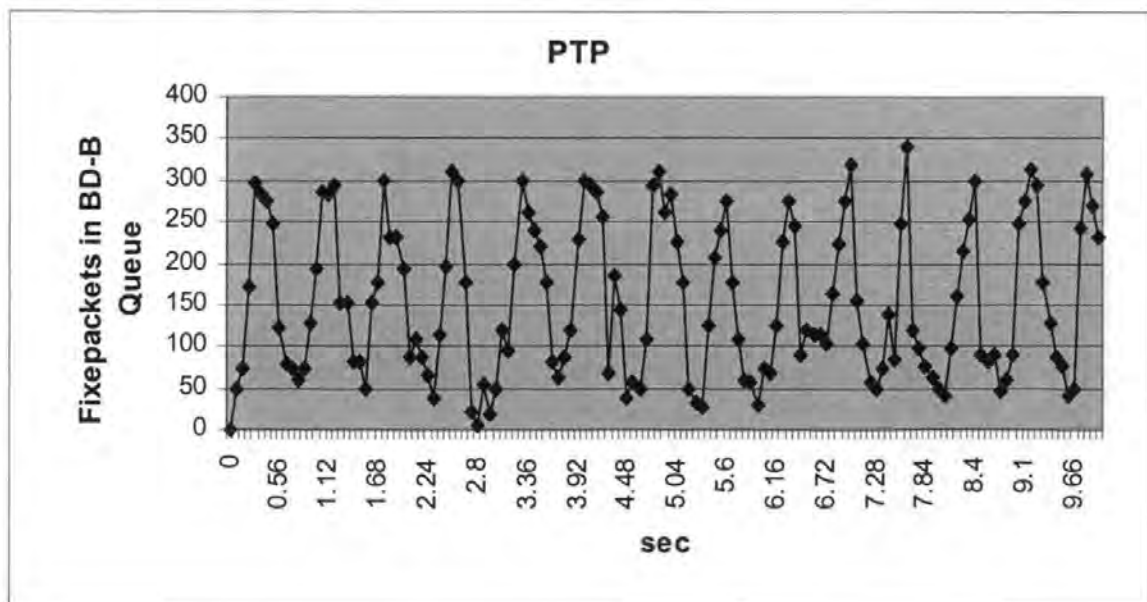


Figure 7.10: BD-B Queue Level Distribution during Simulation with PTP

It is obvious that the BD-B queue of PTP is less “jumpy” than PM. Both queues have been above the threshold for a significant amount of time, but it seems that PM has been more aggressive in generating down messages to throttle the source. It also shows that the PM BD-B queue reaches a lower queue level. This seems due to the fact that PM reacts faster when thresholds have been reached.

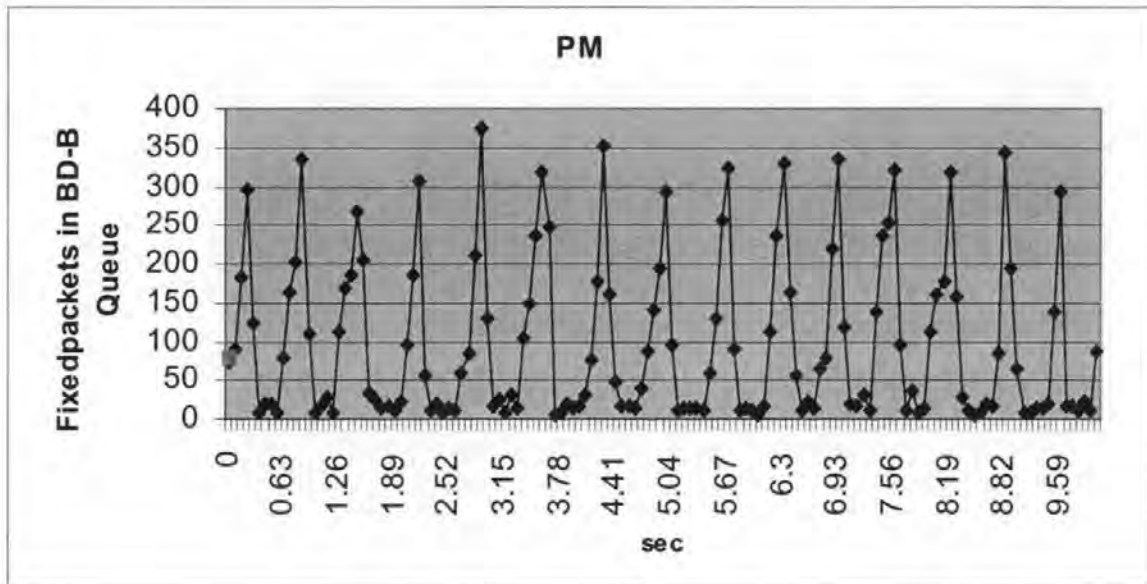


Figure 7.11: BD-B Queue Level Distribution during Simulation with PM

Furthermore, it should be realised that the trunk can only be filled completely when there are Fixedpackets in the BD-B queue at each time instant, which is clearly not the case. The fact that the BD-B queue is nearly empty in PM has two causes. The first cause is the PM reaction delay after clearing the congestion. When the BD-B queue level reaches the BD-B queue threshold, the FECN bits are set in the Fixedpacket headers. If more than 50 per cent of the Fixedpackets arriving at the destination port have the FECN bit set, PM sends a “down” message to the source. This situation continues until the BD-B queue level becomes lower than the BD-B queue threshold. The FECN bit will not be set anymore, but

as there are still a lot of Fixedpackets queued in the network with the FECN bit set, the throttling of the source still continues for a period of time. In the case of PTP, the down messages are not as often, as the mechanism predicts the level of future load and downs not overreact as often. The second cause is that PM has throttled all TCP/IP sources down, regardless of their burstiness. These sources need longer to increase their windows size in comparison to PTP, which has selectively discarded packets and only throttled some sources down. As the CMR is back at the MIR, it needs a couple of “up” messages to get enough data into the network to fill the queue again. In this situation, the MIR is 512 kbit/s and the queue will build up again when both loads are above 800 kbit/s. To raise the CMR from 512 to 800kbit/s with steps of 10 per cent of 512 kbit/s takes around 0.23 seconds.

Here it can clearly be seen that PM actually does not foresee anything, but reacts to the current BD-B queue level. When the BD-B queue level is decreasing, this means that the input rates at the queue are already lower than the service rate of the trunk.

When regarding the robustness of the queue to the input traffic it can be seen that the queue does reach the level of around 300 Fixedpackets at maximum if the BD-B queue threshold is 30. Another conclusion is that PM is probably sensitive to increasing network delays. This does also count for the Transmit Port queue as the PM messages are piggybacked on the Fixedpackets going into the opposite direction of the PVC. This way the reaction time of PM will increase when the network delay in the opposite direction increases, thus making the signalled effect of empty queues worse.

Sometimes the result of the BD-B queue being empty is that the trunk is under-utilised, while traffic is still queued up in the Frame Relay ports. This has the effect that the

average trunk utilisation is around 1.8 Mbit/s. The effect of the BD-B queue being empty sometimes reduces the average throughput by about 10-15 per cent. The shape of the throughput pattern is not affected much by changing the BD-B queue threshold, so in overload situations this loss must be taken for granted. It does not matter for the effect on the trunk if there are 2 PVCs or 200 PVCs as they are all throttled back by 12.5 per cent of their CMR. In overload situations, the sum of the CMRs will be 1.6 Mbit/s so that the PVC throughput will shift in the same way as the trunk throughput.

7.6.2 The Influence of the BD-B Queue Level Threshold on Throughput

To study the influence of various threshold levels and burstiness, the BD-B queue threshold were set at levels ranging from 30 to 150 to see at which levels the throughput starts to degrade.

It is reasoned that setting the BD-B queue threshold at a higher level could increase the trunk throughput, as PM and PTP would react less often. The cost of this could be a higher loss probability in case the queue overflows. A higher threshold also speeds up the CMR earlier; as when the queue level goes down due to PM reactions, the BD-B queue threshold is also reached faster for a higher threshold value. This could also mean that PTP would not recognise the burst increases early enough. To investigate this, the same system as in Figure 7.9 has been used. The four TCP/IP sources with characteristics MMPP=0.1/0.1/300/500 and MMPP=0.1/0.1/400/600 were used. The BD-B threshold queue has been gradually increased from 30 to 160 Fixedpackets and the mean throughput measured during the simulation. The results are given in Figure 7.12 and Figure 7.13. The first conclusion is that raising the BD-B queue threshold does not improve performance a lot. The cost is more significant - an increase in loss probability. It does not seem wise to

increase the threshold to increase performance. Both systems seem to behave similarly to threshold increases.

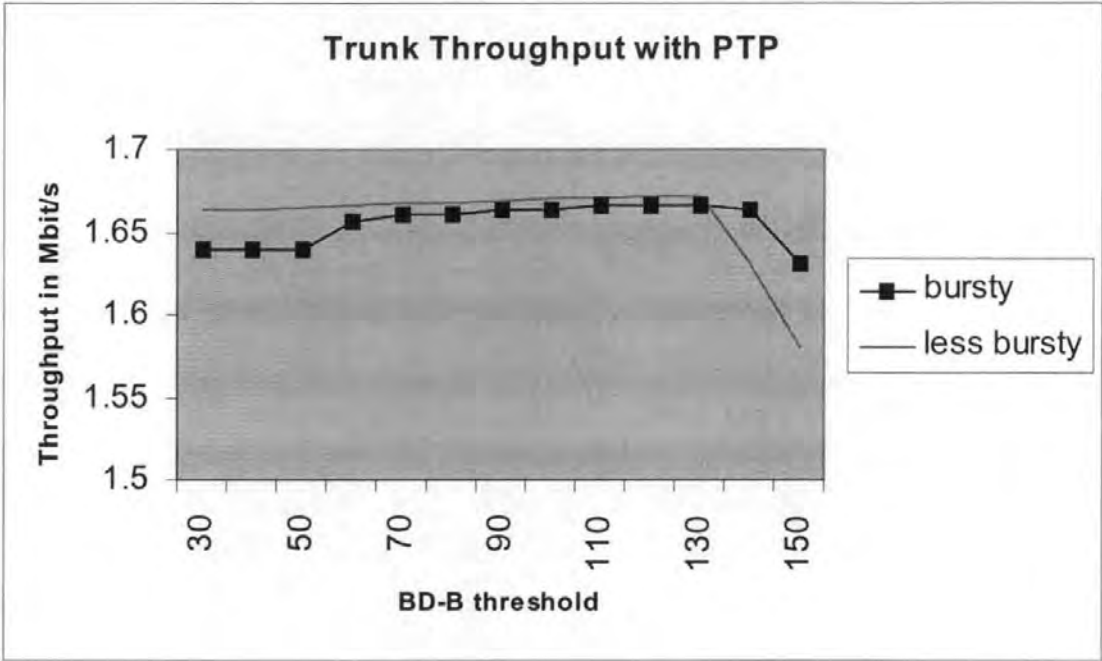


Figure 7.12: Trunk throughput versus BD-B queue threshold – PTP

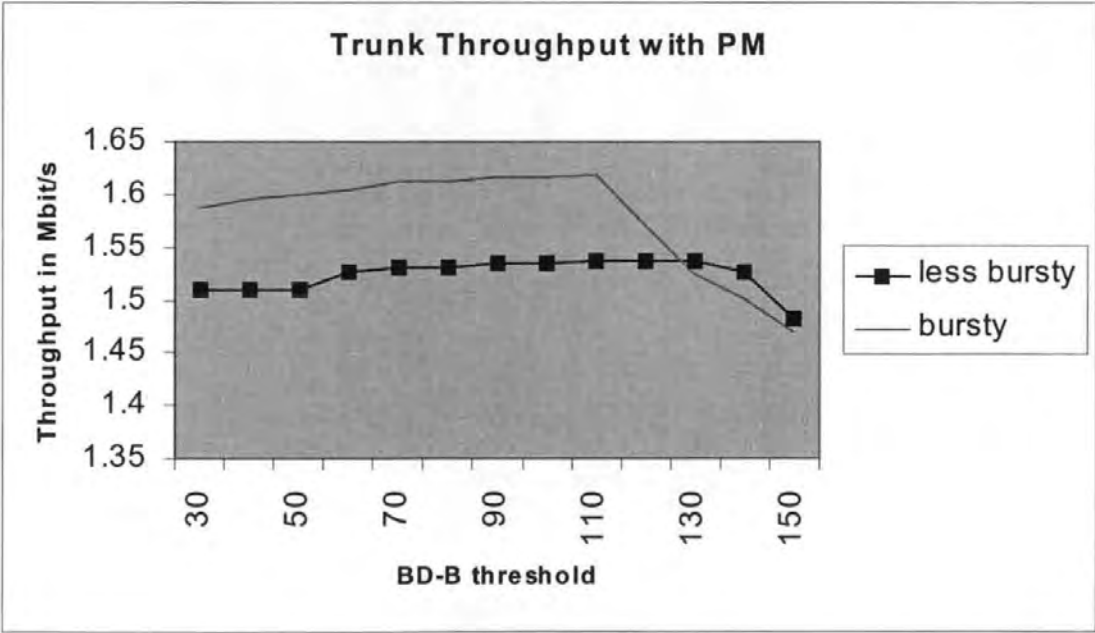


Figure 7.13: Trunk throughput versus BD-B queue threshold – PM

The second observation is that the trunk performance decreases rapidly for increasing burstiness when the threshold is set too high. The difference between the throughput for four MMPP sources with characteristics $MMPP=0.1/0.1/300/500$ and for $MMPP=0.1/0.1/400/600$ is striking for PM at threshold level 100 and above. PTP seems to react better at these levels, but a strong decrease in throughput can also be observed for level of 130. The trunk throughput in Figure 7.12 and Figure 7.13 is including the Fixedpacket header overhead so that for the trunk throughput of user payload, a correction factor of around 0.8 has to be used.

Looking at the average simulation results it is concluded that increasing BD-B thresholds can increase performance slightly in PTP and PM so that the throughput can improve. Raising the BD-B threshold in both mechanisms too high can cause problems in both mechanisms, and decrease throughput and increase delays. The restrictions are that the buffers in the network must not be filled as uncontrolled Frame losses occur, which result in a drastic decrease in performance.

7.7 Simulation for Congestion at the Port

In the past section the focus was on the congestion of the trunk. In this section the focus is on another part of the network: the congestion at the port. In this way both queues that control PM have been investigated.

7.7.1 Overload on the Transmit Port queue

To study the behaviour of the Transmit Port queue, the following set-up has been chosen: Two TCP/IP sources are using one Port of 512 kbit/s and one PVC of 512 kbit/s. The

trunk capacity is 2 Mbit/s. Initially the TCP/IP sources have MMPP parameters set to 0.1/0.1/300/500. This indicates that the traffic will sometimes overflow the queue, however the traffic is not very bursty. At time $t=7$ sec one TCP/IP source has been changed to MMPP=0.5/0.5/100/700 to indicate a higher burst level and compare the behaviour of the two mechanisms. In Figure 7.14 the layout for the simulation can be seen.

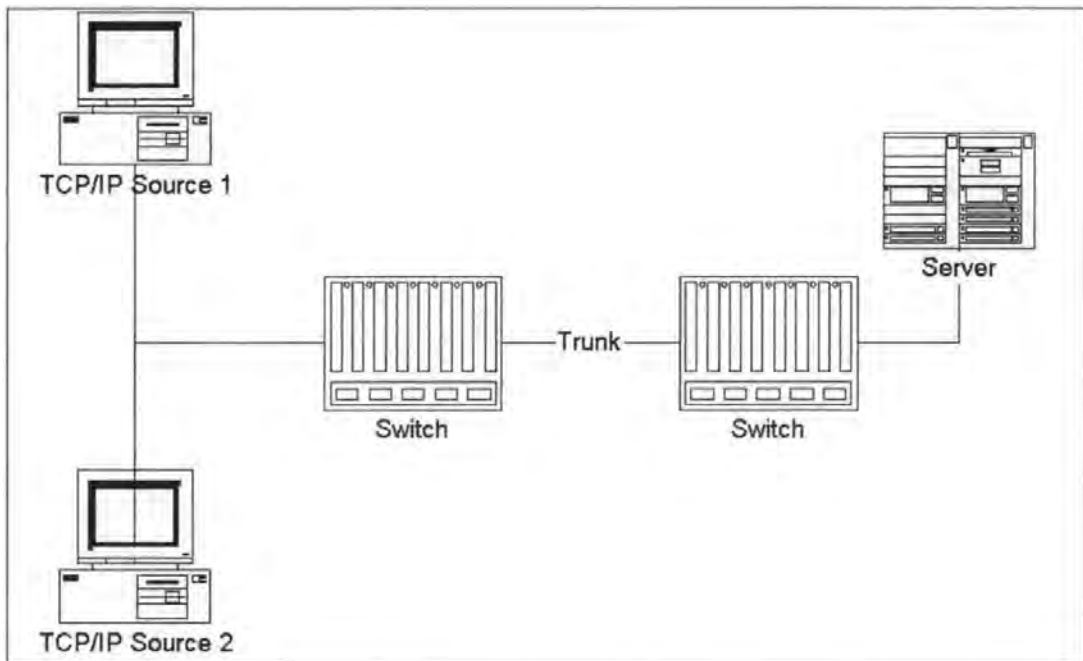


Figure 7.14: Layout for the Transmit Port Queue Simulation

The Transmit Port queue threshold is set at 75 per cent of the queue length (65,535 bytes), which is 49,151 bytes. When this threshold is crossed, PM directly reacts with a down message to the source. To compare the results Figure 7.15 and Figure 7.16 are displayed. Further numerical details are available in Appendix C, Sections C.7.1 and C.7.2.

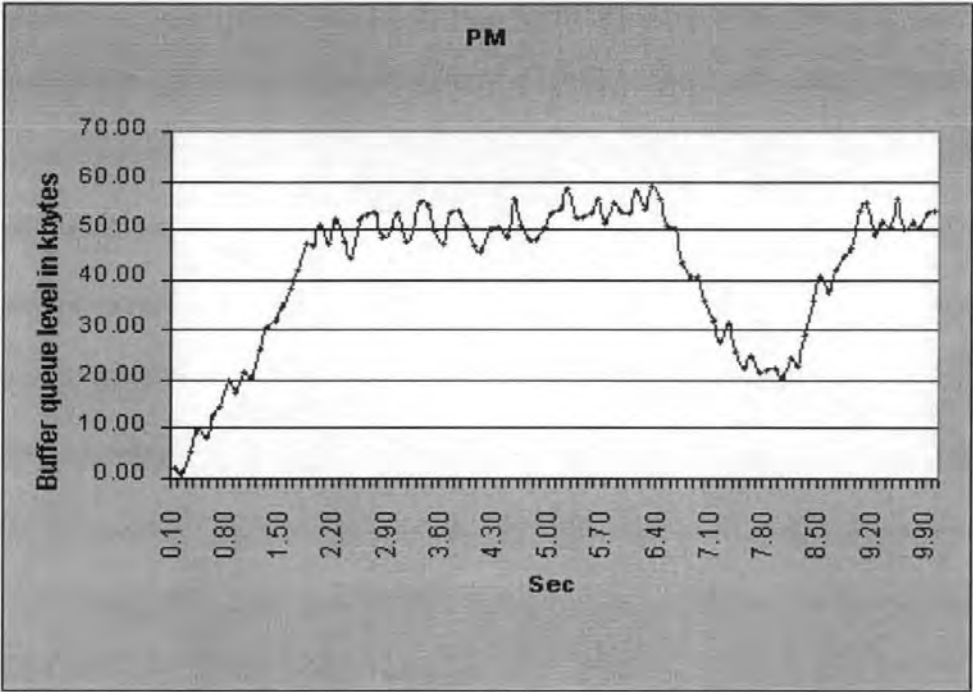


Figure 7.15: Overload on Transmit Port queue in PM

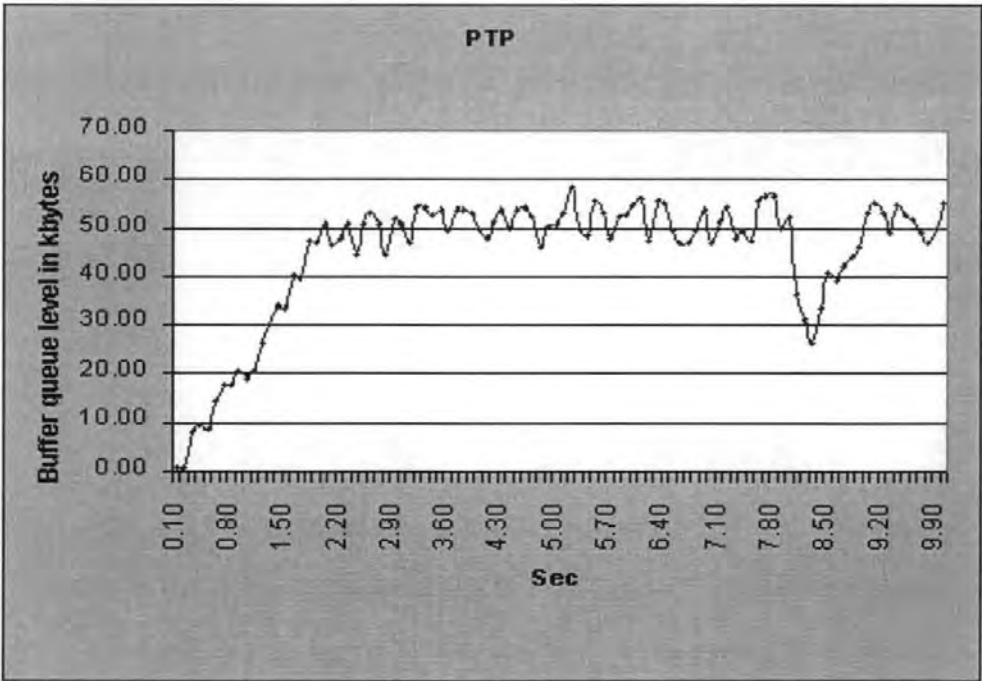


Figure 7.16: Overload on Transmit Port queue in PTP

Figure 7.15 shows how the Transmit port queue builds up when long bursts and high intensity bit rates are used. It is important to note how similar PM and PTP control the

Transmit Port queue level initially. The reason for the Transmit Port queue level dropping at $t=7$ seconds source is that both PVCs experience high intensity from one TCP/IP source. However, it can be seen that the PTP reacts a little smoother and recovers faster.

In the case of PM the queue builds up slower after congestion was experienced for higher deviations from the mean, but also empties faster in comparison to PTP. On first impression it would mean that PM can empty the buffers faster. But this is not the case. The reason that the levels in the PTP buffers are at a higher level than PM buffers is the TCP/IP source. When PM controls traffic, it discards Fixedpackets randomly, therefore in case of congestion the TCP/IP source is throttled back stronger and the “slow start” mechanism of TCP/IP is initiated. In the case of PTP only, these Fixedpackets were discarded, where no congestion is anticipated. Therefore, not all TCP/IP sources have been throttled back.

7.7.2 No Overload at the Transmit Port queue

When the access rate on the port is raised from 512 kbit/s to 1,024 kbit/s in the system of Figure 7.14 there must be no more overload on the Transmit Port queue. The aggregate mean is 800 kbit/s, which will fit easily on average, but what will the effect be of the burstiness? This is shown in Figure 7.17 and Figure 7.18, where the two sources have an aggregate mean of 800 kbit/s and a deviation from the mean varying from changing continuously from 200 (at 2 sec), 150 (at 4 sec) and 100 (at 6 sec).

Both mechanisms behave similarly. This is because the aggregate RateHigh is 1,000 kbit/s, which just fits into the port. But after the deviation was set to 100, the queue levels stayed at relatively low levels.

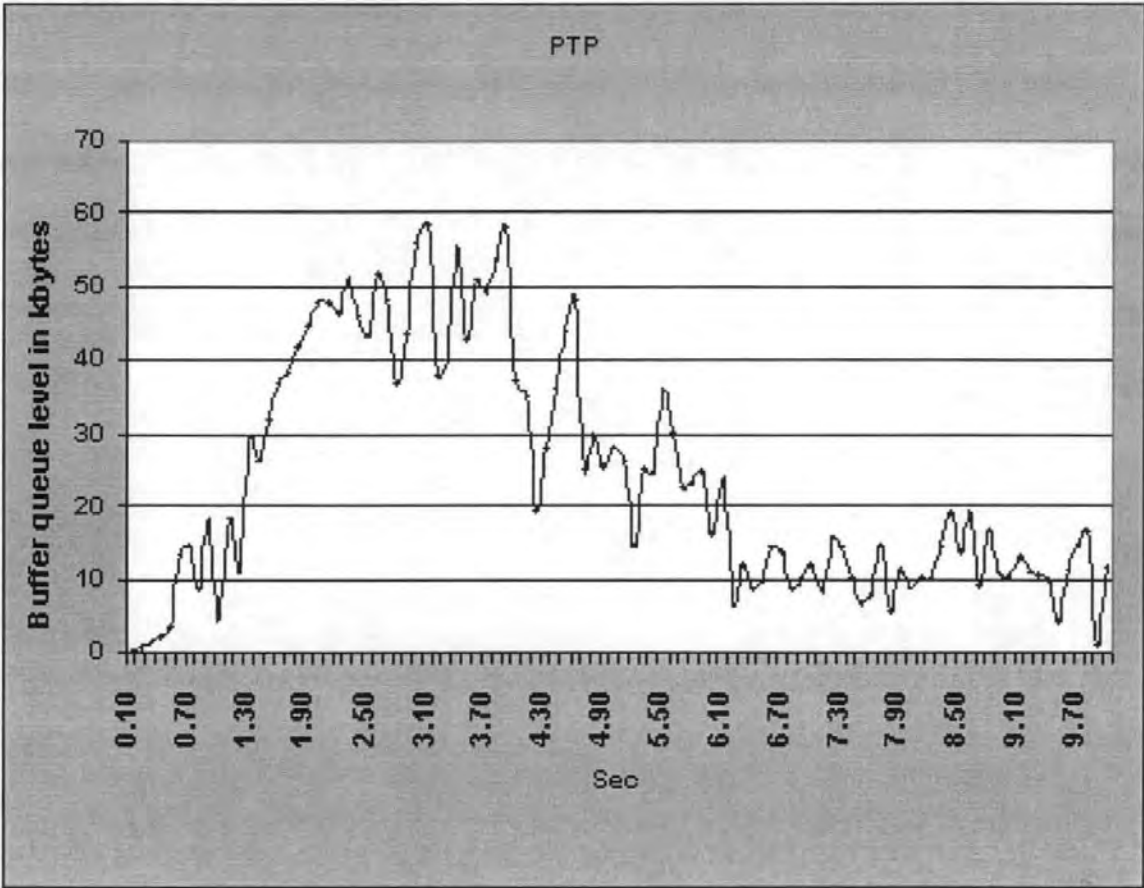


Figure 7.17: No Overload on Transmit Port Queue in PTP

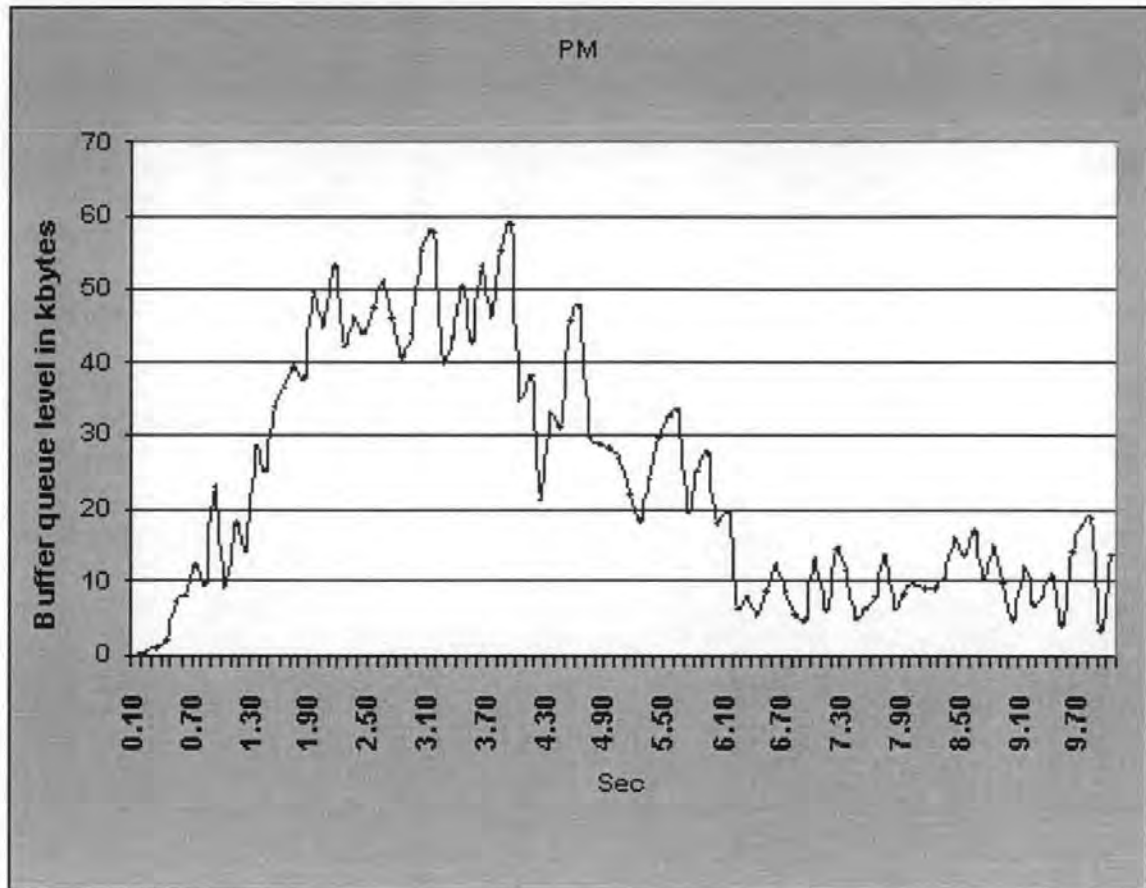


Figure 7.18: No Overload on Transmit Port Queue in PM

The simulation results show that due to the large buffer capacity and the strict reactions of PM, the Transmit Port queue is guarded very well. However, PTP is reacting more favourably with bursty sources.

7.8 Conclusion

In this chapter two possible causes for low effective throughput for TCP over Frame Relay were considered. These are the delivery of inactive Fixedpackets and the retransmission of frames that have already been received.

The primary reason for the low effective throughput of TCP over Frame Relay in the simulations is that when cells are dropped at the switch, the congested link still transmits

the remaining Fixedpackets from “corrupted” frames (that is, frames with at least one Fixedpacket dropped by the switch). This problem of lost throughput due to “dead” Fixedpackets transmitted on the congested link is made worse by any factor that increases the number of Fixedpackets dropped at the switch, such as small buffers, large TCP packets, increased TCP window size, or an increase in the number of active connections.

Larger frame sizes increase the number of wasted Fixedpackets that the congested link transmits when the switch drops a single Fixedpacket from one frame. In addition, the use of larger TCP frames substantially increases the aggressiveness of TCP’s windows algorithm, which in the congestion avoidance phase increases the congestion window by roughly one packet per roundtrip time. Larger frame sizes may be considered advantageous because some end-nodes can process larger packets more cheaply than smaller packets. However, large frames are a performance disadvantage in a congested Frame Relay network, when the congestion control mechanism does not take account for the size of the frames.

A second possible reason for low effective throughput is that the congested link could retransmit frames having already been correctly received. With current TCP window adjustment algorithms, this can only occur when multiple frames are dropped from one window of frames (frames that belong to the same fragment).

The initial set of results served to identify the problem and to motivate the improvement in the congestion handling mechanism. In this section are discussed some of the simulation results with the use of the algorithm described above. The simulation experiments were performed with a maximum TCP packet size of 1,500 bytes, which is the most common

size due to the popularity of Ethernet. From the results it is evident that the end-to-end performance improves as PTP is used in the credit manager.

The use of the algorithm leads to an increase in the TCP/IP throughput and a decrease in the end-to-end delays seen by the end host. The algorithm has two effects on the TCP/IP streams - it detects congestion early and drops a packet as soon as it detects congestion and before it enters the network, also it tries to avoid dropping additional packets. It also reduces the possibility of random drops inside the network. The first packet that is dropped achieves a reduction in the TCP window size and hence the input rates, the algorithm then tries not to drop any more. This keeps the TCP/IP streams active with a higher sustained average rate. In order to improve the PM throughput and delay performance it seems to be important to control the TCP/IP window dynamics and reduce possibilities for negative interaction between the two control loops.

As was seen, uncontrolled multiple packet losses from a window of data can have a catastrophic effect on the TCP throughput. TCP uses a cumulative acknowledgement scheme in which received segments that are not at the left edge of the receive window are not acknowledged. This forces the sender to either wait a roundtrip time to find out about each lost packet, or to unnecessarily retransmit segments which have been correctly received. With the cumulative acknowledgement scheme, multiple dropped segments generally cause the TCP to lose its overall throughput.

Although the principles of the parameters work as described, the influence of changing the parameters to improve the performance of PVCs carrying different traffic characteristics is almost not affected by changing the parameters when the backbone is overloaded. When

interpreting the situation results, it must be realised that the effect of the changing thresholds can have a worse impact on throughput.

The PVC performance in an underload situation is similar for all systems. Smaller packet and message sizes at the source are not recognised by the PTP as well as larger packet sizes and larger windows. For the underload situation it is assumed that the cost for calculation and prediction would be high in comparison to the achieved results.

One important parameter was not taken into account in this research: The increasing amount of hops increases the mixing of different Fixedpackets from different sources. The original packet trains then get more disturbed. The result is that in case of forced buffer clearing, the original mechanism overreacts, as down messages are sent to all sources, regardless of further congestion level expectations.

The importance of network configuration parameters – switch buffer size, TCP packet size, and TCP window size – should not suggest that the fragmentation problem can be completely solved by appropriate configuration settings, which offer only partial solutions. Large buffers can result in unacceptably long delays, and it is not always possible to use small packets in an Internet environment. In addition, the beneficial effect of small windows, small packets, or large buffers can be offset if the number of contending connections increase.

The necessity for considering end-to-end traffic management in TCP/IP over Frame Relay networks was discussed. In particular the impact of the interactions between the TCP and the proprietary congestion control was considered, in providing end-to-end quality of

service to the user. It is important to consider the behaviour of the end-to-end protocol like TCP when selecting the traffic management mechanisms in the TCP/IP Frame Relay networking environment. The PM control pushes the congestion out of the Frame Relay network but builds up queues at the access point of the network. This does not always solve the problem. It was shown that it could have a bad effect on the end-to-end performance. It was also shown that PM with a combination of PTP in the CMR can achieve better performance than pure PM.

An algorithm for handling the congestion was presented which improves the end-to-end throughput and delay significantly while using PM. The algorithm couples the two flow control loops by using the information received by the explicit rate scheme to intelligently discard packets when there is congestion anticipated in the network. In addition the algorithm is sensitive to the TCP/IP flow control and recovery mechanism to make sure that the throughput does not suffer due to closing of the TCP window. As presented in this chapter, the algorithm is limited to cases where different TCP streams are queued separately. As future work, the algorithm will be extended to include shared memory cases.

Chapter 8: Conclusion

8.1 Conclusion

The three main goals of this thesis have been to gather, examine and use operational statistics to characterise network traffic; to develop a methodology with these statistics for overbooking and network planning tasks; and to develop and simulate a congestion-control mechanism that can facilitate a more accurate workload mechanism on Frame Relay than current congestion-control mechanisms allow.

The investigation into these goals have resulted in:

- The definition of traffic into *application signatures* and *PVC life cycle*,
- New overbooking methodology for PVCs, and
- New congestion control mechanism for Frame Relay.

To fulfil the first goal, the workload characteristics in different environments have been measured and evaluated as shown in Chapter 5. These formed a base from which to measure a complex system of interconnected components. Existing operational statistics on core network backbones do not take account of user behaviour. The collection and analysis of traffic profiles for short-term and long-term periods have both led to new classifications of traffic.

Long-term profiles were analysed in granularities of months, weeks, days and hours to get an impression of data loads and growths on Frame Relay systems. The virtual connections often showed the same repeatable patterns, which led to the new classification: *PVC life cycle*. This life cycle occurs on every PVC, however the time granularities differ. Also the cycle is not linear, but can change from stage to stage, influenced by the change of a

company structure, software and hardware upgrades, and introduction of new services.

Short-term profiles were analysed in granularities of seconds. To establish a view of short-term predictability, measurements on individual applications for packet size distributions and probability of packet arrivals have been analysed. The results show that these profiles have unique and repeatable patterns, and have led to a new model: *application signatures*. Using packet traces the effect of several parameters were explored, such as size of packets in packet trains, and metrics of individual flows. These flows include volume in packets, bytes per flow, flow duration, and source-to-target ratios. The research has highlighted the possibilities of *application signatures*, but also shows that there is a need for further investigation.

To fulfil the second goal, the analysed long-term traffic patterns and the resulted PVC life cycle were used to develop a new methodology for PVC overbooking. This resulted in the development of individual overbooking factors (OBF). The use of this methodology allows the network planner to use and find the appropriate OBF in the Frame Relay network. Utilising the PVC life cycle helps a network planner in the assessment of a PVC and assists in the understanding of PVCs' behaviour. It was discussed how the use of pure statistics limits the ability to assess the network, and completely prevents the assessment of other items in the usual planning process. The evaluation highlighted an important issue: statistics collection in currently-deployed network components is typically driven by short-term requirements, e.g., immediate operational status information or engineering data such as aggregated link utilisation. Long-term interpretation of statistics often takes a back-seat to more immediate network management; resulting inattentive data collection prevents collection at the level of detail, completeness and confidence needed for many

workload characterisation tasks. Resulting statistics allow some tracking of Frame Relay growth, but limit the ability to forecast capacity requirements in a network with ever richer functionality and services, and also do not allow quantification of detailed traffic characteristics, which vary considerably in both time and space granularities. The new overbooking provides companies with cheaper network design, less packet loss and higher average throughput. In addition, new requirements like risk factors have been incorporated into the methodology, which lay historically outside the design process. The methodology was developed for describing PVC behaviour in terms of their impact on an aggregate network workload, and was tested on the Frame Relay network of AT&T in Europe. The methodology defines a flow based on actual traffic activity and defines an individual overbooking mechanism, rather than using general overbooking factors.

The third goal of this thesis focuses on the development and simulation of a new mechanism to address the predictability of packet arrivals for congestion control purposes. Specifically, *application signatures* were used for the development of the packet train profiler (PTP) model. This model can be used in future congestion-control mechanisms for better network performance. To compare the new mechanism with an existing proprietary mechanism, various simulations were set up and run and the results were compared and discussed. Buffer utilisation levels in the switch and also at the access port have been analysed and evaluated. The simulations have shown that existing mechanisms do not use buffers as efficiently as the new PTP mechanism. PTP leads to better throughput and delay characteristics and also helps understanding short-term traffic flows and behaviour. Applying the mechanism to the measurements yielded significant observations of protocol behaviour, which have implications for performance requirements of switches, general and specialised flow-based routing algorithms. It could also have influence on future usage-

based accounting requirements and traffic prioritisation.

The author believes that the present limitations in the interpretation of current statistics are in tracking the tremendous growth in Frame Relay application and service diversity. Although the statistics indicate a proliferation of utilised ports, there is no mechanism to determine what application, or even class of application, an arbitrary port carries. Yet, assessing the service profiles of these new applications will be important to accommodate them on a Frame Relay network. In particular, as newer continuous flow multimedia applications contribute to the complexity of the aggregate demand, they will require re-evaluation of design issues such as queuing management in routers in order to provide multiple service classes. Even within the non-continuous flow paradigm, sub-categories of traffic such as interactive or bulk may exhibit performance requirements which necessitate adaptive queue management. Operators of Frame Relay networks could clearly benefit from a more accurate assessment of the impact of certain applications on the overall demand they must satisfy. However, all such collection comes at a cost, and a network operator must weigh these costs against the benefits the availability of such statistics will provide.

It should be noted that applications are changing characteristics, as they introduce traffic flows with different behaviour, particularly real-time continuous media flows, which tend to exhibit greater duration and flow volume. Future work should include the understanding of different protocols and their interactions with each other and with applications, i.e., how individual flows and the aggregate flow profile influence each other. It helps in securing network stability, and requires ongoing flow assessment to track changes in a workload in a given environment. The importance of developing existing methodologies further helps

to understand the ever-increasing complexity of WAN. Factors, which influence network planning, include the increasing complexity in equipment, connectivity, service expectations, and financial structure. The combination of these factors will make critical not only improved statistics collection, but also technology for accounting and billing, accompanied by network mechanisms such as queue management and routing for multiple service qualities and applications.

The Internet will not be able to secure and maintain stability in the face of new traffic types and continued explosive growth without a more dedicated approach to Internet traffic analysis, the first step of which is an accurate workload, or flow, characterisation. The new assessment methodology, described in Chapter 6, can form a complementary component to existing operational planning, yielding insights into larger issues of network evolution, i.e., how environments of different aggregation can cope with resource contention by an ever-changing composition and volume of flows. The focus is on methodologies and representative environments, not an exhaustive exploration of all possible environments or questions. Flow characteristics are changing, and it is intended that the presented methodology shall serve as a tool for those who wish to track and keep pace with its change. For example, as video and audio flows, and even single streams combining voice and audio, become more popular, service providers will need to parameterise them to determine how many such customer streams they will be able to support, and how many more resources each new such customer would require. Multicast flows will also likely constitute an increasingly significant component of network traffic, and applying the methodology to multicast flows would be an important step toward coping with their impact on the infrastructure.

Because it requires comprehensive and detailed statistics collection, service providers may not be able to afford to continuously monitor flow characteristics on an operational basis. Nonetheless, it is the author's opinion that it will be necessary that network equipment providers undertake the described assessments in the future to obtain a more accurate picture of the short-term workload their infrastructure must support. The congestion mechanism developed in Chapter 7 can serve as a valuable tool for such assessments.

There are many opportunities for further research in the area of traffic profiling in short-term and long-term granularities. It is important to see both views combined and investigated for planning purposes. Especially the field of application signatures should be investigated further. This could lead to the development of better congestion control mechanisms, and also overbooking mechanisms for commercial services.

References

[ATT] AT&T StrataCom Design Rules (Release 8.1.71)

[Bae91] J. Bae and T. Suda, "Survey of Traffic Control Schemes and Protocols in ATM Networks", Proc of the IEEE, Feb 1991

[Ber91] A. Berger and A. Eckberg, "A B-ISDN/ATM Traffic Descriptor, and its use in Traffic and Congestion Controls", Proc. IEEE GLOBECOM '91, 1991, pp. 266-270

[Bert92] D. Bertsekas and R. Gallager, (1992) Data Networks, Second Edition, Prentice-Hall

[Case88] Case, Fedor, Schoffstall, & Davin (August 1988) RFC 1067 - SNMP Architecture, Internet Engineering Task Force (IETF)

[Case91] R. Caseres, P. Danzig, S. Jamin & J. Mitzel, (1991) Characteristics of Wide Area TCP/IP Conversations, Proceedings of the 1991 ACM SigComm Conference

[Che89] K. Chen and K. Rege, "A Comparative Performance Study of Various Congestion Controls for ISDN Frame-Relay Networks", Proc. IEEE INFOCOM '89, April 1989, pp. 674-675

[Chiu92] D.M. Chiu and R. Sudama, (1992) Network Monitoring Explained, Design and Application, Ellis Horwood

[Claffy 93] K. Claffy, G. Polyzos, and H.W. Braun, "Traffic Characteristics Of The T1 NSFNET Backbone", Proceedings Of INFOCOM '93, San Francisco, March, 1993

[Clark95] Clark, H., (Sept. 1995), Network Working Group, Request for Comments 1856
Category: Informational

[Com90] D. Comer and R. Yavatkar, "A Rate-Based Congestion Avoidance and Control Scheme for Packet Switched Networks", Proc. Of 10th ICDCS, IEEE, 1990, pp. 390-397

[Dav72] Davies, D., "The Control of Congestion in Packet Switch Networks", IEEE Trans. On Commun., COM 20, no. 3, June 1972

[DEC82] Digital Equipment Corporation, DECnet Digital Network Architecture [phase IV] General Description, Order AA-N149A-TC, Digital Equipment Corporation, 1982

[Dem89] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queuing Algorithm", Proc. Symp. On Communication Architectures and Protocols, ACM SIGCOMM '89, Sept. 1989

[Dos88] B. Doshi and H. Nguyen, "Congestion Control in ISDN Frame-Relay Networks", ATT Technical Journal, Nov/Dec 1988, pp. 35-46

[Eck89] A. Eckberg, D. Luan, and D. Llucantoni, "Meeting the Challenge: Congestion and Flow Control Strategies for Broadband Information Transport", Proc. IEEE GLOBECOM '89, 1989, pp. 1769-1773

[Fin89] Finn, G., "A Connectionless Congestion Control Algorithm", *Computer Commun. Review*, vol. 19, no. 5, Oct 1989, pp. 12-31

[Floyd91] Floyd, S., Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic. *Computer Communications Review*, vol. 21, no. 5, October 1991, pp. 30-47

[Floyd97] S. Floyd and K. Fall, Router Mechanisms to Support End-to-End Congestion Control. Technical Report, February 1997

[Fort90] P.J. Fortier and G.R. Desrochers, (1990) *Modelling and Analysis of Local Area Networks*, Multiscience Press

[Gel91] Gelostani, S., "Congestion-Free Communications in High-Speed Packet Networks", *IEEE Trans. On Commun.*, vol. 39, no. 12, Dec 1991

[Geo82] F. George and G. Young, "SNA Flow Control: Architecture and Implementation", *IBM System Journal*, vol. 21, no. 2, 1982, pp. 179-210

[Ger80] M. Gerla and L. Kleinrock, "Flow Control: A Comparative Survey", *IEEE Trans. On Commun.*, vol. 28, no. 4, Apr 1980, pp. 553-574

[Ger88] M. Geria and L. Kleinrock, "Congestion Control in Interconnected LANs", *IEEE Network*, vol. 2, no. 1, Jan 1988

[Ger89] A. Gersht and K. Lee, "A Congestion Control Framework for ATM Networks", Proc. IEEE INFOCOM '89, April 1989, pp. 701-710

[Gon94] Y. Gong and I. Akyildiz, "Dynamic Traffic Control Using Feedback and Traffic Prediction in ATM Networks", Proc. IEEE INFOCOM, 1994, pp. 91-98

[Gra91] A Gravey, P. Boyer, and G. Hebuterne, Tagging versus Strict Rate Enforcement in ATM Networks, Proc. IEEE GLOBECOM '91, 1991, pp. 271-275

[Grout89] Grout, V. M., (1989) Network Traffic Analysis Project - Final Report, Polytechnic South West

[Haa91] Haas, Z., "Adaptive Admission Congestion Control", ACM SIGCOMM ' 91, 1991

[Harr93] P.G. Harrison and N.M. Patel, (1993) Performance Modelling of Communication Networks and Computer Architectures, Addison-Wesley

[Hege93] H-G. Hegering and A. Laepple, (1993) Ethernet, Building a Communications Infrastructure, Addison-Wesley

[Held92] Held, G., (1992) Network Management, Techniques, Tools and Systems, John Wiley & Sons

[Held93] Held, G., (1993) Internetworking, LANs and WANs, Concepts, Techniques and Methods, John Wiley & Sons

[Held94] Held, G., (1994) Local Area Network Performance, Issues and Answers, John Wiley & Sons

[Jac88] Jacobson, V., Congestion Avoidance and Control. Proceedings of ACM SIGCOMM '88. August 1988, pp. 314-329

[Jai94] Jain, R., (1994) "A Time-based Congestion Control Scheme for Window Flow-controlled Networks", IEEE JSAC, SAC-4, no. 7

[Jai88] R. Jain and K. Ramakrishnan, "Congestion Avoidance in Computer Networks with a Connectionless Network Layer; Concepts, Goals, and Methodology", Proc. of Computer Networking Symposium, April 1988

[Jain86] R. Jain and R. Routhier, (1986) Packet Trains, Measurements and a New Model for Computer Network Traffic, IEEE Journal on Selected Areas in Communications, SAC-4, No. 6

[Jones95] Jones, M., (1995) Routing over the AT&T Frame Relay Network, Dissertation for MSc. in Management Science and Operational Research, University of Warwick

[Ker91] Keshav, S., "A Control-Theoretic Approach to Flow Control", Proc. SIGCOMM '91, Sept. 1991

[Ker93] Kershenbaum, A., (1993) Telecommunications Network Design Algorithms, McGraw-Hill

[Klei75] Kleinrock, L., (1975) Queueing Systems, Volume I: Theory, John Wiley & Sons

[KleinII75] Kleinrock, L., (1975) Queueing Systems, Volume II: Theory, John Wiley & Sons

[Kleinrock 76] Kleinrock, L., (1976) Queueing Systems, Volume II: Theory, John Wiley & Sons

[KM87] C.Kent and J.Mogul, Fragmentation Considered Harmful, ACM SIGCOMM '87, Aug. 1987, p. 390

[Kun94] H. Kung, T. Blackwell, and A. Chapman, "Credit-Based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multitplexing", Proc. SIGCOMM '94, London, UK, Aug 31-Sept 2, 1994, pp. 101-114

[Lam79] S. Lam and M. Reiser, "Congestion Control of Store-and-Forward Network by Input Buffer Limits Analysis", IEEE Trans. On Commun., COM-27, no. 1, Jan 1979

[Lel89] Leland, W., "Window-Based Congestion Management in Broadband ATM Networks: The Performance of Three Access-Control Policies, Proceedings", IEEE GLOBECOM '89, 1989, pp. 1794-1800

[Law91] A. Law and D. Kelton, (1991) Simulation Modeling & Analysis, McGraw Hill, Inc.

[Little61] Little, J.D.C., (1961) A Proof of the Queuing Formula $L=\lambda W$, Operations Research 9,3

[Mak91] Makrucki, B., "On the Performance of Submitting Excess Traffic to ATM Networks", Proc. IEEE Globecom '91, 1991, pp. 281-287

[Man90] Mankin, A., "Random Drop Congestion Control", Proc. Of SIGCOMM '90, 20, no. 4, Sept. 1990

[Man91] A. Mankin and K. Ramakrishnan, "Gateway Congestion Control Survey", RFC-1254, Network Working Group, Aug. 1991

[Mars92] Marsden, B.W., (1992) Communication Network Protocols, OSI Explained, 3rd Edition, Chartwell-Bratt

[Mis92] P. Mishra and H. Kanakia, "A Hop by Hop Rate-based Congestion Control Scheme", Proc. SIGCOMM '92, Oct. 1992

[Mue99] G. Mueller, P. Sanders and J. Allen, "Traffic profiles and application signatures", Computer Communications, (22)12 (1999) pp. 1123-1126

[Muk88] Mukherji, U., "A Schedule-Based Approach for Flow-Control in Data Communication Networks", Ph.D. Thesis, Massachusetts Institute of Technology, Feb. 1986

[Muk92a] A. Mukherjee, L. Landwebber and T. Faber, "Dynamic Time Windows and Generalized Virtual Clock; Combined Closed-loop/Open-loop Congestion Control", IEEE INFOCOMM '92

[Muk92b] A. Mukherjee, L. Landwebber and T. Faber, "Dynamic Time Windows: Packet Admission control with Feedback", Proc. SIGCOMM '92, Oct. 1992

[Mull90] N.J. Muller and R.P. Davidson, (1990) LANs to WANs: Network Management in the 1990s, Artech House Boston

[Murd78] Murdoch, J., (1978) Queueing Theory- Worked Examples and Problems, MacMillan Press Ltd

[Nag87] Nagle, J., "On Packet Switches with Infinite Storage", IEEE Trans. on Commun., vol. 35, no. 4, April 1987, pp. 435-438

[Nuss90] Nussbaumer, H., (1990) Computer Communication Systems, John Wiley & Sons

[OM96] T. Ott, J. Kemperman and M. Mathis, "Window Size Behavior in TCP/IP with Constant Loss Probability", DIMACS Workshop on Performance of Realtime Applications on the Internet, Plainfield NJ, November 6-8, 1996

[Oper93] Operating Manual DA30/31 Network Analyzer (1993) Wandel & Goltermann

[OSI] ISO 7498-4 OSI, Management Framework

[Par93] Park, K., "Warp Control: A Dynamically Stable Congestion Protocol and Analysis", Proc. SIGCOMM '93, Sept. 1993

[Par94] Partridge, C., Gigabit Networking", Addison-Wesley Publishing Co., 1994.

[Paxs91] Paxson, V., (1991) Measurements and Models of Wide Area TCP Conversations, Report LBL-30840, Lawrence Livermore Laboratories

[Paxson94] Paxson, Vern, Growth Trends In Wide- Area TCP Connections, IEEE Networks, 8(4) July/August 1994

[Pax95] V. Paxson and S. Floyd, (1995) Wide Area Traffic: The Failure of Poisson Modelling, IEEE/ACM Transactions on Networking No.3 June

[Phaal95] Phaal, P., (1995) LAN Traffic Management, Prentice Hall

[Postel81] Postel, J., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", RFC 793, DARPA, September 1981

[Pos81] J.B. Postel, C.A. Sunshine, and D. Cohen, "The ARPA Internet Protocol", *Computer Networks*, vol. 5, no. 4, 1981, pp. 171-261

[Pru87] W. Prue and J. Postel, "Something a Host Could Do with Source Quench", RFC-1016, Network Working Group, July 1987

[Ray93] Ray, G., "Computer Network analysis and Optimisation", PhD Thesis, University of Plymouth, August 1993

[Ram90] K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks", *ACM Trans. On Computer Systems*, vol. 8, no. 2, May 1990, pp. 158-181

[Ram91] K. Ramakrishnan, R. Jain, and D. Chiu, "A Selective Binary Feedback Scheme for General Topologies", *Networks*, Tech. Report, DEC-TR-510, Digital Equipment Corporation, 1991

[Reyn93] Reynolds, P. L., *Expert System Approach To Private Telecommunications Network Design*, *BT Technology Journal*, vol.11, no. 4, October 1993

[RFC1191] J. Mogul and S. Deering, Path MTU Discovery. Request For Comments 1191. November 1991

[RFC1271] RFC1271 (November 1991) Remote Network Monitoring MIB, Internet Engineering Task Force (IETF)

[RFC1709] RFC 1709 (November 1994) K-12 Internetworking Guidelines, Internet Engineering Task Force (IETF)

[Rob90] J. Robinson, D. Friedman, and M. Steenstrup, "Congestion Control in BBN Packet-Switched Networks", ACM Computer Commun. Review, Jan. 1990, pp. 76-90

[Rom94] A. Romanow and S. Floyd, "Dynamics of TCP Traffic over ATM Networks", Proc. SIGCOMM '94, London, UK, Aug. 31-Sept. 2, 1994, pp. 79-88

[Rose91] Rose, M.T., (1991) The Simple Book, An Introduction to Management of TCP/IP-based Internets, Prentice Hall

[Ros92] Rose, O., "The Q-bit Scheme", ACM Computer Commun. Review, April 1992, pp. 29-42

[Rosn82] Rosner, R.D., (1982) Packet Switching, Tomorrow's Communications Today, Wadsworth

[Salah94] Salah Aidarous & Thomas Plevyak, (1994) Telecommunications Network Management into the 21st Century, IEEE Press

[Sant91] Santifaller, M., (1991) TCP/IP and NFS, Internetworking in a UNIX Environment, Addison-Wesley

[Schl90] Schlar, S.K., (1990) Inside X.25: A Manager's Guide, McGraw-Hill

[Smith93] Smith, P., (1993) Frame Relay, Principles and Applications, Addison-Wesley

[Sta93] Stallings, W., Networking Standards; a Guide to OI, ISDN, LAN, and MAN Standards, Addison-Wesley Publishing Co., Reading, Mass., 1993

[Stall91] Stallings, W., (1991) Data and Computer Communications, Third Edition, Macmillan

[Stall93] Stallings, W., (1993) SNMP, SNMPv2, and CMIP, The Practical Guide to Network-Management Standards, Addison-Wesley

[Strata] Stratacom IPX System User's Guide Rel. 6.2

[Tan81] Tanenbaum, A., Computer Networks, Prentice-Hall, Englewood Cliffs, NJ, 1981, 3rd edition 1996

[Terp92] Terplan, K., (1992) Effective Management of Local Area Networks, Functions, Instruments, and People, McGraw-Hill

[Var86] R. Varakulsiripunth, N. Shiratori, and S. Noguchi, "A Congestion-Control Policy on the Internetwork Gateway", Computer Networks and ISDN Systems, 1986, pp. 43-58

- [Wan91] Z. Wang and J. Crowcroft, "A New Congestion Control Scheme; Slow Start and Search [Tri-S]", *Computer Commun. Review*, vol. 21, no. 1, Jan. 1991, pp. 32-43
- [Wil93] Williamson, C., "Optimizing File Transfer Response Time Using the Loss-Load Curve, Congestion Control Mechanism", *Proc. SIGCOM '93*, Sept. 1993
- [Wood93] Woodward, M., (1993) *Communication and Computer Networks*, Pentech Press
- [Yin90] N. Yinn, S. Li, and T. Stern, "Congestion Control for Packet Voice by Selective Packet Discarding", *IEEE Trans. On Commun.*, vol. 38, no. 5, May 1990, pp. 674-683
- [Yin94] N. Yin and M Hluchyj, "On Closed-Loop Rate Control for ATM Cell Relay Networks", *Proc. IEEE Infocom*, 1994, pp. 99-108
- [Zha91] H. Zhang and S. Keshav, "Comparison of Rate-Based Service Disciplines", *Proc. SIGCOMM '91*, Sept. 1991
- [Zha87] Zhang, L., "Congestion Control in Packet-Switched Computer Networks", *Proc. Of 2nd Int'l Conf. On Computers and Applications*, 1987, pp. 273-280
- [Zha90] Zhang, L., "VirtualClock: A New Traffic Control Algorithm for Packet Switching Networks", *Proc. SIGCOMM '90*, Sept. 1990, pp. 19-29

Appendix A – Simulation Code

! WITNESS MODEL: Frame Relay

* Title : Frame Relay - Packet Train Simulation
 * Author : Georg Mueller
 * Date : Sat Aug 21 14:34:56 1999
 * Version: WIN-307 Release 8.0

DEFINE

MACHINE: Framer,7,Production,0,0;
 BUFFER: Test,4,30000;
 MACHINE: Trunk,7,Single,0,0;
 BUFFER: LostPckt,1,10000;
 PART: Frame,Variable attributes;
 ATTRIBUTE: Bytes,1,Integer,1;
 BUFFER: OK,2,10000;
 VARIABLE: fpbytes,1,3,Integer;
 VARIABLE: NoFrames,1,3,Integer;
 TIMESERIES: Bursts,1;
 ATTRIBUTE: indx,1,Integer,1;
 VARIABLE: A,1,1,Integer;
 MACHINE: Gen1,1,Single,0,0;
 VARIABLE: INX,1,3,Integer;
 VARIABLE: lenght,1,3,Integer;
 ATTRIBUTE: lenghta,1,Integer,1;
 MACHINE: FlowCont,1,Batch,0,0;
 VARIABLE: Lostpakt,1,1,Integer;
 TIMESERIES: Utilise,1;
 VARIABLE: LostByte,1,1,Real;
 MACHINE: Gen2,1,Single,0,0;
 VARIABLE: lambda,1,1,Real;
 VARIABLE: lambda1,1,5,Real;
 VARIABLE: Byte,1,1,Integer;
 DISTRIBUTION: bimodal,Integer,Discrete;
 DISTRIBUTION: bimod,Real,Discrete;
 VARIABLE: tspeed,1,1,Real;
 TIMESERIES: Packloss,1;
 MACHINE: Gen3,1,Single,0,0;
 DISTRIBUTION: mydistr,Real,Discrete;
 DISTRIBUTION: distr,Real,Discrete;
 DISTRIBUTION: testdist,Integer,Discrete;
 DISTRIBUTION: myd1,Real,Discrete;
 DISTRIBUTION: myd2,Integer,Discrete;
 VARIABLE: pspeed,1,3,Integer;
 VARIABLE: gate,1,3,Integer;
 VARIABLE: b,1,1,Integer;
 VARIABLE: QIR,1,3,Real;
 VARIABLE: CIR,1,3,Real;
 VARIABLE: PIR,1,3,Real;
 VARIABLE: MIR,1,3,Real;
 VARIABLE: Util,1,3,Real;
 VARIABLE: teta,1,3,Real;
 BUFFER: BD_B,7,600;
 VARIABLE: steps,1,1,Real;
 MACHINE: CManager,3,Single,0,0;
 VARIABLE: Cmax,1,3,Integer;
 PART: Credit,Variable attributes;
 VARIABLE: Credits,1,3,Integer;
 VARIABLE: CMR,1,3,Real;
 TIMESERIES: speed,1;

```

PART: timer,Variable attributes;
ATTRIBUTE: EFCN,1,Integer,1;
VARIABLE: EFCNBITS,1,3,Real;
VARIABLE: arrived,1,3,Integer;
VARIABLE: OnOff,1,1,Integer;
ATTRIBUTE: portid,1,Integer,1;
VARIABLE: c,1,3,Integer;
BUFFER: VC_Q,7,1000;
VARIABLE: Routes,2,5,5,Integer;
BUFFER: VCQ,7,1000;
VARIABLE: BDBTH,1,1,Real;
ATTRIBUTE: TStamp,2,Real,1;
MACHINE: TimerM,1,Single,0,0;
VARIABLE: VCQSize,1,3,Real;
VARIABLE: CLPHTH,1,3,Integer;
VARIABLE: CLPLTH,1,3,Integer;
ATTRIBUTE: CLP,1,Integer,1;
ATTRIBUTE: hops,1,Integer,1;
ATTRIBUTE: path,1,Integer,1;
VARIABLE: pathv,1,3,Integer;
PART: rtd,Variable attributes;
VARIABLE: rtdv,1,1,Real;
VARIABLE: ww,1,1,Real;
PART_FILE: tcpip,Read;
FILE: tcpipl,Read;
BUFFER: ciscobuf,3,1000;
TIMESERIES: applic,1;
VARIABLE: bbb,1,1,Real;
FILE: pcktsize,Read;
VARIABLE: bytesize,1,1,Integer;
VARIABLE: hopsv,1,3,Integer;

END DEFINE

REPORT_MODE ON_SHIFT_TIME GRAPHICAL STANDARD

DISPLAY
  OPTIONS
    TIME_SCALE_FACTOR : 1.00,On;
    WALK SPEED : 25;
    TIME INCREMENT : 1;
    BATCH INCREMENT : 5;
    FORM SIZE : Medium;
    SCREEN EDITOR SIZE : Medium;
    ICON EDITOR SIZE : Medium;
  END OPTIONS

  DEFAULTS
    NAME COLOR: White;
    BACKGROUND COLOR: 9;
    TEXT SIZE: Standard;
    PART DISPLAY SIZE: 1;
    LABOR DISPLAY SIZE: 1;
    VEHICLE DISPLAY SIZE: 1;
    CONVEYOR: GAPS: 96,....;
    TRACK: GAPS: 16,....;
    MACHINE: GAPS: 45,....;
    FONT: 8,0,0,0,400,0,0,0,0,0,0,0,0,0,WITNESS;
  END DEFAULTS

  CLOCK
    UNIT : :;

```

```

MULTIPLE : 1,millisecond,1000,0;
MULTIPLE : 2,sec,60,0;
MULTIPLE : 3,min,60,0;
RATIO : 1:1;
DISPLAY : REGULAR;
END CLOCK

WINDOW_TITLES
  TITLE : 1,Layout schematic
  TITLE : 2,Butsts - Fast Packets in System
  TITLE : 3,Process View
  TITLE : 4,Foresight credit manager
  TITLE : 5,Lost Packets
  TITLE : 6,Utilisation
  TITLE : 7,Georg Mueller
  TITLE : 8,Window 8
  TITLE : 9,Designer Elements
  TITLE : 10,Designer Elements Display
  TITLE : 11,Interact Box
  TITLE : 12,Clock
  TITLE : 13,Time
END WINDOW_TITLES

LAYER_STATUS
  LAYER : 0,On,Movable,Simulation
  LAYER : 1,On,Movable,Layer 1
  LAYER : 2,Off,Movable,Layer 2
  LAYER : 3,On,Movable,Layer 3
  LAYER : 4,Off,Movable,Layer 4
  LAYER : 5,Off,Movable,Layer 5
  LAYER : 6,Off,Fixed,Layer 6
  LAYER : 7,Off,Fixed,Layer 7
  LAYER : 8,On,Movable,Layer 8
  LAYER : 9,On,Fixed,Layer 9
  LAYER : 14,On,Movable,Layer
END LAYER_STATUS

BAR_SELECTOR_POSITION : 553,27,80,247;

LIST_SELECTION_FORM_POSITION : 392,154;

SELECT

  BACKDROP

    TEXT:
#0,Eff_Util,Group,2177,877,RGB(0,255,0),RGB(127,127,127),2,8,400,0,WITNESS,Effective Utilisation;
    TEXT:
#0,Utilisation,Group,2180,908,RGB(255,0,0),RGB(127,127,127),2,8,400,0,WITNESS,Utilisation;
    TEXT:
#0,Packetloss,Group,2175,1123,RGB(255,0,0),RGB(127,127,127),2,8,400,0,WITNESS,Lost Packets per Sec;
    KEY_MACHINE: #5,Key,Group,Standard,RGB(255,255,255),RGB(127,127,127),96,1480;
    Icon: #0,GeorgMuellerIcon,Group,RGB(255,255,255),107,1040,32,1,1,0,0;
    LOCK: On;

  END BACKDROP

  NONE

  LOCK: Off;

```

```
END NONE

ROUTE

LOCK: Off;

END ROUTE

WORLD

LOCK: Off;

END WORLD

SHIP

LOCK: Off;

END SHIP

SCRAP

LOCK: Off;

END SCRAP

ASSEMBLE

LOCK: Off;

END ASSEMBLE

TIME

LOCK: Off;

END TIME

Framer

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(0,0,0),2,344,72,400,0,WITNESS;
MACHINE ICON: #0,Icon,Member,RGB(255,255,0),106,336,104,1,1,0,0
                336,184,1,1,0,0
                336,264,1,1,0,0
                335,342,1,1,0,0
                335,419,1,1,0,0
                335,496,1,1,0,0
                336,576,1,1,0,0;

PART: #0,Part
Queue,Member,Count,RGB(255,255,255),RGB(0,0,0),0,8,1,65535,368,168,2
    368,248
    368,328
    367,406
    367,483
    367,560
    368,640;

LOCK: On;
VIEW: 2,0,0,0;

END Framer

Test
```

```

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,648,120,400,0,WITNESS;
BUFFER ICON: #0,Icon,Member,Status,9,656,136,1,1,0,0
              656,184,1,1,0,0
              656,232,1,1,0,0
              656,280,1,1,0,0;

PART: #0,Part
Queue,Member,Count,RGB(255,255,255),RGB(0,0,0),0,8,4,65535,640,160,2
      640,208
      640,256
      640,304;

LOCK: On;

END Test

Trunk

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(0,0,0),2,528,96,400,0,WITNESS;
PART: #0,Part Queue,Member,Count,RGB(255,255,255),RGB(0,0,0),0,8,1,All,546,160,2
      545,241
      547,312
      547,385
      546,458
      545,531
      544,604;
MACHINE ICON: #0,Icon,Member,RGB(255,255,255),114,520,112,1,1,0,0
              520,192,1,1,0,0
              520,264,1,1,0,0
              520,336,1,1,0,0
              520,408,1,1,0,0
              520,480,1,1,0,0
              520,552,1,1,0,0;

LOCK: On;

END Trunk

LostPckt

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(0,0,0),1,536,848,400,0,WITNESS;
BUFFER ICON: #0,Icon,Member,Status,10,560,880,1,1,0,0;
PART: #0,Part Queue,Member,Count,RGB(255,255,255),RGB(0,0,0),0,8,3,All,552,904,1;
LOCK: On;

END LostPckt

Frame

LOCK: Off;
STYLE: Desc,3,Fram;

END Frame

Bytes

LOCK: Off;

END Bytes

OK

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(0,0,0),1,784,96,400,0,WITNESS;
BUFFER ICON: #0,Icon,Member,Status,9,784,112,1,1,0,0
              784,168,1,1,0,0;

```

```

PART: #0,Part Queue,Member,Count,RGB(255,255,255),RGB(0,0,0),0,8,3,All,776,136,1
      776,192;
LOCK: Off;

END OK

fpbytes

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(0,0,0),1,80,864,400,0,WITNESS;
VALUES: #0,Variable,Group,Standard,RGB(255,255,255),RGB(0,0,0),4,240,872,40,0,0,1
        0,0,0;
LOCK: Off;

END fpbytes

NoFrames

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),1,80,912,400,0,WITNESS;
VALUES: #0,Variable,Group,Standard,RGB(255,255,255),RGB(0,0,0),6,224,920,56,0,0,1
        0,0,0;
LOCK: Off;

END NoFrames

Bursts

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(0,0,0),1,568,1792,400,0,WITNESS;
TIMESERIES DISPLAY:
#0,Timeseries,Member,RGB(0,0,0),RGB(255,0,0),RGB(0,255,0),RGB(0,0,255),RGB(0,255,25
5),RGB(255,255,0),RGB(255,0,255),RGB(192,192,192),RGB(255,255,255),Standard
      712,1800,30,12,0.00,1000.00,0.04,8,5,0,10,2,0;
LOCK: On;

END Bursts

indx

LOCK: Off;

END indx

A

LOCK: Off;

END A

Gen1

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),1,168,88,400,0,WITNESS;
PART: #0,Part Queue,Member,Count,RGB(255,255,255),RGB(0,0,0),0,8,1,All,184,168,2;
MACHINE ICON: #0,Icon,Member,RGB(255,255,255),111,160,112,1,1,0,0;
LOCK: On;

END Gen1

INX

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),1,80,888,400,0,WITNESS;
VALUES: #0,Variable,Group,Standard,RGB(255,255,255),RGB(0,0,0),6,224,896,56,0,0,1
        0,0,0;
LOCK: Off;

```

```

END INX

lenght

LOCK: Off;

END lenght

lenghta

LOCK: Off;

END lenghta

FlowCont

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(0,0,0),1,456,848,400,0,WITNESS;
MACHINE ICON: #0,Icon,Member,Status,97,472,872,1,1,2,0;
PART: #0,Part Queue,Member,Count,RGB(255,255,255),RGB(0,0,0),0,8,4,All,462,901,1;
LOCK: On;

END FlowCont

Lostpakt

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),1,80,960,400,0,WITNESS;
VALUES: #0,Variable,Group,Standard,RGB(255,255,255),RGB(0,0,0),6,224,968,56,0,0,1
        0,0,0;

LOCK: Off;

END Lostpakt

Utilise

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),1,672,960,400,0,WITNESS;
TIMESERIES DISPLAY:
#0,Timeseries,Member,RGB(0,0,0),RGB(255,0,0),RGB(0,255,0),RGB(0,0,255),RGB(0,255,25
5),RGB(255,255,0),RGB(255,0,255),RGB(192,192,192),RGB(255,255,255),Standard
        664,816,30,10,0.00,100.00,0.04,15,10,2,10,2,0;

LOCK: Off;

END Utilise

LostByte

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),1,80,984,400,0,WITNESS;
VALUES:
#0,Variable,Group,Standard,RGB(255,255,255),RGB(0,0,0),8,0,208,992,72,0,0,1
        0,0,0;

LOCK: Off;

END LostByte

Gen2

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),1,168,192,400,0,WITNESS;
MACHINE ICON: #0,Icon,Member,RGB(0,255,0),113,160,208,1,1,0,0;
PART: #0,Part Queue,Member,Count,RGB(255,255,255),RGB(0,0,0),0,8,1,All,186,264,2;
LOCK: On;

END Gen2

lambda

```

```

LOCK: Off;

END lambda

lambda1

TEXT:
#0,lambda1,Group,80,1032,RGB(255,255,255),RGB(127,127,127),1,8,400,0,WITNESS,Frame
Arrival;
EXPRESSION:
#0,Expression,Group,8,RGB(255,255,255),RGB(127,127,127),232,1032,2,0,400,0,WITNESS,
1,0,1.00,1 / lambda1 (1) * 8 * 300;
NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,80,1128,400,0,WITNESS;
VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),7,5,216,1136,64,0,0,1
0,0,0;
LOCK: Off;

END lambda1

Byte

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),1,80,934,400,0,WITNESS;
VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),7,216,944,64,0,0,1
0,0,0;
LOCK: Off;

END Byte

bimodal

LOCK: Off;

END bimodal

bimod

LOCK: Off;

END bimod

tspeed

EXPRESSION: #0,Trunk
Speed,Group,8,RGB(255,255,255),RGB(127,127,127),232,1008,2,0,400,0,WITNESS,1,0,1.00
,1 / tspeed * 8 * 24;
TEXT:
#0,trunk_speed,Group,80,1008,RGB(255,255,255),RGB(127,127,127),1,8,400,0,WITNESS,Tr
unk Speed;
LOCK: Off;

END tspeed

Packloss

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),1,478,1083,400,0,WITNESS;
TIMESERIES DISPLAY:
#0,Timeseries,Member,RGB(255,0,0),RGB(255,0,0),RGB(0,255,0),RGB(0,0,255),RGB(0,255,
255),RGB(255,255,0),RGB(255,0,255),RGB(192,192,192),RGB(255,255,255),Standard
672,2184,30,6,0.00,300.00,0.04,1,10,2,10,2,0;
LOCK: Off;

END Packloss

```

Gen3

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),1,168,296,400,0,WITNESS;
MACHINE ICON: #0,Icon,Member,RGB(255,255,255),112,160,312,1,1,0,0;
PART: #0,Part
Queue,Member,Count,RGB(255,255,255),RGB(0,0,0),0,8,1,65535,181,367,2;
LOCK: On;

END Gen3

mydistr

LOCK: Off;

END mydistr

distr

LOCK: Off;

END distr

testdist

LOCK: Off;

END testdist

myd1

LOCK: Off;

END myd1

myd2

LOCK: Off;

END myd2

pspeed

LOCK: Off;

END pspeed

gate

LOCK: Off;

END gate

b

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,80,1168,400,0,WITNESS;
VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),4,224,1176,0,0,0,1
0,0,0;

LOCK: Off;

END b

QIR


```

LOCK: Off;

END QIR

CIR

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,88,1360,400,0,WITNESS;
VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),7,5,176,1368,88,0,0,1
        0,0,0;
LOCK: Off;

END CIR

PIR

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,88,1336,400,0,WITNESS;
VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),7,5,178,1345,90,0,0,1
        0,0,0;
LOCK: Off;

END PIR

MIR

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,88,1312,400,0,WITNESS;
VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),7,5,178,1316,89,-
1,0,1
        0,0,0;
LOCK: Off;

END MIR

Util

LOCK: Off;

END Util

teta

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,80,1056,400,0,WITNESS;
VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),8,4,208,1064,72,0,0,1
        0,0,0;
LOCK: Off;

END teta

BD_B

TEXT:
#0,Text,Group,432,96,RGB(255,255,255),RGB(127,127,127),2,8,400,0,WITNESS,BD-B
Queue;
  BUFFER ICON: #0,Icon,Member,RGB(255,255,255),116,448,112,1,1,0,0
                448,192,1,1,0,0
                448,264,1,1,0,0
                448,336,1,1,0,0
                448,408,1,1,0,0
                448,480,1,1,0,0
                448,552,1,1,0,0;

PART: #0,Part
Queue,Member,Count,RGB(255,255,255),RGB(0,0,0),0,8,4,65535,456,160,2
        456,240

```

```

456,312
456,384
456,456
456,528
456,600;
LOCK: On;

END BD_B

steps

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,80,1080,400,0,WITNESS;
VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),7,5,216,1088,64,0,0,1
0,0,0;
LOCK: Off;

END steps

CManager

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,208,1216,400,0,WITNESS;
MACHINE ICON: #0,Icon,Member,RGB(255,255,255),118,152,1232,1,1,0,0
248,1232,1,1,0,0
352,1232,1,1,0,0;
LOCK: On;

END CManager

Cmax

LOCK: Off;

END Cmax

Credit

LOCK: Off;
STYLE: Desc,3,Cred;

END Credit

Credits

VALUES: #0,Value,Group,Standard,RGB(0,0,0),RGB(192,192,192),3,176,1272,-101,-
101,1,1
0,0,0
272,1272
368,1272;
LOCK: Off;

END Credits

CMR

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,80,1104,400,0,WITNESS;
VALUES:
#0,Value,Group,Standard,RGB(255,255,255),RGB(192,192,192),7,5,216,1112,64,0,0,1
0,0,0;
LOCK: Off;

END CMR

speed

```

```

TIMESERIES DISPLAY:
#0,Timeseries,Member,RGB(0,0,0),RGB(255,0,0),RGB(0,255,0),RGB(0,0,255),RGB(0,255,25
5),RGB(255,255,0),RGB(255,0,255),RGB(192,192,192),RGB(255,255,255),Standard
        672,1264,30,20,4000.00,65000.00,0.04,7,10,0,6,2,0;
LOCK: Off;
VIEW: 0,16,32,0;

END speed

timer

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,48,1216,400,0,WITNESS;
Icon: #0,Icon,Group,RGB(255,255,255),119,64,1232,1,1,0,0;
LOCK: On;
STYLE: Desc,3,t;

END timer

EFCN

LOCK: Off;

END EFCN

EFCNBITS

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,80,1152,400,0,WITNESS;
VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),7,3,200,1160,80,0,0,1
        0,0,0;
LOCK: Off;

END EFCNBITS

arrived

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,80,1192,400,0,WITNESS;
VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),4,205,1188,-101,-
101,1,1
        0,0,0
        284,1188
        361,1188;
LOCK: Off;

END arrived

OnOff

LOCK: Off;

END OnOff

portid

LOCK: Off;

END portid

c

LOCK: Off;

END c

```

VC_Q

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,288,104,400,0,WITNESS;

BUFFER ICON: #0,Icon,Member,RGB(255,255,255),117,280,120,1,1,0,0
 280,192,1,1,0,0
 280,280,1,1,0,0
 280,352,1,1,0,0
 280,424,1,1,0,0
 280,496,1,1,0,0
 280,568,1,1,0,0;

PART: #0,Part

Queue,Member,Count,RGB(255,255,255),RGB(0,0,0),0,8,4,65535,264,144,2
 264,216
 264,304
 264,376
 264,448
 264,520
 264,592;

LOCK: On;

END VC_Q

Routes

VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),4,800,480,40,56,0,2
 0,0,0
 0,0,0;

LOCK: Off;

END Routes

VCQ

BUFFER ICON: #0,Icon,Member,RGB(255,255,255),115,304,120,1,1,0,0
 304,192,1,1,0,0
 304,280,1,1,0,0
 304,352,1,1,0,0
 304,424,1,1,0,0
 304,496,1,1,0,0
 304,568,1,1,0,0;

PART: #0,Part

Queue,Member,Count,RGB(255,255,255),RGB(0,0,0),0,8,2,65535,304,144,2
 304,216
 304,304
 304,376
 304,448
 304,520
 304,592;

LOCK: On;

END VCQ

BDBTH

LOCK: Off;

END BDBTH

TStamp

LOCK: Off;

END TStamp

TimerM

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,640,1256,400,0,WITNESS;
LOCK: On;

END TimerM

VCQSize

NAME: #0,Name,Group,8,RGB(255,255,255),RGB(127,127,127),2,88,1384,400,0,WITNESS;
VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),5,0,168,1392,88,0,0,1
0,0,0;

LOCK: Off;

END VCQSize

CLPHTH

LOCK: Off;

END CLPHTH

CLPLTH

LOCK: Off;

END CLPLTH

CLP

LOCK: Off;

END CLP

hops

LOCK: Off;

END hops

path

LOCK: Off;

END path

pathv

LOCK: Off;

END pathv

rtd

LOCK: Off;
STYLE: Desc,3,rtd;

END rtd

rtdv

```

VALUES: #0, Value, Group, Standard, RGB(0,0,0), RGB(192,192,192), 5, 3, 80, 1280, 0, 0, 0, 1
        0, 0, 0;
LOCK: Off;

END rtdv

ww

LOCK: Off;

END ww

tcpip

NAME: #0, Name, Group, 8, RGB(255,255,255), RGB(127,127,127), 2, 24, 120, 400, 0, WITNESS;
LOCK: Off;

END tcpip

tcpipl

LOCK: Off;

END tcpipl

ciscobuf

TEXT:
#0, Buffer, Group, 156, 446, RGB(255,255,255), RGB(127,127,127), 2, 8, 400, 0, WITNESS, ;
TEXT:
#0, Text, Group, 96, 96, RGB(255,255,255), RGB(127,127,127), 2, 8, 400, 0, WITNESS, Buffer;
  BUFFER ICON: #0, Icon, Member, RGB(255,255,255), 117, 112, 120, 1, 1, 0, 0
                108, 222, 1, 1, 0, 0
                108, 326, 1, 1, 0, 0;

PART: #0, Part
Queue, Member, Count, RGB(255,255,255), RGB(0,0,0), 0, 8, 3, 65535, 109, 144, 2
        104, 246
        104, 352;
LOCK: On;

END ciscobuf

applic

NAME:
#0, Name, Group, 8, RGB(255,255,255), RGB(127,127,127), 2, 1296, 2480, 400, 0, WITNESS;
TIMESERIES DISPLAY:
#0, Timeseries, Member, RGB(255,255,255), RGB(255,0,0), RGB(0,255,0), RGB(0,0,255), RGB(0,
255,255), RGB(255,255,0), RGB(255,0,255), RGB(192,192,192), RGB(192,192,192), Standard
        928, 2416, 32, 20, 0.00, 20000.00, 0.10, 1, 10, 0, 10, 1, 0;
LOCK: Off;

END applic

bbb

LOCK: Off;

END bbb

pcktsize

LOCK: Off;

```

```

END pcktsize

bytesize

VALUES: #0,Value,Group,Standard,RGB(255,255,255),RGB(0,0,0),4,304,168,0,0,0,1
        0,0,0;
LOCK: Off;

END bytesize

hopsv

LOCK: Off;

END hopsv

END SELECT

END DISPLAY

DETAIL
  OPTIONS
    BREAKDOWN MODEL: Actual;
    REPAIR MODEL: Actual;
    LABOR TO UNLOAD : No;
    WARMUP PERIOD : 5.00;
    OUTPUT INTERVAL : None;
    AUTO SIM SAVE :
Off,1000.00,Simulation,Untit.sim,999,0,Display,Discard,Untit.sim;
    UNBLOCK BASIS : Priority;
    MONITOR STEP : Undefined;
    MIXTURE STEP : Undefined;
    MODULE ELEMENT NAMES : Use local preferences;
  END OPTIONS

SELECT

  Framer

  NAME OF MACHINE: Framer;
  QUANTITY: 7;
  TYPE: Production;
  * Part type: Same;
  * Production qty: 1;
  PRIORITY: 1;
  LABOR:
    Cycle: None;
  END
  DISCRETE LINKS :
    Fill: None
  END
  DISCRETE LINKS :
    Empty: None
  END
  CYCLE TIME: pspeed (N);
  ACTIONS, Start
  Add
    lenght (N) = lenghta
    fbytes (N) = Bytes
    INX (N) = indx
    pathv (N) = path
    hopsv (N) = hops

```

```

End Actions
ACTIONS, Finish
Add
  IF M = 1
    Bytes = MIN (20,fpbytes (N))
    fpbytes (N) = fpbytes (N) - Bytes
  ELSE
    Bytes = fpbytes (N)
    indx = INX (N)
    lenghta = lenght (N)
    portid = N
    path = pathv (N)
    hops = hopsv (N)
    TStamp (1) = TIME
    IF VCQSize (N) > 48000
      EFCN = 1
    ELSE
      EFCN = 0
    ENDIF
  ENDIF
  IF CMR (N) <= CIR (N)
    CLP = 1
  ELSE
    CLP = 0
  ENDIF
End Actions
INPUT RULE: PULL from VCQ(N),VC_Q(N);
OUTPUT RULE: IF Bytes > 20
  PUSH to VCQ(N)
ELSEIF Bytes > 0 AND Credits (N) > 0 AND OnOff = 1
  IF CLP = 1 AND CLPHTH (N) > 450
    PUSH to LostPckt
  ELSE
    PUSH to BD_B(Routes (path,hops))
  ENDIF
ELSEIF Bytes > 0 AND Credits (N) = 0 AND OnOff = 1
  Wait
ELSEIF Bytes > 0 AND OnOff = 0
  PUSH to BD_B,LostPckt
ELSE
  PUSH to SHIP
ENDIF;
REPORTING: Individual;
SHIFT: Undefined,0,0;

END Framer

Test

NAME OF BUFFER: Test;
QUANTITY: 4;
CAPACITY: 30000;
DELAY MODE : None;
INPUT POSITION: Rear;
OUTPUT SCAN FROM: Front;
* Select: First;
REPORTING: Individual;
SHIFT: Undefined,0;

END Test

Trunk

```

```

NAME OF MACHINE: Trunk;
QUANTITY: 7;
TYPE: Single;
PRIORITY: Lowest;
LABOR:
    Cycle: None;
END
DISCRETE LINKS :
    Fill: None
END
DISCRETE LINKS :
    Empty: None
END
CYCLE TIME: tspeed;
ACTIONS, Start
Add
    IF OnOff = 1
        IF EFCN = 1
            arrived (portid) = arrived (portid) + 1
            EFCNBITS (portid) = EFCNBITS (portid) + 1
        ELSEIF EFCN = 0
            arrived (portid) = arrived (portid) + 1
        ENDIF
    ENDIF
End Actions
INPUT RULE: PULL from BD_B(N)
            !SEQUENCE /Next Frame out of BD_B(1)#(1),
            !BD_B(2)#(1),
            !BD_B(3)#(1);
OUTPUT RULE: IF Routes (portid,hops) = -1
            PUSH to SHIP
            ELSE
            PUSH to BD_B(Routes (path,hops))
            ENDIF
            !PUSH to SHIP
            !PUSH to OK;
REPORTING: Individual;
SHIFT: Undefined,0,0;

END Trunk

LostPckt

NAME OF BUFFER: LostPckt;
QUANTITY: 1;
CAPACITY: 10000;
DELAY MODE : None;
INPUT POSITION: Rear;
OUTPUT SCAN FROM: Front;
* Select: First;
ACTIONS, In
Add
    Lostpakt = Lostpakt + 1
    b = b + 1
End Actions
REPORTING: Individual;
SHIFT: Undefined,0;

END LostPckt

Frame

NAME OF PART: Frame;

```

```
TYPE: Variable attributes;
GROUP NUMBER: 1;
MAXIMUM ARRIVALS: 0;
ACTIONS, Create
Add
!Bytes = 300
!K = K + 1
!indx = K
End Actions
OUTPUT RULE: !PUSH to ROUTE,Pcktloss
Wait;
PART ROUTE: None
REPORTING: Yes;
CONTAINS FLUIDS: No;
SHIFT: Undefined;

END Frame

Bytes

NAME OF ATTRIBUTE: Bytes;
QUANTITY: 1;

END Bytes

OK

NAME OF BUFFER: OK;
QUANTITY: 2;
CAPACITY: 10000;
DELAY MODE : None;
INPUT POSITION: Rear;
OUTPUT SCAN FROM: Front;
* Select: First;
REPORTING: Individual;
SHIFT: Undefined,0;

END OK

fpbytes

NAME OF VARIABLE: fpbytes;
QUANTITY: 3;
REPORTING: Yes;

END fpbytes

NoFrames

NAME OF VARIABLE: NoFrames;
QUANTITY: 3;
REPORTING: Yes;

END NoFrames

Bursts

NAME OF TIMESERIES: Bursts;
QUANTITY: 1;
RECORDING INTERVAL: 0.040000;
PLOT EXPRESSION : 1,Undefined;
PLOT EXPRESSION : 2,Undefined;
PLOT EXPRESSION : 3,Undefined;
```

```

PLOT EXPRESSION : 4,NWIP (Frame);
PLOT EXPRESSION : 5,Undefined;
PLOT EXPRESSION : 6,Undefined;
PLOT EXPRESSION : 7,Undefined;
STATS RESET : Yes;
SHIFT: Undefined;
REPEAT OPTION : No;

END Bursts

indx

NAME OF ATTRIBUTE: indx;
QUANTITY: 1;

END indx

A

NAME OF VARIABLE: A;
QUANTITY: 1;
REPORTING: Yes;

END A

Gen1

NAME OF MACHINE: Gen1;
QUANTITY: 1;
TYPE: Single;
PRIORITY: Lowest;
LABOR:
    Cycle: None;
END
DISCRETE LINKS :
    Fill: None
END
DISCRETE LINKS :
    Empty: None
END
CYCLE TIME: 0.0;
ACTIONS, Start
Add
!Bytes = NEGEXP (300,6)
!Bytes = bimod (11)
    lenghta = Bytes
    indx = NoFrames (1)
    NoFrames (1) = NoFrames (1) + 1
    Byte = Byte + Bytes
End Actions
ACTIONS, Finish
Add
    INX (1) = indx
    portid = 1
    EFCN = 0
    path = 1
    hops = 1
End Actions
INPUT RULE: !Wait
    PULL from Frame out of ciscobuf;
OUTPUT RULE: PUSH to VC_Q(1),LostPckt
    !PUSH to Test(1)
    !PUSH to SHIP;

```

```

REPORTING: Individual;
SHIFT: Undefined,0,0;

END Gen1

INX

NAME OF VARIABLE: INX;
QUANTITY: 3;
REPORTING: Yes;

END INX

lenght

NAME OF VARIABLE: lenght;
QUANTITY: 3;
REPORTING: Yes;

END lenght

lenghta

NAME OF ATTRIBUTE: lenghta;
QUANTITY: 1;

END lenghta

FlowCont

NAME OF MACHINE: FlowCont;
QUANTITY: 1;
TYPE: Batch;
* Batch min: 1;
* Batch max: 100;
PRIORITY: Lowest;
LABOR:
  Cycle: None;
END
DISCRETE LINKS :
  Fill: None
END
DISCRETE LINKS :
  Empty: None
END
CYCLE TIME: pspeed (N) * 2;
ACTIONS, Start
Add
  IF indx > A
    Bytes = lenghta
  ELSE
    Bytes = 0
  ENDIF
  A = indx
End Actions
ACTIONS, Finish
Add
  LostByte = LostByte + Bytes
End Actions
INPUT RULE: !PULL from OK
          PULL from LostPckt;
OUTPUT RULE: !PUSH to pktizer
          PUSH to SHIP;

```

```

REPORTING: Individual;
SHIFT: Undefined,0,0;

END FlowCont

Lostpakt

NAME OF VARIABLE: Lostpakt;
QUANTITY: 1;
REPORTING: Yes;

END Lostpakt

Utilise

NAME OF TIMESERIES: Utilise;
QUANTITY: 1;
RECORDING INTERVAL: 0.040000;
PLOT EXPRESSION : 1,PUTIL (Trunk(1),2);
PLOT EXPRESSION : 2,PUTIL (Trunk(2),2);
PLOT EXPRESSION : 3,PUTIL (Trunk(3),2);
PLOT EXPRESSION : 4,PUTIL (Trunk(4),2);
PLOT EXPRESSION : 5,Undefined;
PLOT EXPRESSION : 6,Undefined;
PLOT EXPRESSION : 7,Undefined;
STATS RESET : No;
SHIFT: Undefined;
REPEAT OPTION : No;

END Utilise

LostByte

NAME OF VARIABLE: LostByte;
QUANTITY: 1;
REPORTING: Yes;

END LostByte

Gen2

NAME OF MACHINE: Gen2;
QUANTITY: 1;
TYPE: Single;
PRIORITY: Lowest;
LABOR:
    Cycle: None;
END
DISCRETE LINKS :
    Fill: None
END
DISCRETE LINKS :
    Empty: None
END
CYCLE TIME: POISSON (lambda1 (2),2);
ACTIONS, Start
Add
    Bytes = 300
!Bytes = bimod (7)
    indx = NoFrames (2)
    NoFrames (2) = NoFrames (2) + 1
    Byte = Byte + Bytes
End Actions

```

```

ACTIONS, Finish
Add
  INX (2) = indx
  lenght (2) = Bytes
  portid = 2
  EFCN = 0
  path = 2
  hops = 1
End Actions
INPUT RULE: PULL from Frame out of WORLD;
OUTPUT RULE: PUSH to VC_Q(2),LostPckt
              !PUSH to Test(2);
REPORTING: Individual;
SHIFT: Undefined,0,0;

END Gen2

lambda

NAME OF VARIABLE: lambda;
QUANTITY: 1;
REPORTING: Yes;

END lambda

lambda1

NAME OF VARIABLE: lambda1;
QUANTITY: 5;
REPORTING: Yes;

END lambda1

Byte

NAME OF VARIABLE: Byte;
QUANTITY: 1;
REPORTING: Yes;

END Byte

bimodal

NAME OF DISTRIBUTION: bimodal;
! Type: Integer,Discrete;
VALUES: 2
  64,5.100000
  1500,1.000000
END bimodal

bimod

NAME OF DISTRIBUTION: bimod;
! Type: Real,Discrete;
VALUES: 2
  64.000000,5.100000
  1500.000000,1.000000
END bimod

tspeed

NAME OF VARIABLE: tspeed;
QUANTITY: 1;

```

```

REPORTING: Yes;

END tspeed

Packloss

NAME OF TIMESERIES: Packloss;
QUANTITY: 1;
RECORDING INTERVAL: 0.040000;
PLOT EXPRESSION : 1,b;
PLOT EXPRESSION : 2,Undefined;
PLOT EXPRESSION : 3,Undefined;
PLOT EXPRESSION : 4,Undefined;
PLOT EXPRESSION : 5,Undefined;
PLOT EXPRESSION : 6,Undefined;
PLOT EXPRESSION : 7,Undefined;
STATS RESET : Yes;
ACTIONS, After
Add
    b = 0
End Actions
SHIFT: Undefined;
REPEAT OPTION : No;

END Packloss

Gen3

NAME OF MACHINE: Gen3;
QUANTITY: 1;
TYPE: Single;
PRIORITY: Lowest;
LABOR:
    Cycle: None;
END
DISCRETE LINKS :
    Fill: None
END
DISCRETE LINKS :
    Empty: None
END
CYCLE TIME: POISSON (lambda1 (3),12);
ACTIONS, Start
Add
!Bytes = NEGEXP (300,6)
    Bytes = bimod (10)
    indx = NoFrames (3)
    NoFrames (3) = NoFrames (3) + 1
    Byte = Byte + Bytes
End Actions
ACTIONS, Finish
Add
    INX (3) = indx
    lenght (3) = Bytes
    portid = 3
    EFCN = 0
    path = 3
    hops = 1
End Actions
INPUT RULE: PULL from Frame out of WORLD;
OUTPUT RULE: PUSH to VC_Q(3),LostPckt
    !PUSH to Test(3);
REPORTING: Individual;

```

```
SHIFT: Undefined,0,0;

END Gen3

mydistr

NAME OF DISTRIBUTION: mydistr;
! Type: Real,Discrete;
VALUES: 98
  0.001000,418.000000
  0.002000,253.000000
  0.003000,50.000000
  0.004000,11.000000
  0.005000,10.000000
  0.006000,5.000000
  0.007000,5.000000
  0.008000,16.000000
  0.009000,3.000000
  0.010000,4.000000
  0.011000,7.000000
  0.012000,9.000000
  0.013000,2.000000
  0.014000,1.000000
  0.015000,14.000000
  0.016000,17.000000
  0.017000,6.000000
  0.018000,2.000000
  0.019000,1.000000
  0.020000,11.000000
  0.021000,8.000000
  0.022000,13.000000
  0.023000,6.000000
  0.024000,6.000000
  0.025000,5.000000
  0.026000,4.000000
  0.027000,2.000000
  0.028000,6.000000
  0.029000,7.000000
  0.030000,5.000000
  0.031000,11.000000
  0.032000,7.000000
  0.033000,1.000000
  0.034000,7.000000
  0.035000,1.000000
  0.036000,3.000000
  0.037000,2.000000
  0.038000,2.000000
  0.039000,5.000000
  0.040000,1.000000
  0.041000,1.000000
  0.042000,4.000000
  0.043000,1.000000
  0.044000,2.000000
  0.045000,1.000000
  0.046000,6.000000
  0.047000,3.000000
  0.048000,1.000000
  0.049000,1.000000
  0.050000,5.000000
  0.051000,1.000000
  0.052000,0.000000
  0.053000,0.000000
  0.054000,1.000000
```

```
0.055000,0.000000
0.056000,1.000000
0.057000,1.000000
0.058000,0.000000
0.059000,1.000000
0.060000,0.000000
0.061000,0.000000
0.062000,1.000000
0.063000,1.000000
0.064000,0.000000
0.065000,1.000000
0.066000,0.000000
0.067000,1.000000
0.068000,1.000000
0.069000,0.000000
0.070000,1.000000
0.071000,1.000000
0.072000,0.000000
0.073000,0.000000
0.074000,1.000000
0.075000,2.000000
0.076000,0.000000
0.077000,1.000000
0.078000,0.000000
0.079000,0.000000
0.080000,0.000000
0.081000,3.000000
0.082000,0.000000
0.083000,1.000000
0.084000,0.000000
0.085000,0.000000
0.086000,1.000000
0.087000,0.000000
0.088000,0.000000
0.089000,0.000000
0.090000,1.000000
0.091000,0.000000
0.092000,1.000000
0.093000,0.000000
0.094000,1.000000
0.095000,1.000000
0.096000,1.000000
0.097000,1.000000
0.098000,1.000000
END mydistr

testdist

NAME OF DISTRIBUTION: testdist;
! Type: Integer,Discrete;
VALUES: 2
    1,10000.000000
    2,1.000000
END testdist

myd1

NAME OF DISTRIBUTION: myd1;
! Type: Real,Discrete;
VALUES: 98
    1.000000,418.000000
    2.000000,253.000000
    3.000000,50.000000
```

4.000000,11.000000
5.000000,10.000000
6.000000,5.000000
7.000000,5.000000
8.000000,16.000000
9.000000,3.000000
10.000000,4.000000
11.000000,7.000000
12.000000,9.000000
13.000000,2.000000
14.000000,1.000000
15.000000,14.000000
16.000000,17.000000
17.000000,6.000000
18.000000,2.000000
19.000000,1.000000
20.000000,11.000000
21.000000,8.000000
22.000000,13.000000
23.000000,6.000000
24.000000,6.000000
25.000000,5.000000
26.000000,4.000000
27.000000,2.000000
28.000000,6.000000
29.000000,7.000000
30.000000,5.000000
31.000000,11.000000
32.000000,7.000000
33.000000,1.000000
34.000000,7.000000
35.000000,1.000000
36.000000,3.000000
37.000000,2.000000
38.000000,2.000000
39.000000,5.000000
40.000000,1.000000
41.000000,1.000000
42.000000,4.000000
43.000000,1.000000
44.000000,2.000000
45.000000,1.000000
46.000000,6.000000
47.000000,3.000000
48.000000,1.000000
49.000000,1.000000
50.000000,5.000000
51.000000,1.000000
52.000000,0.000000
53.000000,0.000000
54.000000,1.000000
55.000000,0.000000
56.000000,1.000000
57.000000,1.000000
58.000000,0.000000
59.000000,1.000000
60.000000,0.000000
61.000000,0.000000
62.000000,1.000000
63.000000,1.000000
64.000000,0.000000
65.000000,1.000000
66.000000,0.000000

```
67.000000,1.000000
68.000000,1.000000
69.000000,0.000000
70.000000,1.000000
71.000000,1.000000
72.000000,0.000000
73.000000,0.000000
74.000000,1.000000
75.000000,2.000000
76.000000,0.000000
77.000000,1.000000
78.000000,0.000000
79.000000,0.000000
80.000000,0.000000
81.000000,3.000000
82.000000,0.000000
83.000000,1.000000
84.000000,0.000000
85.000000,0.000000
86.000000,1.000000
87.000000,0.000000
88.000000,0.000000
89.000000,0.000000
90.000000,1.000000
91.000000,0.000000
92.000000,1.000000
93.000000,0.000000
94.000000,1.000000
95.000000,1.000000
96.000000,1.000000
97.000000,1.000000
98.000000,1.000000
END myd1

myd2

NAME OF DISTRIBUTION: myd2;
! Type: Integer,Discrete;
VALUES: 12
0,450.000000
1,250.000000
2,53.000000
3,11.000000
4,11.000000
5,10.000000
6,11.000000
7,10.000000
8,8.000000
9,8.000000
10,5.000000
11,1.000000
END myd2

pspeed

NAME OF VARIABLE: pspeed;
QUANTITY: 3;
REPORTING: Yes;

END pspeed

gate
```

NAME OF VARIABLE: gate;
QUANTITY: 3;
REPORTING: Yes;

END gate

b

NAME OF VARIABLE: b;
QUANTITY: 1;
REPORTING: Yes;

END b

QIR

NAME OF VARIABLE: QIR;
QUANTITY: 3;
REPORTING: Yes;

END QIR

CIR

NAME OF VARIABLE: CIR;
QUANTITY: 3;
REPORTING: Yes;

END CIR

PIR

NAME OF VARIABLE: PIR;
QUANTITY: 3;
REPORTING: Yes;

END PIR

MIR

NAME OF VARIABLE: MIR;
QUANTITY: 3;
REPORTING: Yes;

END MIR

Util

NAME OF VARIABLE: Util;
QUANTITY: 3;
REPORTING: Yes;

END Util

teta

NAME OF VARIABLE: tetra;
QUANTITY: 3;
REPORTING: Yes;

END tetra

BD_B

```

NAME OF BUFFER: BD_B;
QUANTITY: 7;
CAPACITY: 600;
DELAY MODE : None;
INPUT POSITION: Rear;
OUTPUT SCAN FROM: Front;
* Select: First;
ACTIONS, In
Add
  IF Credits (portid) > 0 AND hops < 2
    Credits (portid) = Credits (portid) - 1
  ENDIF
  hops = hops + 1
End Actions
ACTIONS, Out
Add
  IF NPARTS (BD_B(N)) > BDBTH
    EFCN = 1
  ENDIF
End Actions
REPORTING: Individual;
SHIFT: Undefined,0;

END BD_B

steps

NAME OF VARIABLE: steps;
QUANTITY: 1;
REPORTING: Yes;

END steps

CManager

NAME OF MACHINE: CManager;
QUANTITY: 3;
TYPE: Single;
PRIORITY: Lowest;
LABOR:
  Cycle: None;
END
DISCRETE LINKS :
  Fill: None
END
DISCRETE LINKS :
  Empty: None
END
CYCLE TIME: CMR (N);
ACTIONS, Start
Add
  IF Credits (N) < Cmax (N)
    Credits (N) = Credits (N) + 1
  ENDIF
End Actions
INPUT RULE: PULL from Credit out of WORLD;
OUTPUT RULE: PUSH to SHIP;
REPORTING: Individual;
SHIFT: Undefined,0,0;

END CManager

```

Cmax

NAME OF VARIABLE: Cmax;
QUANTITY: 3;
REPORTING: Yes;

END Cmax

Credit

NAME OF PART: Credit;
TYPE: Variable attributes;
GROUP NUMBER: 1;
MAXIMUM ARRIVALS: 0;
OUTPUT RULE: Wait;
PART ROUTE: END
REPORTING: Yes;
CONTAINS FLUIDS: No;
SHIFT: Undefined;

END Credit

Credits

NAME OF VARIABLE: Credits;
QUANTITY: 3;
REPORTING: Yes;

END Credits

CMR

NAME OF VARIABLE: CMR;
QUANTITY: 3;
REPORTING: Yes;

END CMR

speed

NAME OF TIMESERIES: speed;
QUANTITY: 1;
RECORDING INTERVAL: 0.040000;
PLOT EXPRESSION : 1,1 / CMR (1) * ((23 + 1) * 8);
PLOT EXPRESSION : 2,1 / CMR (2) * ((23 + 1) * 8);
PLOT EXPRESSION : 3,1 / CMR (3) * ((23 + 1) * 8);
PLOT EXPRESSION : 4,Undefined;
PLOT EXPRESSION : 5,Undefined;
PLOT EXPRESSION : 6,Undefined;
PLOT EXPRESSION : 7,Undefined;
STATS RESET : No;
SHIFT: Undefined;
REPEAT OPTION : No;

END speed

timer

NAME OF PART: timer;
TYPE: Variable attributes;
GROUP NUMBER: 1;
MAXIMUM ARRIVALS: Unlimited;
INTER ARRIVAL TIME: rtdv;

```

FIRST ARRIVAL AT: 0.04;
LOT SIZE: 1;
ACTIONS, Leave
Add
  FOR b = 1 TO 3
    IF arrived (b) > 0
      EFCNBITS (b) = EFCNBITS (b) / arrived (b)
      IF EFCNBITS (b) >= 0.5 AND CMR (b) < MIR (b)
        CMR (b) = 1 / CMR (b) * ((23 + 1) * 8)
        CMR (b) = CMR (b) - CMR (b) * 13 / 100
        CMR (b) = 1 / CMR (b) * 8 * 24
      ELSEIF EFCNBITS (b) <= 0.5 AND CMR (b) > PIR (b)
        CMR (b) = 1 / CMR (b) * ((23 + 1) * 8)
        CMR (b) = CMR (b) + 800
        CMR (b) = 1 / CMR (b) * 8 * 24
      ENDIF
    !reset variables
    ENDIF
    arrived (b) = 0
    EFCNBITS (b) = 0
  NEXT
End Actions
OUTPUT RULE: Wait;
PART ROUTE: END
REPORTING: Yes;
CONTAINS FLUIDS: No;
SHIFT: Undefined;

END timer

EFCN

NAME OF ATTRIBUTE: EFCN;
QUANTITY: 1;

END EFCN

EFCNBITS

NAME OF VARIABLE: EFCNBITS;
QUANTITY: 3;
REPORTING: Yes;

END EFCNBITS

arrived

NAME OF VARIABLE: arrived;
QUANTITY: 3;
REPORTING: Yes;

END arrived

OnOff

NAME OF VARIABLE: OnOff;
QUANTITY: 1;
REPORTING: Yes;

END OnOff

portid

```

```

NAME OF ATTRIBUTE: portid;
QUANTITY: 1;

END portid

c

NAME OF VARIABLE: c;
QUANTITY: 3;
REPORTING: Yes;

END c

VC_Q

NAME OF BUFFER: VC_Q;
QUANTITY: 7;
CAPACITY: 1000;
DELAY MODE : None;
INPUT POSITION: Rear;
OUTPUT SCAN FROM: Front;
* Select: First;
ACTIONS, In
Add
    TStamp (1) = TIME
    VCQSize (N) = VCQSize (N) + Bytes
End Actions
ACTIONS, Out
Add
    VCQSize (N) = VCQSize (N) - Bytes
End Actions
REPORTING: Individual;
SHIFT: Undefined,0;

END VC_Q

Routes

NAME OF VARIABLE: Routes;
QUANTITY: 5,5;
REPORTING: Yes;

END Routes

VCQ

NAME OF BUFFER: VCQ;
QUANTITY: 7;
CAPACITY: 1000;
DELAY MODE : None;
INPUT POSITION: Rear;
OUTPUT SCAN FROM: Front;
* Select: First;
ACTIONS, In
Add
    bytesize = Bytes
End Actions
REPORTING: Individual;
SHIFT: Undefined,0;

END VCQ

BDBTH

```

```
NAME OF VARIABLE: BDBTH;
QUANTITY: 1;
REPORTING: Yes;

END BDBTH

TStamp

NAME OF ATTRIBUTE: TStamp;
QUANTITY: 2;

END TStamp

TimerM

NAME OF MACHINE: TimerM;
QUANTITY: 1;
TYPE: Single;
PRIORITY: Lowest;
LABOR:
    Cycle: None;
END
DISCRETE LINKS :
    Fill: None
END
DISCRETE LINKS :
    Empty: None
END
CYCLE TIME: 0.04;
INPUT RULE: Wait
    !PULL from timer out of WORLD;
OUTPUT RULE: PUSH to SHIP;
REPORTING: Individual;
SHIFT: Undefined,0,0;

END TimerM

VCQSize

NAME OF VARIABLE: VCQSize;
QUANTITY: 3;
REPORTING: Yes;

END VCQSize

CLPHTH

NAME OF VARIABLE: CLPHTH;
QUANTITY: 3;
REPORTING: Yes;

END CLPHTH

CLPLTH

NAME OF VARIABLE: CLPLTH;
QUANTITY: 3;
REPORTING: Yes;

END CLPLTH

CLP
```

```
NAME OF ATTRIBUTE: CLP;
QUANTITY: 1;

END CLP

hops

NAME OF ATTRIBUTE: hops;
QUANTITY: 1;

END hops

path

NAME OF ATTRIBUTE: path;
QUANTITY: 1;

END path

pathv

NAME OF VARIABLE: pathv;
QUANTITY: 3;
REPORTING: Yes;

END pathv

rtd

NAME OF PART: rtd;
TYPE: Variable attributes;
GROUP NUMBER: 1;
MAXIMUM ARRIVALS: 0;
OUTPUT RULE: Wait;
PART ROUTE: None
REPORTING: Yes;
CONTAINS FLUIDS: No;
SHIFT: Undefined;

END rtd

rtdv

NAME OF VARIABLE: rtdv;
QUANTITY: 1;
REPORTING: Yes;

END rtdv

ww

NAME OF VARIABLE: ww;
QUANTITY: 1;
REPORTING: Yes;

END ww

tcpip

NAME OF PART FILE: tcpip;
FILES ACTUAL NAME: tcpip1.par;
! TYPE: Read;
```

```
TIME: Absolute;
RESTART: Yes;
START OFFSET: 0.00;
DELAY: 0.00;
OUTPUT RULE: PUSH Frame to ciscobuf;
SHIFT: Undefined;

END tcpip

tcpip1

NAME OF FILE: tcpip1;
FILES ACTUAL NAME: tcpip1.dat;
| TYPE: Read;
RESTART: Yes;

END tcpip1

ciscobuf

NAME OF BUFFER: ciscobuf;
QUANTITY: 3;
CAPACITY: 1000;
DELAY MODE : None;
INPUT POSITION: Rear;
OUTPUT SCAN FROM: Front;
* Select: First;
ACTIONS, In
Add
    bbb = bbb + bimod (11)
    READ pktsize Bytes
End Actions
REPORTING: Individual;
SHIFT: Undefined,0;

END ciscobuf

applic

NAME OF TIMESERIES: applic;
QUANTITY: 1;
RECORDING INTERVAL: 0.100000;
PLOT EXPRESSION : 1,bbb;
PLOT EXPRESSION : 2,Undefined;
PLOT EXPRESSION : 3,Undefined;
PLOT EXPRESSION : 4,Undefined;
PLOT EXPRESSION : 5,Undefined;
PLOT EXPRESSION : 6,Undefined;
PLOT EXPRESSION : 7,Undefined;
STATS RESET : Yes;
ACTIONS, After
Add
    bbb = 0
End Actions
SHIFT: Undefined;
REPEAT OPTION : No;

END applic

bbb

NAME OF VARIABLE: bbb;
QUANTITY: 1;
```

```
REPORTING: Yes;

END bbb

pcktsize

NAME OF FILE: pcktsize;
FILES ACTUAL NAME: pcktsize.dat;
! TYPE: Read;
RESTART: Yes;

END pcktsize

bytesize

NAME OF VARIABLE: bytesize;
QUANTITY: 1;
REPORTING: Yes;

END bytesize

hopsv

NAME OF VARIABLE: hopsv;
QUANTITY: 3;
REPORTING: Yes;

END hopsv

END SELECT

END DETAIL

DEFINE_USER
Add
    gate (2) = 1000
End Actions
END DEFINE_USER

INITIALISE
Add
    Routes (1,1) = 1
    Routes (1,2) = 2
    Routes (1,3) = 3
    Routes (1,4) = -1
    Routes (2,1) = 1
    Routes (2,2) = 3
    Routes (2,3) = 4
    Routes (2,4) = -1
    Routes (3,1) = 1
    Routes (3,2) = 2
    Routes (3,3) = -1
    Routes (4,1) = 1
    Routes (4,2) = -1
    rtdv = 0.04
    FOR b = 1 TO 3
        CIR (b) = 64000
        QIR (b) = 64000
        MIR (b) = 8000
        PIR (b) = 256000
        Util (b) = 0.4
        tspeed = 512000
        lambda1 (b) = 64000
```

```
teta (b) = 0.04 * ((PIR (b) - MIR (b)) / (0.1 * MIR (b)))
Cmax (b) = 200
Credits (b) = 20
OnOff = 1
BDBTH = 25
!*****
steps = 1 / MIR (b) * 8 * 24 / 10
NoFrames (1) = 0
lambda = 4000
lambda1 (b) = 1 / lambda1 (b) * 8 * 300
tspeed = 1 / tspeed * 8 * 24
MIR (b) = 1 / MIR (b) * 8 * 24
PIR (b) = 1 / PIR (b) * 8 * 24
QIR (b) = 1 / QIR (b) * 8 * 24
CIR (b) = 1 / CIR (b) * 8 * 24
gate (b) = 1
pspeed (b) = 0.003
CMR (b) = CIR (b)
BDBTH = BDBTH * 600 / 100
NEXT
End Actions
END INITIALISE
```

Appendix B – Simulation Screens

Appendix B:

The following pages show some screen shots of the developed simulation. The first screen shot shows some of the components of the network.

The screen shows the traffic generators (Gen1, Gen2, Gen3) which feed into the access queues (VC_Q) of the port. The port feeds into BD-B queues of the switch which connects to the trunk. The connection between the trunks and the other components of the network cannot be displayed on the screen. Below the components the amount of Fixedpackets can be seen. E.g., it can be seen that the top icon resembling the BD-B queue has 36 Fixedpackets while one Fixedpacket is sent via the top icon resembling the trunk.

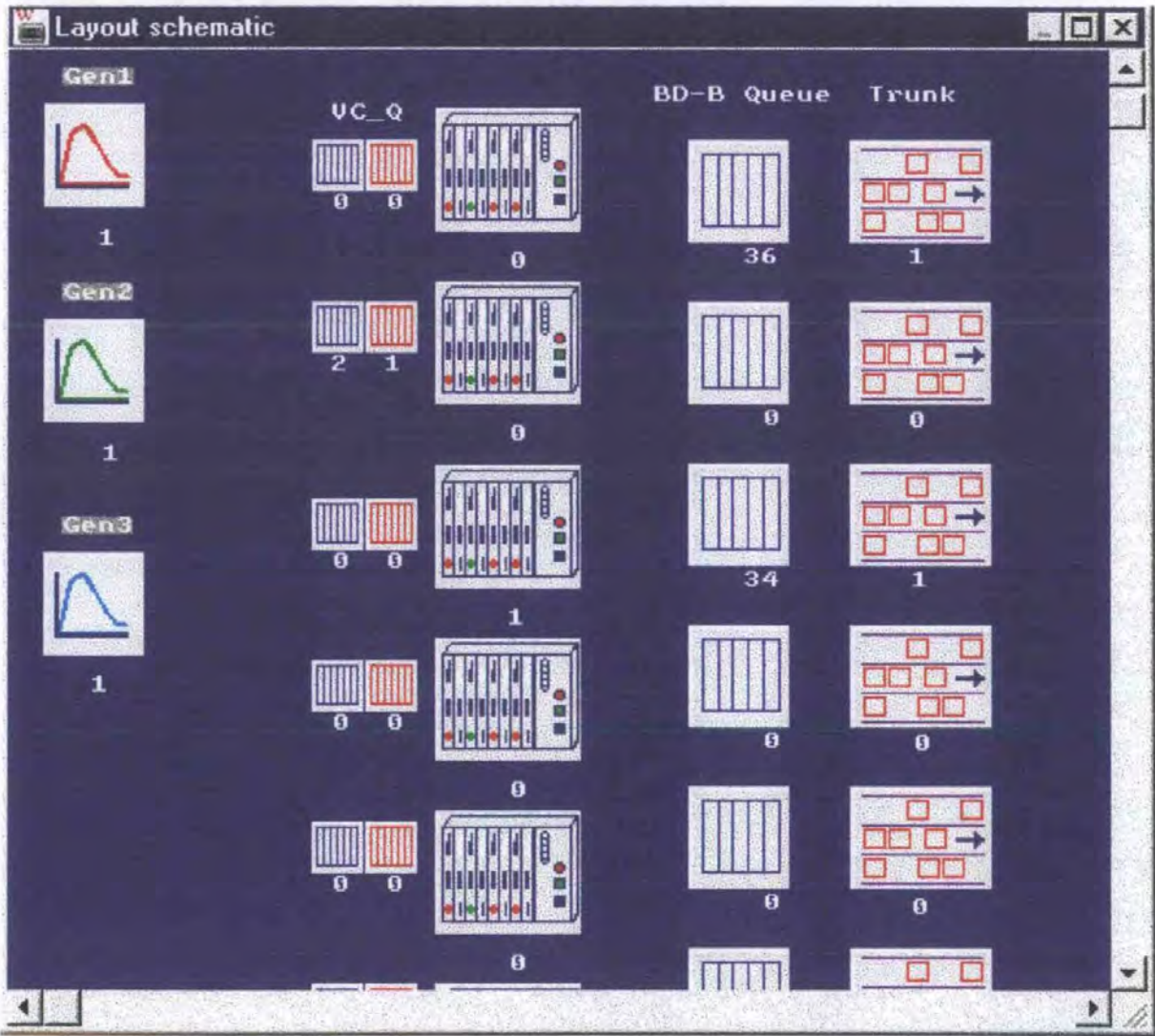


Figure B. 1: Screenshot of layout schematic

Figure B.2 shows a screen shot of three credit managers and the timer for the round trip delay (RTD). It also shows MIR and PIR settings for individual PVCs and the access queue size (VC_Q). This view allows real-time monitoring of the simulation and helps understand the individual PVC behaviours.

As indicated by the numbers in this example, the MIR threshold for one credit cycle is set at 24ms (equivalent of 41.66 credits per second), and the PIR threshold for one credit cycle is set at 3ms (equivalent to 333.33 credits per second).

The three credit manager icons show different numbers and represent the amount of credits created.

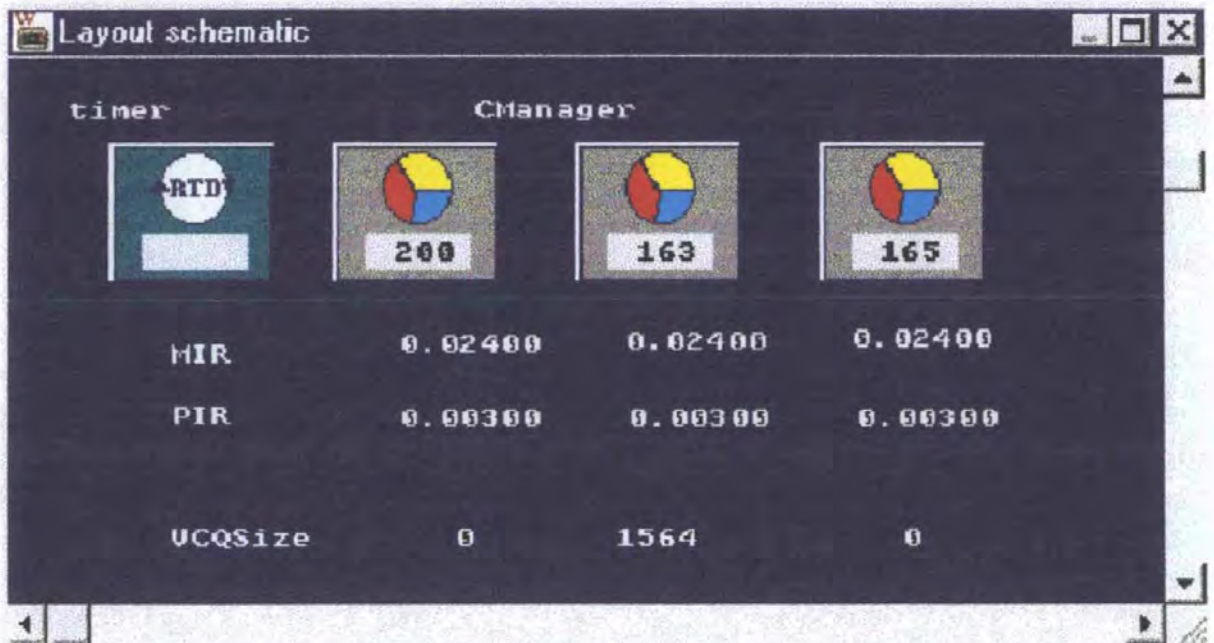


Figure B.2: Screenshot of Credit Manager simulation

Appendix C – Simulation Results

C.1 Delay, Goodput and Loss Results

Application A:

Simulation run 1	ITU-T	PM	PM - PTP
Delay in ms	324	324	324
Goodput in kbit/s	65.91	72.12	75.22
Loss at access point in %	3.22	3.01	2.91
Simulation run 2	ITU-T	PM	PM - PTP
Delay in ms	337	309	321
Goodput in kbit/s	67.57	72.22	76.57
Loss at access point in %	3.71	3.36	3.00
Simulation run 3	ITU-T	PM	PM - PTP
Delay in ms	325	334	324
Goodput in kbit/s	67.71	73.93	75.65
Loss at access point in %	3.36	3.05	3.05
Simulation run 4	ITU-T	PM	PM - PTP
Delay in ms	325	337	309
Goodput in kbit/s	67.36	73.36	76.82
Loss at access point in %	3.62	3.50	2.96
Simulation run 5	ITU-T	PM	PM - PTP
Delay in ms	322	323	325
Goodput in kbit/s	66.51	72.36	76.01
Loss at access point in %	3.60	3.28	2.99
Simulation run 6	ITU-T	PM	PM - PTP
Delay in ms	317	336	321
Goodput in kbit/s	66.64	72.21	76.63
Loss at access point in %	3.55	3.25	3.26
Simulation run 7	ITU-T	PM	PM - PTP
Delay in ms	326	313	335
Goodput in kbit/s	67.59	73.01	76.11
Loss at access point in %	3.25	3.03	3.05
Simulation run 8	ITU-T	PM	PM - PTP
Delay in ms	324	322	337
Goodput in kbit/s	66.12	72.60	77.09
Loss at access point in %	3.66	3.15	2.95

Simulation run 9	ITU-T	PM	PM - PTP
Delay in ms	333	336	318
Goodput in kbit/s	66.01	73.44	75.55
Loss at access point in %	3.25	3.20	3.26
Simulation run 10	ITU-T	PM	PM - PTP
Delay in ms	326	322	320
Goodput in kbit/s	67.17	73.17	76.22
Loss at access point in %	3.26	3.24	3.33
Consolidated numbers of all runs			
	ITU-T	PM	PM - PTP
Mean Delay in ms	326	326	323
Mean Goodput in kbit/s	66.86	72.84	76.19
Mean Loss at access point in %	3.45	3.21	3.07

Application B:

Simulation run 1	ITU-T	PM	PM - PTP
Delay in ms	240	237	226
Goodput in kbit/s	69.98	72.62	75.67
Loss at access point in %	8.96	8.99	7.32
Simulation run 2	ITU-T	PM	PM - PTP
Delay in ms	246	236	237
Goodput in kbit/s	70.28	73.87	77.31
Loss at access point in %	9.36	9.32	7.36
Simulation run 3	ITU-T	PM	PM - PTP
Delay in ms	235	230	212
Goodput in kbit/s	71.95	74.54	75.76
Loss at access point in %	9.25	9.30	7.43
Simulation run 4	ITU-T	PM	PM - PTP
Delay in ms	238	252	213
Goodput in kbit/s	71.80	73.03	76.41
Loss at access point in %	9.03	9.20	7.36
Simulation run 5	ITU-T	PM	PM - PTP
Delay in ms	248	233	214
Goodput in kbit/s	70.84	73.71	75.95
Loss at access point in %	9.31	9.47	7.61

Simulation run 6	ITU-T	PM	PM - PTP
Delay in ms	245	233	236
Goodput in kbit/s	70.81	73.00	77.34
Loss at access point in %	9.26	9.03	7.55
Simulation run 7	ITU-T	PM	PM - PTP
Delay in ms	239	225	230
Goodput in kbit/s	70.03	73.09	76.55
Loss at access point in %	9.01	9.10	7.51
Simulation run 8	ITU-T	PM	PM - PTP
Delay in ms	253	244	237
Goodput in kbit/s	70.18	73.56	75.82
Loss at access point in %	9.20	9.15	7.67
Simulation run 9	ITU-T	PM	PM - PTP
Delay in ms	233	240	236
Goodput in kbit/s	70.67	72.96	76.38
Loss at access point in %	9.34	9.46	7.66
Simulation run 10	ITU-T	PM	PM - PTP
Delay in ms	247	242	224
Goodput in kbit/s	70.00	73.24	75.89
Loss at access point in %	9.38	9.46	7.59
Consolidated numbers of all runs			
	ITU-T	PM	PM - PTP
Mean Delay in ms	242	237	227
Mean Goodput in kbit/s	70.65	73.36	76.31
Mean Loss at access point in %	9.21	9.25	7.51

Application C:

Simulation run 1	ITU-T	PM	PM - PTP
Delay in ms	521	523	521
Goodput in kbit/s	61.00	71.00	72.00
Loss at access point in %	5.43	7.21	5.21
Simulation run 2	ITU-T	PM	PM - PTP
Delay in ms	528	510	527
Goodput in kbit/s	62.47	72.48	73.22
Loss at access point in %	5.46	7.24	5.69
Simulation run 3	ITU-T	PM	PM - PTP
Delay in ms	536	512	534
Goodput in kbit/s	62.43	71.46	73.06
Loss at access point in %	5.90	7.37	5.28
Simulation run 4	ITU-T	PM	PM - PTP
Delay in ms	521	519	534
Goodput in kbit/s	62.64	72.85	73.70
Loss at access point in %	5.72	7.25	5.49
Simulation run 5	ITU-T	PM	PM - PTP
Delay in ms	517	533	528
Goodput in kbit/s	62.32	72.27	73.44
Loss at access point in %	5.53	7.64	5.65
Simulation run 6	ITU-T	PM	PM - PTP
Delay in ms	534	516	519
Goodput in kbit/s	61.46	72.26	72.61
Loss at access point in %	5.60	7.60	5.60
Simulation run 7	ITU-T	PM	PM - PTP
Delay in ms	511	530	514
Goodput in kbit/s	61.04	72.80	73.08
Loss at access point in %	5.75	7.50	5.21
Simulation run 8	ITU-T	PM	PM - PTP
Delay in ms	524	510	510
Goodput in kbit/s	61.05	71.89	73.37
Loss at access point in %	5.91	7.28	5.45

Simulation run 9	ITU-T	PM	PM - PTP
Delay in ms	533	536	509
Goodput in kbit/s	62.44	72.28	73.80
Loss at access point in %	5.78	7.32	5.34
Simulation run 10	ITU-T	PM	PM - PTP
Delay in ms	533	524	509
Goodput in kbit/s	62.48	72.34	73.10
Loss at access point in %	5.81	7.36	5.57
Consolidated numbers of all runs			
	ITU-T	PM	PM - PTP
Mean Delay in ms	526	521	521
Mean Goodput in kbit/s	61.93	72.16	73.14
Mean Loss at access point in %	5.69	7.38	5.45

C.2 Utilisation levels by Night

Please note only weekdays were profiled. No weekend data was monitored.

Time	University by Night during Holidays	University by Night	AT&T by Night
19:00	4.20	26.11	2.17
19:02	5.40	25.60	2.41
19:04	4.97	28.25	2.11
19:06	5.06	28.42	2.21
19:08	3.43	22.10	2.45
19:10	3.89	23.28	2.97
19:12	4.91	24.98	2.46
19:14	3.07	26.78	2.22
19:16	4.05	24.74	3.95
19:18	4.34	28.44	3.50
19:20	5.12	26.04	3.67
19:22	6.30	25.50	3.51
19:24	5.74	22.50	3.50
19:26	6.14	26.46	3.65
19:28	3.52	22.41	3.82
19:30	5.94	23.44	8.22
19:32	5.65	24.68	8.68
19:34	3.72	26.15	8.92
19:36	2.99	23.34	8.55
19:38	6.38	24.60	10.03
19:40	6.30	26.28	10.70
19:42	5.97	25.48	10.25
19:44	6.08	31.42	8.66
19:46	5.53	35.54	8.90
19:48	4.93	22.41	8.36
19:50	3.54	23.04	8.81
19:52	4.22	24.91	8.90
19:54	3.33	26.60	8.93
19:56	2.69	37.18	8.88
19:58	6.00	26.21	10.06
20:00	6.25	26.23	12.05
20:02	12.40	20.17	6.96
20:04	4.43	26.24	16.91
20:06	6.45	25.77	5.91
20:08	3.58	31.40	14.07
20:10	3.42	28.38	12.15
20:12	3.82	22.84	2.91
20:14	5.05	23.17	12.25
20:16	4.38	24.47	12.47
20:18	2.54	33.20	3.62
20:20	4.60	24.77	4.00
20:22	3.30	28.19	8.34
20:24	4.00	26.09	6.78
20:26	4.21	25.76	4.41

Time	University by Night during Holidays	University by Night	AT&T by Night
20:28	4.15	31.72	3.76
20:30	4.93	26.50	7.19
20:32	3.87	22.27	13.04
20:34	4.86	32.85	10.74
20:36	3.57	25.94	6.64
20:38	4.34	31.22	7.25
20:40	4.03	34.85	6.29
20:42	10.30	22.63	7.56
20:44	6.05	23.19	7.44
20:46	4.57	24.59	8.83
20:48	4.65	26.41	6.67
20:50	5.05	24.73	9.13
20:52	2.97	28.89	6.13
20:54	4.65	26.67	9.95
20:56	2.79	25.93	9.37
20:58	4.44	31.56	6.84
21:00	3.68	26.55	6.72
21:02	5.94	22.97	8.46
21:04	6.33	26.02	9.87
21:06	5.87	25.22	10.88
21:08	4.04	31.71	9.51
21:10	4.73	34.43	6.67
21:12	2.61	22.29	18.12
21:14	3.87	23.43	10.36
21:16	5.31	24.65	3.60
21:18	2.68	26.68	8.41
21:20	4.34	24.79	6.23
21:22	3.61	32.73	6.89
21:24	5.42	26.70	10.63
21:26	6.08	25.43	6.37
21:28	2.79	39.78	10.43
21:30	5.11	26.07	4.49
21:32	6.33	22.16	3.03
21:34	2.73	26.89	4.01
21:36	2.95	34.41	3.36
21:38	3.16	31.33	6.96
21:40	4.49	28.31	9.53
21:42	5.46	22.28	3.23
21:44	5.17	23.96	2.46
21:46	3.35	24.73	8.75
21:48	6.48	26.10	15.79
21:50	2.98	33.20	20.36
21:52	6.29	28.33	22.74
21:54	4.48	26.87	15.09
21:56	3.96	37.05	20.56
21:58	4.27	31.56	15.13
22:00	6.18	26.23	20.68
22:02	3.40	22.28	22.50
22:04	6.11	26.43	5.43

Time	University by Night during Holidays	University by Night	AT&T by Night
22:06	2.86	25.00	2.54
22:08	4.98	31.06	3.65
22:10	3.06	28.60	4.00
22:12	3.57	22.18	1.12
22:14	4.51	23.19	5.12
22:16	4.46	24.01	3.69
22:18	6.46	26.79	2.01
22:20	3.89	24.74	4.30
22:22	3.10	28.12	1.95
22:24	5.70	26.71	3.57
22:26	5.40	25.30	5.55
22:28	4.71	31.26	5.75
22:30	4.63	26.12	2.96
22:32	4.54	22.81	6.04
22:34	3.37	26.71	1.71
22:36	4.46	25.74	2.56
22:38	3.27	31.62	2.08
22:40	4.61	28.91	1.70
22:42	2.63	22.51	16.75
22:44	4.44	23.95	20.87
22:46	5.20	24.79	22.28
22:48	4.49	26.20	22.26
22:50	2.82	24.23	22.10
22:52	4.52	28.72	18.25
22:54	2.93	26.13	12.38
22:56	4.01	25.31	5.03
22:58	5.33	31.21	4.91
23:00	5.81	26.32	2.58
23:02	4.18	22.36	1.84
23:04	2.84	15.51	1.80
23:06	5.89	17.07	1.42
23:08	4.38	21.39	1.54
23:10	5.08	13.44	1.25
23:12	3.97	10.91	1.46
23:14	5.47	3.76	1.72
23:16	1.87	3.98	1.06
23:18	6.10	5.80	1.76
23:20	2.64	5.00	1.45
23:22	1.87	3.65	1.29
23:24	4.37	4.93	1.79
23:26	3.58	4.18	1.08
23:28	5.22	3.69	1.85
23:30	1.54	3.20	1.11
23:32	6.75	5.97	1.38
23:34	6.41	4.44	1.75
23:36	3.70	3.13	1.29
23:38	3.58	4.26	1.66
23:40	4.48	4.03	7.86
23:42	4.59	3.72	5.42

Time	University by Night during Holidays	University by Night	AT&T by Night
23:44	13.13	3.04	3.17
23:46	4.98	11.03	4.18
23:48	5.48	4.14	1.42
23:50	5.39	3.27	2.62
23:52	2.85	4.23	5.08
23:54	5.50	4.42	2.85
23:56	4.43	3.23	2.53
23:58	4.15	3.83	3.27
00:00	6.74	5.08	2.02
00:02	3.79	4.79	2.88
00:04	3.15	3.75	5.69
00:06	2.82	4.35	4.09
00:08	3.17	4.47	5.86
00:10	5.06	3.22	2.25
00:12	2.89	3.06	4.89
00:14	7.38	5.56	4.18
00:16	5.47	4.77	1.55
00:18	4.40	3.54	3.49
00:20	5.39	4.00	3.45
00:22	4.76	4.85	1.78
00:24	4.60	3.15	3.56
00:26	3.00	3.33	5.01
00:28	1.84	3.83	2.24
00:30	3.38	3.34	1.94
00:32	6.62	5.21	5.43
00:34	5.76	4.29	3.10
00:36	3.11	3.65	3.90
00:38	6.20	4.87	3.91
00:40	5.65	4.39	4.13
00:42	3.52	3.86	3.88
00:44	2.11	3.19	3.81
00:46	13.15	11.57	13.10
00:48	3.91	4.06	4.71
00:50	4.95	3.14	2.74
00:52	3.44	4.69	3.24
00:54	3.46	4.72	1.49
00:56	4.75	3.33	6.00
00:58	3.30	3.95	4.51
01:00	2.78	3.92	1.04
01:02	4.92	5.43	2.12
01:04	4.21	4.62	5.36
01:06	4.12	3.87	1.92
01:08	2.69	4.05	4.13
01:10	6.36	4.25	2.21
01:12	3.13	3.11	3.77
01:14	1.79	3.50	3.29
01:16	5.55	5.29	4.17
01:18	4.88	11.96	12.21
01:20	4.13	3.39	3.19

Time	University by Night during Holidays	University by Night	AT&T by Night
01:22	4.12	4.52	3.17
01:24	5.41	4.11	4.21
01:26	3.97	3.47	3.16
01:28	3.54	3.40	2.43
01:30	4.50	3.21	3.67
01:32	4.16	5.76	2.53
01:34	4.68	4.92	3.33
01:36	4.45	3.88	3.36
01:38	4.19	4.39	4.93
01:40	2.61	4.63	3.31
01:42	3.92	3.36	1.21
01:44	3.01	3.59	3.02
01:46	7.33	5.93	3.15
01:48	6.48	4.78	4.31
01:50	2.31	3.77	4.88
01:52	4.19	4.41	2.03
01:54	5.22	4.42	3.67
01:56	4.15	3.80	4.06
01:58	1.83	3.57	5.97
02:00	3.35	3.63	4.80
02:02	14.60	5.82	4.08
02:04	4.97	15.11	15.52
02:06	7.25	5.26	5.58
02:08	5.44	4.97	3.99
02:10	4.28	3.90	2.81
02:12	5.14	4.62	2.70
02:14	5.68	4.93	2.16
02:16	4.33	3.92	4.18
02:18	4.34	3.89	3.47
02:20	4.43	5.14	5.91
02:22	6.35	4.14	3.60
02:24	3.55	3.11	1.73
02:26	4.65	4.97	3.31
02:28	4.93	4.19	1.54
02:30	2.77	3.10	4.64
02:32	3.01	3.46	5.67
02:34	3.72	5.66	2.00
02:36	3.78	4.31	2.82
02:38	4.93	3.25	1.52
02:40	4.04	4.84	5.55
02:42	2.80	4.90	3.25
02:44	1.96	3.94	4.38
02:46	2.24	3.58	2.20
02:48	4.29	5.43	3.72
02:50	8.33	4.92	3.59
02:52	3.17	3.97	5.77
02:54	4.84	9.94	10.80
02:56	3.63	4.52	5.27
02:58	2.34	3.90	4.81

Time	University by Night during Holidays	University by Night	AT&T by Night
03:00	4.03	3.21	2.46
03:02	6.17	5.51	2.33
03:04	4.43	4.24	3.74
03:06	3.52	3.22	3.03
03:08	3.18	4.48	5.71
03:10	6.06	4.67	2.02
03:12	5.28	3.12	5.68
03:14	2.62	3.98	2.62
03:16	6.55	5.99	5.87
03:18	3.94	4.28	2.69
03:20	2.88	3.32	1.74
03:22	5.68	4.17	11.94
03:24	5.90	4.62	12.99
03:26	3.43	3.96	10.31
03:28	1.85	3.95	7.41
03:30	6.73	5.40	5.46
03:32	4.48	13.09	14.51
03:34	3.96	3.27	12.58
03:36	4.40	4.89	11.89
03:38	5.59	4.77	10.21
03:40	2.34	3.52	7.44
03:42	3.84	3.89	7.67
03:44	6.85	5.68	5.67
03:46	5.06	4.72	8.66
03:48	2.84	3.15	6.23
03:50	4.37	4.78	8.10
03:52	6.10	4.67	4.19
03:54	2.02	3.77	2.78
03:56	5.10	3.66	1.69
03:58	6.22	5.81	5.26
04:00	2.67	4.48	4.54
04:02	5.10	3.54	3.45
04:04	2.79	4.79	1.64
04:06	3.87	4.11	2.44
04:08	5.28	3.75	3.27
04:10	4.27	3.97	3.25
04:12	1.84	3.99	3.52
04:14	2.00	4.00	5.47
04:16	4.00	5.69	2.09
04:18	3.20	4.55	4.59
04:20	2.93	3.58	3.78
04:22	3.22	4.09	2.14
04:24	3.76	4.98	4.96
04:26	4.94	3.02	2.30
04:28	2.24	3.28	3.13
04:30	5.62	5.82	3.73
04:32	6.28	4.60	4.43
04:34	4.82	3.75	4.20
04:36	4.39	4.09	1.14

Time	University by Night during Holidays	University by Night	AT&T by Night
04:38	2.71	4.49	4.59
04:40	3.51	3.14	4.69
04:42	1.57	3.30	5.59
04:44	5.07	3.48	3.11
04:46	4.32	5.29	3.19
04:48	3.52	4.86	1.04
04:50	3.66	3.78	3.55
04:52	4.12	4.21	4.71
04:54	6.49	4.03	2.62
04:56	3.10	3.20	2.42
04:58	2.14	3.34	1.72
05:00	5.70	5.03	2.61
05:02	5.33	4.10	2.14
05:04	4.51	3.70	5.35
05:06	5.82	4.00	4.87
05:08	3.42	4.60	5.48
05:10	4.00	3.02	1.08
05:12	3.51	3.40	3.00
05:14	1.59	3.24	5.69
05:16	7.18	5.02	6.75
05:18	4.12	4.56	8.54
05:20	2.49	3.55	6.32
05:22	3.14	4.59	4.00
05:24	3.74	4.10	8.87
05:26	3.83	3.10	8.39
05:28	1.80	3.57	7.14
05:30	6.38	5.61	8.22
05:32	4.77	4.29	6.38
05:34	4.17	3.22	8.96
05:36	5.55	4.91	2.41
05:38	3.17	4.07	10.08
05:40	10.40	3.86	6.83
05:42	4.39	3.99	8.17
05:44	3.72	3.15	8.80
05:46	6.03	5.29	8.25
05:48	5.22	8.91	9.48
05:50	5.58	5.22	8.88
05:52	4.57	4.12	3.37
05:54	3.49	3.90	8.76
05:56	4.46	4.29	10.16
05:58	4.52	4.81	8.04
06:00	2.75	3.81	6.53
06:02	2.74	3.39	8.41
06:04	5.95	5.74	8.69
06:06	5.46	4.91	8.83
06:08	2.92	3.20	6.31
06:10	4.64	4.83	2.19
06:12	5.58	4.38	8.65
06:14	3.96	3.09	8.05

Time	University by Night during Holidays	University by Night	AT&T by Night
06:16	3.09	3.72	8.39
06:18	5.16	5.96	8.19
06:20	2.93	4.02	11.05
06:22	3.23	3.56	7.91
06:24	3.89	4.19	11.57
06:26	5.22	4.38	11.65
06:28	4.78	3.10	12.39
06:30	4.48	3.71	11.90
06:32	4.22	5.43	9.43
06:34	4.41	4.86	8.63
06:36	4.96	3.86	11.97
06:38	5.97	4.46	11.59
06:40	6.26	4.38	9.96
06:42	5.40	3.96	14.65
06:44	3.17	3.97	11.82
06:46	7.47	5.02	11.82
06:48	5.58	4.28	11.70
06:50	2.51	3.17	9.68
06:52	4.96	4.86	14.07
06:54	5.18	4.64	14.95
06:56	2.02	3.68	14.14
06:58	5.33	3.82	10.91
07:00	7.49	5.14	11.89

C.3 Utilisation levels by Day

Time	University by Day during Holidays	University by Day	AT&T by Day
07:00	6.45	6.32	6.00
07:02	4.18	11.36	4.44
07:04	3.83	4.88	4.83
07:06	6.46	9.69	6.68
07:08	5.72	8.73	6.79
07:10	6.42	2.01	6.01
07:12	4.35	10.81	4.68
07:14	3.65	3.28	4.81
07:16	4.06	0.13	4.08
07:18	4.49	6.34	4.92
07:20	5.53	4.94	6.58
07:22	6.41	8.49	6.09
07:24	6.43	5.20	6.29
07:26	5.00	3.34	5.72
07:28	2.95	7.95	3.06
07:30	5.77	8.12	6.57
07:32	3.27	7.84	3.32
07:34	5.35	3.13	5.47
07:36	6.01	4.06	6.33
07:38	6.43	4.72	6.32
07:40	3.60	2.39	4.47
07:42	4.03	10.44	4.18
07:44	4.73	3.72	5.38
07:46	3.59	3.58	4.06
07:48	3.82	8.73	4.41
07:50	3.51	10.09	4.44
07:52	5.14	7.58	5.77
07:54	1.67	5.21	2.60
07:56	5.75	6.50	6.47
07:58	4.79	6.78	5.97
08:00	0.73	10.70	1.40
08:02	2.88	9.23	6.51
08:04	0.70	10.22	8.32
08:06	4.03	6.61	6.94
08:08	1.47	10.15	10.62
08:10	2.76	17.41	10.21
08:12	4.58	15.15	12.37
08:14	0.86	13.86	11.17
08:16	3.61	17.67	14.78
08:18	2.83	19.88	16.87
08:20	1.81	10.23	7.23
08:22	2.92	28.32	11.70
08:24	4.66	26.11	19.83
08:26	1.05	25.27	13.40
08:28	2.06	38.41	13.00

Time	University by Day during Holidays	University by Day	AT&T by Day
08:30	3.10	26.71	27.72
08:32	2.34	22.61	12.55
08:34	2.03	35.54	11.97
08:36	1.38	25.23	20.42
08:38	1.57	31.76	9.79
08:40	1.99	34.40	19.82
08:42	2.03	22.97	18.31
08:44	3.79	23.48	12.57
08:46	2.84	24.33	15.38
08:48	0.98	32.96	14.90
08:50	3.56	24.76	16.45
08:52	3.62	29.00	14.75
08:54	5.31	26.13	15.10
08:56	1.28	25.11	14.19
08:58	4.94	37.21	12.04
09:00	2.77	26.48	20.37
09:02	9.72	22.96	17.59
09:04	10.58	26.54	11.41
09:06	11.07	25.41	11.91
09:08	11.61	31.49	12.33
09:10	14.27	35.26	14.28
09:12	17.36	22.35	17.26
09:14	15.83	23.51	16.53
09:16	11.52	24.72	18.13
09:18	18.22	26.18	18.88
09:20	8.71	24.98	9.68
09:22	22.39	32.98	23.37
09:24	11.26	26.82	11.76
09:26	15.95	25.47	16.89
09:28	12.76	35.80	13.23
09:30	20.53	26.36	21.61
09:32	15.65	22.89	16.80
09:34	15.82	26.79	16.68
09:36	15.44	34.96	15.98
09:38	16.95	31.41	17.95
09:40	13.63	28.44	14.70
09:42	14.83	22.22	15.84
09:44	17.24	23.54	17.72
09:46	10.29	24.44	10.58
09:48	19.26	26.36	19.61
09:50	18.31	33.41	18.95
09:52	22.99	28.62	23.01
09:54	10.60	26.39	11.51
09:56	18.86	34.77	19.81
09:58	16.17	31.92	16.22
10:00	15.52	26.75	16.11
10:02	17.78	22.31	18.84
10:04	18.57	26.42	19.18
10:06	16.87	25.44	17.45

Time	University by Day during Holidays	University by Day	AT&T by Day
10:08	21.20	31.31	21.91
10:10	20.13	28.51	20.61
10:12	11.78	22.14	12.25
10:14	15.08	23.36	15.96
10:16	10.56	24.07	11.94
10:18	18.92	26.67	19.14
10:20	15.59	24.17	16.47
10:22	24.75	28.85	25.85
10:24	26.06	26.64	26.01
10:26	19.52	25.84	19.60
10:28	21.03	31.95	21.92
10:30	18.09	26.93	18.11
10:32	15.88	22.97	16.72
10:34	23.02	26.42	23.05
10:36	12.64	25.44	13.64
10:38	8.74	31.25	9.73
10:40	8.18	28.03	8.75
10:42	8.57	22.38	9.51
10:44	8.68	23.30	8.70
10:46	17.11	24.88	17.65
10:48	15.23	26.61	15.07
10:50	15.59	24.27	16.90
10:52	17.08	28.91	17.33
10:54	15.31	26.60	15.86
10:56	15.25	25.36	15.67
10:58	16.12	34.40	16.17
11:00	9.62	26.28	10.95
11:02	13.66	22.07	14.10
11:04	13.22	28.86	14.36
11:06	9.96	17.45	10.47
11:08	15.04	21.03	15.43
11:10	14.70	40.90	15.33
11:12	10.68	27.61	11.03
11:14	13.55	31.25	14.72
11:16	15.13	31.28	15.61
11:18	11.22	30.97	11.49
11:20	15.14	31.90	15.78
11:22	11.72	24.98	12.29
11:24	14.73	27.30	15.20
11:26	31.81	22.34	18.43
11:28	15.82	24.67	16.94
11:30	13.93	26.90	14.76
11:32	13.97	33.19	14.04
11:34	12.75	23.44	13.47
11:36	15.36	24.61	15.88
11:38	14.07	20.55	14.98
11:40	12.05	28.36	12.48
11:42	18.92	23.13	19.15
11:44	14.07	30.38	14.11

Time	University by Day during Holidays	University by Day	AT&T by Day
11:46	16.45	26.82	16.54
11:48	20.14	24.99	20.82
11:50	13.88	21.37	14.94
11:52	17.87	28.41	18.06
11:54	15.00	29.66	15.05
11:56	13.03	29.16	13.10
11:58	11.76	38.81	12.14
12:00	16.11	34.87	16.25
12:02	18.87	27.85	19.30
12:04	18.15	20.37	18.33
12:06	16.31	31.48	16.97
12:08	13.81	23.04	14.36
12:10	19.85	41.39	20.90
12:12	16.71	31.97	17.60
12:14	12.86	33.79	13.44
12:16	17.05	17.17	17.94
12:18	20.04	31.92	20.44
12:20	18.95	23.20	19.63
12:22	17.89	38.75	18.67
12:24	10.32	26.36	10.01
12:26	15.31	24.18	15.70
12:28	18.75	26.55	19.30
12:30	11.10	28.90	11.28
12:32	13.74	32.06	14.83
12:34	16.58	23.80	17.82
12:36	12.86	33.84	13.02
12:38	10.94	28.96	11.12
12:40	12.32	27.81	12.20
12:42	10.01	36.20	10.21
12:44	13.64	23.54	13.29
12:46	18.04	23.13	18.19
12:48	11.59	29.72	11.76
12:50	17.02	28.32	18.06
12:52	11.62	30.56	12.82
12:54	19.54	37.31	20.42
12:56	14.10	23.31	14.53
12:58	12.00	30.71	12.99
13:00	11.31	24.63	11.83
13:02	9.14	24.00	10.10
13:04	15.63	23.57	16.25
13:06	19.02	29.36	18.85
13:08	10.74	24.22	11.62
13:10	21.22	27.89	21.74
13:12	15.75	38.49	16.06
13:14	19.33	33.61	19.25
13:16	14.28	28.87	14.87
13:18	15.84	25.43	16.67
13:20	10.25	24.05	10.56
13:22	20.57	31.88	21.39

Time	University by Day during Holidays	University by Day	AT&T by Day
13:24	16.10	37.92	16.01
13:26	19.66	24.47	20.58
13:28	14.51	23.89	15.00
13:30	22.15	23.10	22.95
13:32	17.95	27.75	18.05
13:34	16.54	27.71	17.26
13:36	18.63	40.45	19.86
13:38	11.47	33.84	11.98
13:40	10.94	27.67	11.77
13:42	16.66	41.44	17.94
13:44	17.12	33.45	17.49
13:46	29.96	22.09	22.32
13:48	13.43	28.93	13.81
13:50	14.92	35.82	15.24
13:52	13.78	34.57	14.66
13:54	15.11	34.94	15.95
13:56	17.99	21.31	18.18
13:58	14.70	22.91	15.44
14:00	15.36	26.40	15.03
14:02	8.42	25.50	8.89
14:04	20.21	39.61	20.56
14:06	14.28	28.84	14.45
14:08	16.15	30.96	16.07
14:10	21.03	38.22	21.10
14:12	9.78	29.35	10.40
14:14	17.06	26.63	17.77
14:16	10.01	22.16	10.30
14:18	13.16	32.21	13.73
14:20	10.63	34.07	11.06
14:22	14.51	20.03	15.24
14:24	13.87	30.55	14.06
14:26	17.11	24.28	17.74
14:28	14.47	28.49	14.46
14:30	10.94	23.46	11.15
14:32	11.25	35.85	11.75
14:34	15.10	28.60	15.64
14:36	12.32	26.04	12.55
14:38	10.63	23.14	11.96
14:40	17.07	32.29	17.30
14:42	17.70	21.39	18.77
14:44	9.64	20.24	10.35
14:46	11.96	28.34	12.34
14:48	14.79	26.10	15.49
14:50	18.23	24.91	18.42
14:52	19.32	21.83	19.36
14:54	10.88	27.72	11.87
14:56	21.67	23.66	22.38
14:58	9.21	30.28	9.27
15:00	10.91	36.34	11.67

Time	University by Day during Holidays	University by Day	AT&T by Day
15:02	18.39	20.74	18.45
15:04	17.46	20.34	17.12
15:06	11.03	21.11	11.37
15:08	19.54	29.99	20.89
15:10	20.89	36.55	21.57
15:12	21.31	32.13	21.44
15:14	10.18	31.45	10.85
15:16	20.71	25.57	21.83
15:18	13.40	31.82	13.42
15:20	11.44	26.04	11.38
15:22	7.75	28.60	8.13
15:24	10.55	35.16	11.47
15:26	10.18	23.91	10.56
15:28	19.97	24.00	20.61
15:30	16.93	23.25	17.69
15:32	19.82	35.37	20.66
15:34	19.34	26.63	19.17
15:36	13.44	25.82	13.01
15:38	9.77	24.76	10.69
15:40	12.21	32.62	12.36
15:42	10.82	34.45	11.94
15:44	16.21	33.37	16.26
15:46	11.04	28.43	11.58
15:48	16.05	49.45	16.46
15:50	13.44	29.02	13.69
15:52	12.13	28.91	12.66
15:54	14.78	33.80	15.78
15:56	13.23	23.19	13.59
15:58	9.46	25.31	9.53
16:00	14.31	30.52	15.72
16:02	17.99	29.33	18.21
16:04	13.97	30.93	14.18
16:06	10.57	35.84	11.90
16:08	13.16	32.34	13.93
16:10	16.08	28.80	16.10
16:12	16.03	33.10	16.76
16:14	14.11	30.15	14.87
16:16	11.52	28.52	12.60
16:18	16.40	25.92	16.60
16:20	14.71	28.75	15.58
16:22	11.05	40.23	11.27
16:24	16.58	34.28	17.40
16:26	12.37	21.66	12.47
16:28	19.27	24.92	19.33
16:30	15.98	22.23	16.64
16:32	15.17	28.94	15.52
16:34	15.05	26.34	15.76
16:36	13.46	31.17	13.76
16:38	12.51	33.60	13.04

Time	University by Day during Holidays	University by Day	AT&T by Day
16:40	13.73	27.20	14.90
16:42	11.58	34.24	11.42
16:44	17.56	24.51	18.45
16:46	15.93	21.45	16.56
16:48	11.42	33.24	11.15
16:50	8.44	32.15	11.90
16:52	6.15	37.80	15.19
16:54	4.11	31.39	16.98
16:56	4.89	19.03	14.91
16:58	4.64	26.51	14.56
17:00	2.78	29.32	15.55
17:02	6.70	33.46	14.94
17:04	7.06	37.56	9.01
17:06	7.03	34.36	11.86
17:08	5.48	25.32	9.95
17:10	4.67	34.89	6.46
17:12	3.84	32.64	13.38
17:14	2.96	25.36	17.17
17:16	5.82	24.75	9.56
17:18	7.09	29.72	6.06
17:20	5.73	30.95	13.72
17:22	5.02	25.73	8.39
17:24	5.96	22.71	10.72
17:26	4.38	24.15	8.06
17:28	2.59	27.89	9.25
17:30	6.30	30.82	13.25
17:32	3.37	29.39	10.38
17:34	3.32	28.58	11.56
17:36	6.24	23.05	7.25
17:38	5.59	25.87	5.47
17:40	3.10	26.41	8.60
17:42	6.13	24.02	7.03
17:44	3.71	20.15	4.25
17:46	3.90	27.21	0.10
17:48	4.62	21.15	4.42
17:50	3.22	26.65	3.03
17:52	4.23	23.27	7.12
17:54	5.49	30.95	6.13
17:56	6.47	31.09	4.10
17:58	5.36	27.68	6.55
18:00	2.60	30.55	2.06
18:02	5.94	32.61	5.70
18:04	6.58	21.54	0.24
18:06	5.29	26.16	0.80
18:08	4.49	30.77	2.25
18:10	5.46	31.41	7.56
18:12	3.28	34.87	3.54
18:14	3.77	28.93	1.29
18:16	3.51	26.57	5.19

Time	University by Day during Holidays	University by Day	AT&T by Day
18:18	5.04	24.87	10.43
18:20	4.63	25.67	9.48
18:22	4.05	28.21	1.18
18:24	3.63	30.10	1.34
18:26	3.32	27.64	3.97
18:28	3.64	24.63	8.54
18:30	3.51	29.40	7.48
18:32	3.51	28.62	2.08
18:34	6.31	34.16	5.71
18:36	7.47	23.57	0.02
18:38	5.79	26.49	1.33
18:40	4.51	30.59	5.62
18:42	4.50	26.72	5.04
18:44	4.18	36.86	0.16
18:46	7.27	26.10	6.11
18:48	6.63	20.45	1.40
18:50	4.74	35.19	6.40
18:52	3.12	24.22	1.30
18:54	4.39	29.20	7.17
18:56	2.85	42.30	5.12
18:58	6.04	23.01	3.91
19:00	6.78	22.07	7.36

C.4 Application Ratios

C.4.1 Application 1

Date	Time	Packets Sent	Packets Received	Ratio
26.1.1998	09:23	243733	23864	10.21 :1
26.1.1999	09:29	248610	22416	11.09 :1
26.1.1998	09:35	208821	22438	9.31 :1
26.1.1998	09:41	266924	24182	11.04 :1
26.1.1998	09:47	243355	26110	9.32 :1
26.1.1998	10:05	288020	26286	10.96 :1
27.1.1998	09:23	253168	26779	9.45 :1
27.1.1998	09:29	300436	26998	11.13 :1
27.1.1998	09:35	249053	23864	10.44 :1
27.1.1998	09:41	285329	25735	11.09 :1
27.1.1998	09:47	266999	27203	9.82 :1
27.1.1998	10:05	286017	27193	10.52 :1
28.1.1998	09:23	241820	26023	9.29 :1
28.1.1998	09:29	285736	27136	10.53 :1
28.1.1998	09:35	248685	26289	9.46 :1
28.1.1998	09:41	289648	27278	10.62 :1
28.1.1998	09:47	266338	23864	11.16 :1
28.1.1998	10:05	250380	23326	10.73 :1

Date	Time	Packets Sent	Packets Received	Ratio
29.1.1998	09:23	224200	23908	9.38 :1
29.1.1998	09:29	270123	24262	11.13 :1
29.1.1998	09:35	230169	23266	9.89 :1
29.1.1998	09:41	265168	23992	11.05 :1
29.1.1998	09:47	234863	23406	10.03 :1
29.1.1998	10:05	235978	21794	10.83 :1
30.1.1998	09:23	264318	23864	11.08 :1
30.1.1998	09:29	269105	24106	11.16 :1
30.1.1998	09:35	242911	24417	9.95 :1
30.1.1998	09:41	240459	22839	10.53 :1
30.1.1998	09:47	212961	21260	10.02 :1
30.1.1998	10:05	212898	22567	9.43 :1
02.2.1998	09:23	222796	22108	10.08 :1
02.2.1998	09:29	211843	22719	9.32 :1
02.2.1998	09:35	223432	22127	10.10 :1
02.2.1998	09:41	193747	20203	9.59 :1
02.2.1998	09:47	191226	20165	9.48 :1
02.2.1998	10:05	198360	21208	9.35 :1
03.2.1998	09:23	225343	22733	9.91 :1
03.2.1998	09:29	206873	21074	9.82 :1
03.2.1998	09:35	191765	19135	10.02 :1
03.2.1998	09:41	180788	17791	10.16 :1
03.2.1998	09:47	174398	17726	9.84 :1
03.2.1998	10:05	165215	16948	9.75 :1
04.2.1998	09:23	178675	18072	9.89 :1
04.2.1998	09:29	172970	17551	9.86 :1
04.2.1998	09:35	172476	17711	9.74 :1
04.2.1998	09:41	169950	17642	9.63 :1
04.2.1998	09:47	164227	17350	9.47 :1
04.2.1998	10:05	172109	18456	9.33 :1
05.2.1998	09:23	171420	17425	9.84 :1
05.2.1998	09:29	150014	15859	9.46 :1
05.2.1998	09:35	140264	13876	10.11 :1
05.2.1998	09:41	143008	15273	9.36 :1
05.2.1998	09:47	150764	15379	9.80 :1
05.2.1998	10:05	153916	15804	9.74 :1

C.4.2 Application 2

Date	Time	Packets Sent	Packets Received	Ratio
26.1.1998	11:16	111126	8171	13.60 :1
26.1.1999	12:57	114134	8337	13.69 :1
26.1.1998	13:34	112363	8539	13.16 :1
26.1.1998	14:50	122754	9007	13.63 :1
26.1.1998	15:42	113948	8623	13.21 :1
26.1.1998	16:22	127250	8855	14.37 :1
27.1.1998	12:16	118220	8861	13.34 :1
27.1.1998	12:58	120246	8461	14.21 :1

Date	Time	Packets Sent	Packets Received	Ratio
27.1.1998	13:34	119276	8631	13.82 :1
27.1.1998	14:50	119019	8289	14.36 :1
27.1.1998	15:42	107147	8151	13.15 :1
27.1.1998	16:22	117109	8356	14.01 :1
28.1.1998	13:16	114492	8769	13.06 :1
28.1.1998	12.:59	113825	8273	13.76 :1
28.1.1998	13:34	108102	8542	12.66 :1
28.1.1998	14:50	125372	9031	13.88 :1
28.1.1998	15:42	130366	9337	13.96 :1
28.1.1998	16:22	123961	9078	13.66 :1
29.1.1998	14:16	113854	8793	12.95 :1
29.1.1998	12.:60	130240	9042	14.40 :1
29.1.1998	13:34	117988	8886	13.28 :1
29.1.1998	14:50	132642	9354	14.18 :1
29.1.1998	15:42	126130	9672	13.04 :1
29.1.1998	16:22	132520	9629	13.76 :1
30.1.1998	15:16	144668	9933	14.56 :1
30.1.1998	12.:61	140959	9740	14.47 :1
30.1.1998	13:34	127506	9493	13.43 :1
30.1.1998	14:50	128786	9151	14.07 :1
30.1.1998	15:42	120754	8941	13.51 :1
30.1.1998	16:22	122066	9292	13.14 :1
02.2.1998	16:16	123890	9724	12.74 :1
02.2.1998	12.:62	129489	9586	13.51 :1
02.2.1998	13:34	131647	10038	13.11 :1
02.2.1998	14:50	130007	9798	13.27 :1
02.2.1998	15:42	121960	9303	13.11 :1
02.2.1998	16:22	117308	9131	12.85 :1
03.2.1998	17:16	127669	9396	13.59 :1
03.2.1998	12.:63	119675	9042	13.24 :1
03.2.1998	13:34	115994	8650	13.41 :1
03.2.1998	14:50	113392	8883	12.77 :1
03.2.1998	15:42	117538	8811	13.34 :1
03.2.1998	16:22	124249	9147	13.58 :1
04.2.1998	18:16	118906	9304	12.78 :1
04.2.1998	12.:64	125496	9236	13.59 :1
04.2.1998	13:34	121271	9376	12.93 :1
04.2.1998	14:50	129654	9752	13.30 :1
04.2.1998	15:42	135124	10155	13.31 :1
04.2.1998	16:22	130171	9699	13.42 :1
05.2.1998	19:16	131603	9903	13.29 :1
05.2.1998	12.:65	133274	9875	13.50 :1
05.2.1998	13:34	131222	9759	13.45 :1
05.2.1998	14:50	120801	9269	13.03 :1
05.2.1998	15:42	121145	9554	12.68 :1
05.2.1998	16:22	126573	9766	12.96 :1

C.4.3 Application 3

Date	Time	Packets Sent	Packets Received	Ratio
26.1.1998	10:02	854529	43822	19.50 :1
26.1.1999	11:18	855630	41914	20.41 :1
26.1.1998	12:24	781682	40935	19.10 :1
26.1.1998	13:56	787019	39308	20.02 :1
26.1.1998	14:43	753213	39376	19.13 :1
26.1.1998	15:31	802518	39238	20.45 :1
27.1.1998	10:02	750665	39428	19.04 :1
27.1.1998	11:18	786362	39966	19.68 :1
27.1.1998	12:24	819275	40864	20.05 :1
27.1.1998	13:56	808499	40869	19.78 :1
27.1.1998	14:43	742614	38702	19.19 :1
27.1.1998	15:31	749304	38231	19.60 :1
28.1.1998	10:02	769947	40655	18.94 :1
28.1.1998	11:18	860189	41990	20.49 :1
28.1.1998	12:24	779100	40088	19.43 :1
28.1.1998	13:56	756721	38752	19.53 :1
28.1.1998	14:43	837705	41080	20.39 :1
28.1.1998	15:31	861555	42780	20.14 :1
29.1.1998	10:02	874238	45242	19.32 :1
29.1.1998	11:18	926804	46402	19.97 :1
29.1.1998	12:24	941033	48455	19.42 :1
29.1.1998	13:56	1021243	50126	20.37 :1
29.1.1998	14:43	952085	49338	19.30 :1
29.1.1998	15:31	969532	48283	20.08 :1
30.1.1998	10:02	970747	49265	19.70 :1
30.1.1998	11:18	1016983	51351	19.80 :1
30.1.1998	12:24	980787	50594	19.39 :1
30.1.1998	13:56	1010380	51325	19.69 :1
30.1.1998	14:43	990845	51712	19.16 :1
30.1.1998	15:31	956904	51697	18.51 :1
02.2.1998	10:02	936875	50427	18.58 :1
02.2.1998	11:18	964994	50594	19.07 :1
02.2.1998	12:24	948857	50665	18.73 :1
02.2.1998	13:56	988508	52938	18.67 :1
02.2.1998	14:43	994230	52497	18.94 :1
02.2.1998	15:31	960534	50721	18.94 :1
03.2.1998	10:02	1001383	52453	19.09 :1
03.2.1998	11:18	978868	52751	18.56 :1
03.2.1998	12:24	969064	51315	18.88 :1
03.2.1998	13:56	1020440	52698	19.36 :1
03.2.1998	14:43	1022329	54866	18.63 :1
03.2.1998	15:31	1052868	56729	18.56 :1
04.2.1998	10:02	1131499	58850	19.23 :1
04.2.1998	11:18	1119506	58048	19.29 :1
04.2.1998	12:24	1099848	58452	18.82 :1
04.2.1998	13:56	1122515	60547	18.54 :1
04.2.1998	14:43	1100871	58753	18.74 :1

Date	Time	Packets Sent	Packets Received	Ratio
04.2.1998	15:31	1172448	61205	19.16 :1
05.2.1998	10:02	1167909	61025	19.14 :1
05.2.1998	11:18	1151539	60772	18.95 :1
05.2.1998	12:24	1141528	58653	19.46 :1
05.2.1998	13:56	1085551	58609	18.52 :1
05.2.1998	14:43	1170455	60972	19.20 :1
05.2.1998	15:31	1139212	60892	18.71 :1

C.4.4 Application 4

Date	Time	Packets Sent	Packets Received	Ratio
26.1.1998	09:08	172163	55358	3.11 :1
26.1.1999	11:53	170185	52875	3.22 :1
26.1.1998	13:16	117787	52695	2.24 :1
26.1.1998	14:58	168437	50746	3.32 :1
26.1.1998	15:47	162042	52895	3.06 :1
26.1.1998	16:28	172286	54486	3.16 :1
27.1.1998	09:08	116199	54677	2.13 :1
27.1.1998	11:53	210346	55419	3.80 :1
27.1.1998	13:16	171848	55198	3.11 :1
27.1.1998	14:58	229845	56889	4.04 :1
27.1.1998	15:47	118923	54971	2.16 :1
27.1.1998	16:28	168272	53841	3.13 :1
28.1.1998	09:08	122929	54903	2.24 :1
28.1.1998	11:53	225217	55284	4.07 :1
28.1.1998	13:16	143088	55772	2.57 :1
28.1.1998	14:58	201563	53676	3.76 :1
28.1.1998	15:47	202333	53534	3.78 :1
28.1.1998	16:28	171699	52775	3.25 :1
29.1.1998	09:08	157318	51476	3.06 :1
29.1.1998	11:53	163375	50067	3.26 :1
29.1.1998	13:16	143623	48908	2.94 :1
29.1.1998	14:58	202512	49548	4.09 :1
29.1.1998	15:47	113637	47605	2.39 :1
29.1.1998	16:28	153723	47456	3.24 :1
30.1.1998	09:08	158642	45632	3.48 :1
30.1.1998	11:53	146395	43487	3.37 :1
30.1.1998	13:16	129786	43830	2.96 :1
30.1.1998	14:58	176807	45507	3.89 :1
30.1.1998	15:47	114340	46441	2.46 :1
30.1.1998	16:28	134537	47561	2.83 :1
02.2.1998	09:08	113806	45423	2.51 :1
02.2.1998	11:53	116662	44843	2.60 :1
02.2.1998	13:16	116434	45905	2.54 :1
02.2.1998	14:58	105341	45431	2.32 :1
02.2.1998	15:47	110110	44866	2.45 :1
02.2.1998	16:28	103254	42387	2.44 :1
03.2.1998	09:08	106043	42604	2.49 :1

Date	Time	Packets Sent	Packets Received	Ratio
03.2.1998	11:53	113340	42298	2.68 :1
03.2.1998	13:16	131758	42668	3.09 :1
03.2.1998	14:58	131539	44655	2.95 :1
03.2.1998	15:47	105540	46626	2.26 :1
03.2.1998	16:28	122888	47359	2.59 :1
04.2.1998	09:08	100992	46572	2.17 :1
04.2.1998	11:53	147918	47787	3.10 :1
04.2.1998	13:16	128854	47883	2.69 :1
04.2.1998	14:58	137473	45623	3.01 :1
04.2.1998	15:47	120734	43412	2.78 :1
04.2.1998	16:28	136136	44563	3.05 :1
05.2.1998	09:08	133826	44346	3.02 :1
05.2.1998	11:53	118836	46042	2.58 :1
05.2.1998	13:16	113694	47978	2.37 :1
05.2.1998	14:58	121878	48992	2.49 :1
05.2.1998	15:47	134236	50970	2.63 :1
05.2.1998	16:28	132711	52057	2.55 :1

C.4.5 Application 5

Date	Time	Packets Sent	Packets Received	Ratio
26.1.1998	09:58	330205	63379	5.21 :1
26.1.1999	10:17	376363	66203	5.68 :1
26.1.1998	11:45	294572	67006	4.40 :1
26.1.1998	13:26	402394	67109	6.00 :1
26.1.1998	14:58	350792	68987	5.08 :1
26.1.1998	16:47	382038	68884	5.55 :1
27.1.1998	09:58	303792	67339	4.51 :1
27.1.1998	10:17	418409	68112	6.14 :1
27.1.1998	11:45	365254	66932	5.46 :1
27.1.1998	13:26	409079	66028	6.20 :1
27.1.1998	14:58	345175	66673	5.18 :1
27.1.1998	16:47	366031	68291	5.36 :1
28.1.1998	09:58	297788	66384	4.49 :1
28.1.1998	10:17	376715	66206	5.69 :1
28.1.1998	11:45	271836	63347	4.29 :1
28.1.1998	13:26	324549	60963	5.32 :1
28.1.1998	14:58	325861	60940	5.35 :1
28.1.1998	16:47	345385	62438	5.53 :1
29.1.1998	09:58	267343	62966	4.25 :1
29.1.1998	10:17	351970	61485	5.72 :1
29.1.1998	11:45	314841	61034	5.16 :1
29.1.1998	13:26	341548	62449	5.47 :1
29.1.1998	14:58	280084	61909	4.52 :1
29.1.1998	16:47	367275	59795	6.14 :1
30.1.1998	09:58	353595	61985	5.70 :1
30.1.1998	10:17	343477	61587	5.58 :1
30.1.1998	11:45	272809	63315	4.31 :1

Date	Time	Packets Sent	Packets Received	Ratio
30.1.1998	13:26	329904	61418	5.37 :1
30.1.1998	14:58	302512	63380	4.77 :1
30.1.1998	16:47	275999	62962	4.38 :1
02.2.1998	09:58	262290	61391	4.27 :1
02.2.1998	10:17	311403	62714	4.97 :1
02.2.1998	11:45	266481	62429	4.27 :1
02.2.1998	13:26	324237	63733	5.09 :1
02.2.1998	14:58	298708	66306	4.50 :1
02.2.1998	16:47	313107	68003	4.60 :1
03.2.1998	09:58	290822	67394	4.32 :1
03.2.1998	10:17	292156	69293	4.22 :1
03.2.1998	11:45	292655	66563	4.40 :1
03.2.1998	13:26	279414	66242	4.22 :1
03.2.1998	14:58	294970	65136	4.53 :1
03.2.1998	16:47	311890	63729	4.89 :1
04.2.1998	09:58	322681	64478	5.00 :1
04.2.1998	10:17	310613	66386	4.68 :1
04.2.1998	11:45	310612	64433	4.82 :1
04.2.1998	13:26	306032	62314	4.91 :1
04.2.1998	14:58	261118	60463	4.32 :1
04.2.1998	16:47	288712	62143	4.65 :1
05.2.1998	09:58	274783	60110	4.57 :1
05.2.1998	10:17	262716	60662	4.33 :1
05.2.1998	11:45	327095	63038	5.19 :1
05.2.1998	13:26	294440	62221	4.73 :1
05.2.1998	14:58	297788	62472	4.77 :1
05.2.1998	16:47	259538	59515	4.36 :1

C.4.6 Application 6

Date	Time	Packets Sent	Packets Received	Ratio
26.1.1998	11:03	354366	10033	35.32 :1
26.1.1999	12:55	333850	9413	35.47 :1
26.1.1998	13:48	342380	9828	34.84 :1
26.1.1998	14:29	350680	9749	35.97 :1
26.1.1998	15:47	337238	9723	34.68 :1
26.1.1998	16:51	377306	10415	36.23 :1
27.1.1998	11:03	375990	10915	34.45 :1
27.1.1998	12:55	393410	11048	35.61 :1
27.1.1998	13:48	403595	11252	35.87 :1
27.1.1998	14:29	398143	11042	36.06 :1
27.1.1998	15:47	381165	10891	35.00 :1
27.1.1998	16:51	386942	10825	35.75 :1
28.1.1998	11:03	364456	10388	35.08 :1
28.1.1998	12:55	367032	10381	35.36 :1
28.1.1998	13:48	347998	10116	34.40 :1
28.1.1998	14:29	385549	10691	36.06 :1
28.1.1998	15:47	403493	11294	35.73 :1

Date	Time	Packets Sent	Packets Received	Ratio
28.1.1998	16:51	430099	11883	36.19 :1
29.1.1998	11:03	410370	11803	34.77 :1
29.1.1998	12:55	402806	11356	35.47 :1
29.1.1998	13:48	382006	11123	34.34 :1
29.1.1998	14:29	377477	10444	36.14 :1
29.1.1998	15:47	388541	11010	35.29 :1
29.1.1998	16:51	400207	11068	36.16 :1
30.1.1998	11:03	399871	11035	36.24 :1
30.1.1998	12:55	379961	10644	35.70 :1
30.1.1998	13:48	355354	10330	34.40 :1
30.1.1998	14:29	364344	10302	35.37 :1
30.1.1998	15:47	372944	10803	34.52 :1
30.1.1998	16:51	355531	10227	34.76 :1
02.2.1998	11:03	380656	11009	34.58 :1
02.2.1998	12:55	358435	10281	34.86 :1
02.2.1998	13:48	374112	10596	35.31 :1
02.2.1998	14:29	355032	10151	34.98 :1
02.2.1998	15:47	354226	10233	34.62 :1
02.2.1998	16:51	342704	9958	34.41 :1
03.2.1998	11:03	346341	10088	34.33 :1
03.2.1998	12:55	372614	10566	35.27 :1
03.2.1998	13:48	351758	10048	35.01 :1
03.2.1998	14:29	372588	10786	34.54 :1
03.2.1998	15:47	391700	11092	35.31 :1
03.2.1998	16:51	389906	11207	34.79 :1
04.2.1998	11:03	364065	10456	34.82 :1
04.2.1998	12:55	334115	9705	34.43 :1
04.2.1998	13:48	338076	9769	34.61 :1
04.2.1998	14:29	324769	9317	34.86 :1
04.2.1998	15:47	330012	9367	35.23 :1
04.2.1998	16:51	329438	9351	35.23 :1
05.2.1998	11:03	315039	9017	34.94 :1
05.2.1998	12:55	326392	9279	35.18 :1
05.2.1998	13:48	300153	8633	34.77 :1
05.2.1998	14:29	290135	8405	34.52 :1
05.2.1998	15:47	315043	8935	35.26 :1
05.2.1998	16:51	287422	8299	34.63 :1

C.4.7 Application 7

Date	Time	Packets Sent	Packets Received	Ratio
26.1.1998	10:17	719570	102357	7.03 :1
26.1.1999	12:33	763756	98361	7.76 :1
26.1.1998	13:41	623192	94832	6.57 :1
26.1.1998	15:16	727534	92163	7.89 :1
26.1.1998	15:58	604304	94140	6.42 :1
26.1.1998	16:39	728338	97459	7.47 :1
27.1.1998	10:17	614109	97004	6.33 :1

Date	Time	Packets Sent	Packets Received	Ratio
27.1.1998	12:33	759930	99643	7.63 :1
27.1.1998	13:41	768349	100140	7.67 :1
27.1.1998	15:16	758530	102058	7.43 :1
27.1.1998	15:58	668596	104713	6.39 :1
27.1.1998	16:39	780997	102780	7.60 :1
28.1.1998	10:17	681992	100396	6.79 :1
28.1.1998	12:33	759842	98079	7.75 :1
28.1.1998	13:41	641788	100430	6.39 :1
28.1.1998	15:16	751411	97756	7.69 :1
28.1.1998	15:58	750841	97158	7.73 :1
28.1.1998	16:39	750669	99912	7.51 :1
29.1.1998	10:17	685841	101925	6.73 :1
29.1.1998	12:33	722731	98661	7.33 :1
29.1.1998	13:41	641757	98865	6.49 :1
29.1.1998	15:16	769216	100293	7.67 :1
29.1.1998	15:58	592159	97477	6.07 :1
29.1.1998	16:39	685894	94101	7.29 :1
30.1.1998	10:17	681043	92317	7.38 :1
30.1.1998	12:33	656829	89047	7.38 :1
30.1.1998	13:41	546144	89467	6.10 :1
30.1.1998	15:16	661234	91856	7.20 :1
30.1.1998	15:58	636657	93316	6.82 :1
30.1.1998	16:39	604081	91550	6.60 :1
02.2.1998	10:17	646682	92960	6.96 :1
02.2.1998	12:33	591378	90951	6.50 :1
02.2.1998	13:41	569339	87495	6.51 :1
02.2.1998	15:16	517919	84000	6.17 :1
02.2.1998	15:58	524376	84252	6.22 :1
02.2.1998	16:39	558793	86376	6.47 :1
03.2.1998	10:17	565233	88522	6.39 :1
03.2.1998	12:33	580602	87145	6.66 :1
03.2.1998	13:41	554185	85318	6.50 :1
03.2.1998	15:16	536411	87133	6.16 :1
03.2.1998	15:58	613352	87539	7.01 :1
03.2.1998	16:39	605827	90811	6.67 :1
04.2.1998	10:17	535207	87362	6.13 :1
04.2.1998	12:33	606833	89756	6.76 :1
04.2.1998	13:41	621109	89509	6.94 :1
04.2.1998	15:16	614669	92121	6.67 :1
04.2.1998	15:58	581299	95843	6.07 :1
04.2.1998	16:39	633312	94296	6.72 :1
05.2.1998	10:17	593901	92482	6.42 :1
05.2.1998	12:33	581537	93932	6.19 :1
05.2.1998	13:41	629597	90713	6.94 :1
05.2.1998	15:16	626286	90781	6.90 :1
05.2.1998	15:58	545204	87797	6.21 :1
05.2.1998	16:39	566540	88490	6.40 :1

C.5 Session Duration for Different Departments

Examples of sessions during one week have been profiled. The number of session duration have been counted and displayed. This is one a sample of the overall data which has been collected. It can be seen that the profiles for some departments are very similar on different days. However, it can also be noted that some departments do not seem to have a specific profile. This could especially be seen in the case of the multimedia lab.

C.5.1 Session Duration - Secretarial

Number of Sessions during the Day lasting x seconds					
Seconds	Day 1	Day 2	Day 3	Day 4	Day 5
1	100	306	73	89	98
2	500	149	830	477	360
3	181	236	280	222	212
4	435	272	487	278	370
5	407	150	530	341	334
6	576	600	409	296	418
7	446	444	486	340	446
8	580	713	549	549	809
9	786	696	704	707	666
10	742	836	1040	840	1165
11	826	812	992	814	842
12	918	916	789	1096	1000
13	710	954	674	905	952
14	973	971	1008	1027	1006
15	920	820	1042	1033	900
16	985	993	804	760	1607
17	976	766	983	868	868
18	673	1157	975	674	857
19	826	849	831	970	691
20	856	616	884	764	466
21	710	709	683	695	670
22	737	400	488	750	240
23	554	663	691	373	484
24	484	327	573	429	406
25	657	357	356	311	366
26	252	160	423	243	241
27	200	186	354	329	354
28	159	230	327	148	354
29	87	117	76	110	233
30	66	200	61	290	63
31	52	66	480	40	46
32	83	360	86	389	43
33	50	46	55	29	48
34	41	56	51	44	640

C.5.2 Session Duration – Maintenance Department

Number of Sessions during the Day lasting x seconds					
Seconds	Day 1	Day 2	Day 3	Day 4	Day 5
1	100	79	257	78	113
2	200	462	655	103	339
3	506	257	527	70	760
4	450	777	514	158	329
5	724	591	624	205	610
6	769	809	848	147	724
7	822	947	935	324	918
8	1280	1048	785	293	1605
9	910	938	920	349	1040
10	1069	1060	1341	313	965
11	1369	970	930	329	984
12	750	789	777	257	869
13	711	562	591	213	596
14	540	687	530	183	543
15	365	533	600	255	288
16	716	726	562	185	511
17	612	1060	629	265	631
18	920	1028	1015	304	925
19	1080	1240	1000	353	844
20	1065	1130	1432	308	658
21	957	1252	1192	382	760
22	1105	1120	1149	376	1029
23	1413	880	867	291	1208
24	647	880	925	313	812
25	750	800	796	242	806
26	600	619	561	246	637
27	340	617	704	181	594
28	707	492	320	177	409
29	520	527	520	171	607
30	45	706	652	28	516

C.5.3 Session Duration – Researchers in Multimedia Lab

Number of Sessions during the Day lasting x seconds					
Seconds	Day 1	Day 2	Day 3	Day 4	Day 5
1	160	92	121	33	80
2	137	224	201	179	19
3	126	356	332	372	51
4	87	464	426	385	70
5	68	610	614	516	95
6	84	0	713	631	64
7	96	729	747	704	95
8	87	490	607	771	127
9	79	300	550	1098	95
10	79	330	556	703	45
11	49	765	345	1135	986
12	30	2000	211	574	64
13	49	4830	345	95	64
14	57	1760	400	691	140
15	171	727	1200	783	89
16	41	400	288	720	127
17	47	110	326	704	32
18	37	350	262	480	19
19	194	220	1355	748	32
20	29	300	200	797	32
21	36	2820	249	340	25
22	152	430	1061	791	32
23	25	350	173	816	45
24	17	350	120	905	51
25	43	220	300	717	51
26	163	430	1144	1053	5
27	109	200	762	771	1
28	68	780	473	736	7
29	21	50	147	874	2
30	30	570	294	173	8
31	5	200	256	801	8
32	1	889	153	900	9
33	7	160	262	921	9
34	2	490	64	874	6
35	8	870	3	501	3
36	8	2400	2	271	2
37	9	846	1	295	1
38	9	821	8	455	10
39	6	380	8	560	3
40	3	240	9	672	0
41	2	430	9	734	1
42	1	200	6	796	2
43	10	300	3	514	1
44	3	1440	2	877	5
45	0	140	1	706	4
46	1	160	10	510	9

Number of Sessions during the Day lasting x seconds					
Seconds	Day 1	Day 2	Day 3	Day 4	Day 5
48	1	140	5	804	5
49	5	490	3	742	6
50	4	110	6	668	3
51	9	3770	7	818	1
52	1	846	8	206	8
53	5	350	3	788	5
54	6	80	1	173	3
55	3	240	4	867	6
56	1	430	8	920	7
57	8	200	9	818	8
58	5	80	0	709	3
59	3	110	2	649	1
60	6	140	8	713	4
61	7	160	9	886	8
62	8	160	2	495	9
63	3	110	1	553	0
64	1	980	6	737	2
65	4	5	8	736	8
66	8	3	0	712	9
67	9	6	4	1051	2
68	0	7	1	289	1
69	2	8	1	1112	6
70	8	3	9	568	8
71	9	1	1	777	0
72	2	4	9	694	4
73	1	8	1	694	1
74	6	9	2	657	1
75	8	0	10	109	9
76	0	2	3	749	1
77	4	8	5	469	9
78	1	9	5	817	1
79	1	2	1	250	2
80	9	1	2	793	10
81	1	6	8	777	3
82	9	8	3	803	5
83	1	0	6	100	5
84	2	4	10	123	1
85	10	1	2	4	2
86	3	1	5	24	8
87	5	9	1	4	3
88	5	1	9	1	6
89	1	9	2	1	10
90	2	1	3	9	2
91	8	2	5	1	5
92	3	10	3	9	1
93	6	3	1	1	9
94	10	5	5	2	2
95	2	5	3	10	3
96	5	1	1	3	5
97	1	2	6	5	3

Number of Sessions during the Day lasting x seconds					
Seconds	Day 1	Day 2	Day 3	Day 4	Day 5
99	2	3	9	1	5
100	3	6	2	10	3
101	5	10	2	2	1
102	3	2	6	5	6
103	1	5	7	1	1
104	5	1	2	9	9
105	3	9	7	2	2
106	1	2	7	3	2
107	6	3	1	5	6
108	1	5	5	3	7
109	9	3	10	1	2
110	2	1	2	5	7
111	2	5	5	3	7
112	6	3	1	1	1
113	7	1	9	6	5
114	2	6	2	1	5
115	7	1	3	9	3
116	7	9	5	2	1
117	1	2	3	2	6
118	5	2	1	6	1
119	6	6	5	10	9
120	10	7	3	2	2
121	2	2	1	5	2
122	5	7	6	1	6

C.6 No Overload on Transmit Port Queue

C.6.1 No Overload PM

Time	PM	PM	PM	PM	PM	PM
0.1	0.72	0.72	0.72	0.72	0.72	0.72
0.2	1.19	1.19	1.19	1.19	1.19	1.19
0.3	3.33	2.00	2.00	2.00	2.00	2.00
0.4	6.45	7.45	12.45	7.45	3.45	8.45
0.5	8.44	7.44	6.44	8.44	13.44	8.44
0.6	10.52	14.52	16.52	12.52	14.52	14.52
0.7	12.72	9.72	14.72	9.72	8.72	13.72
0.8	15.95	16.95	20.95	22.95	17.95	13.95
0.9	16.61	13.61	12.61	9.61	4.61	4.61
1	18.25	20.25	21.25	18.25	18.25	16.25
1.1	20.47	16.47	15.47	14.47	11.47	9.47
1.2	23.29	28.29	25.29	28.29	29.29	24.29
1.3	25.43	22.43	20.43	25.43	26.43	25.43
1.4	29.00	29.00	32.00	34.00	32.00	37.00
1.5	32.66	36.66	32.66	36.66	36.66	32.66
1.6	37.20	34.20	37.20	39.20	38.20	43.20
1.7	38.88	37.88	36.88	37.88	41.88	40.88

Time	PM	PM	PM	PM	PM	PM
1.8	42.52	46.52	51.52	49.52	44.52	48.52
1.9	45.89	48.89	45.89	44.89	47.89	42.89
2	48.93	52.93	53.93	52.93	47.93	45.93
2.1	46.42	51.42	46.42	42.42	46.42	48.42
2.2	48.10	45.10	44.10	46.10	51.10	47.10
2.3	45.98	42.98	42.98	43.98	45.98	47.98
2.4	44.35	42.35	43.35	47.35	43.35	38.35
2.5	47.90	46.90	48.90	50.90	51.90	47.90
2.6	49.27	53.27	48.27	46.27	48.27	45.27
2.7	48.78	48.78	45.78	40.78	36.78	35.78
2.8	43.75	38.75	43.75	43.75	43.75	45.75
2.9	46.99	50.99	49.99	54.99	55.99	54.99
3	48.64	53.64	55.64	57.64	58.64	53.64
3.1	46.85	46.85	41.85	39.85	37.85	41.85
3.2	50.20	47.20	47.20	43.20	40.20	42.20
3.3	51.36	47.36	47.36	50.36	55.36	56.36
3.4	49.81	45.81	41.81	42.81	42.81	45.81
3.5	48.92	51.92	51.92	52.92	50.92	49.92
3.6	45.36	49.36	50.36	46.36	49.36	46.36
3.7	49.54	54.54	51.54	55.54	53.54	55.54
3.8	49.90	49.90	53.90	58.90	57.90	57.90
3.9	40.10	40.10	40.10	35.10	37.10	37.10
4	35.00	39.00	39.00	38.00	35.00	39.00
4.1	28.30	27.30	24.30	21.30	19.30	23.30
4.2	35.20	34.20	31.20	33.20	28.20	23.20
4.3	39.30	38.30	36.30	31.30	36.30	32.30
4.4	44.00	48.00	47.00	46.00	44.00	42.00
4.5	45.30	47.30	50.30	47.30	48.30	48.30
4.6	35.20	34.20	30.20	30.20	25.20	27.20
4.7	23.00	21.00	25.00	29.00	30.00	33.00
4.8	21.30	25.30	30.30	28.30	25.30	30.30
4.9	22.10	24.10	25.10	27.10	28.10	26.10
5	20.90	16.90	17.90	21.90	25.90	21.90
5.1	23.60	24.60	20.60	18.60	14.60	9.60
5.2	27.10	22.10	22.10	24.10	25.10	20.10
5.3	32.70	32.70	27.70	29.70	24.70	26.70
5.4	30.80	32.80	32.80	32.80	35.80	38.80
5.5	27.30	31.30	36.30	33.30	30.30	26.30
5.6	23.80	27.80	23.80	19.80	22.80	18.80
5.7	21.30	21.30	23.30	25.30	23.30	18.30
5.8	18.80	19.80	22.80	27.80	24.80	20.80
5.9	16.30	17.30	17.30	18.30	16.30	16.30
6	13.70	18.70	19.70	19.70	23.70	27.70
6.1	2.70	5.70	5.70	6.70	6.70	10.70
6.2	2.00	2.00	4.00	8.00	12.00	13.00
6.3	1.70	4.70	6.70	5.70	8.70	12.70
6.4	1.90	5.90	6.90	8.90	9.90	14.90
6.5	2.50	4.50	9.50	12.50	14.50	17.50
6.6	1.90	0.90	5.90	8.90	13.90	14.90
6.7	2.70	3.70	2.70	5.70	8.70	10.70
6.8	2.00	4.00	6.00	5.00	10.00	15.00

Time	PM	PM	PM	PM	PM	PM
6.9	2.00	5.00	8.00	13.00	12.00	11.00
7	2.30	7.30	6.30	6.30	8.30	8.30
7.1	2.50	6.50	10.50	14.50	15.50	15.50
7.2	2.50	5.50	8.50	11.50	14.50	15.50
7.3	2.30	1.30	2.30	5.30	10.30	14.30
7.4	2.50	3.50	6.50	6.50	6.50	5.50
7.5	2.90	4.90	8.90	7.90	7.90	7.90
7.6	2.70	4.70	8.70	13.70	14.70	16.70
7.7	2.70	1.70	1.70	6.70	5.70	9.70
7.8	3.50	4.50	6.50	8.50	11.50	15.50
7.9	2.90	6.90	7.90	9.90	8.90	13.90
8	3.10	3.10	6.10	9.10	10.10	12.10
8.1	3.10	6.10	10.10	9.10	10.10	14.10
8.2	3.70	6.70	6.70	10.70	14.70	17.70
8.3	6.20	10.20	11.20	16.20	19.20	19.20
8.4	7.70	10.70	12.70	13.70	13.70	12.70
8.5	11.00	15.00	16.00	17.00	19.00	20.00
8.6	7.40	7.40	10.40	10.40	9.40	12.40
8.7	5.80	8.80	9.80	14.80	16.80	20.80
8.8	3.70	7.70	6.70	9.70	10.70	11.70
8.9	4.10	4.10	3.10	5.10	10.10	10.10
9	4.10	3.10	7.10	12.10	13.10	12.10
9.1	5.20	4.20	6.20	7.20	11.20	12.20
9.2	6.40	7.40	9.40	8.40	10.40	10.40
9.3	7.90	6.90	9.90	10.90	9.90	11.90
9.4	9.10	8.10	7.10	4.10	4.10	1.10
9.5	10.30	11.30	9.30	14.30	11.30	15.30
9.6	12.80	14.80	12.80	17.80	14.80	18.80
9.7	13.90	18.90	21.90	18.90	16.90	12.90
9.8	14.70	13.70	8.70	3.70	1.00	3.70
9.9	15.70	16.70	11.70	13.70	11.70	9.70
10	16.60	15.60	19.60	19.60	16.60	20.60

C.6.1 No Overload PTP

Time	PTP	PTP	PTP	PTP	PTP	PTP
0.1	1.72	1.72	1.72	1.72	0.28	0.28
0.2	0.19	0.19	1.19	1.19	0.19	2.19
0.3	2.33	3.00	3.00	1.00	2.00	2.00
0.4	6.45	8.45	11.45	8.45	3.45	9.45
0.5	8.44	7.44	6.44	9.44	13.44	7.44
0.6	15.52	14.52	20.52	9.52	11.52	16.52
0.7	12.72	5.72	9.72	11.72	5.72	10.72
0.8	16.95	15.95	21.95	22.95	18.95	14.95
0.9	14.61	9.61	16.61	6.61	6.61	1.61
1	20.25	23.25	21.25	16.25	21.25	20.25
1.1	23.47	11.47	17.47	19.47	12.47	6.47
1.2	23.29	32.29	24.29	24.29	29.29	29.29
1.3	30.43	17.43	24.43	28.43	26.43	28.43
1.4	29.00	28.00	27.00	35.00	36.00	41.00

Time	PTP	PTP	PTP	PTP	PTP	PTP
1.5	29.66	39.66	30.66	40.66	38.66	33.66
1.6	41.20	38.20	37.20	37.20	43.20	45.20
1.7	41.88	36.88	39.88	36.88	37.88	41.88
1.8	41.52	48.52	54.52	51.52	39.52	52.52
1.9	47.89	48.89	50.89	44.89	47.89	41.89
2	47.93	52.93	55.93	53.93	43.93	46.93
2.1	44.42	46.42	48.42	44.42	44.42	52.42
2.2	43.10	43.10	47.10	48.10	46.10	43.10
2.3	45.98	40.98	45.98	41.98	48.98	52.98
2.4	47.35	44.35	46.35	44.35	41.35	40.35
2.5	45.90	50.90	50.90	46.90	50.90	46.90
2.6	52.27	55.27	46.27	51.27	53.27	44.27
2.7	52.78	45.78	40.78	39.78	41.78	39.78
2.8	46.75	40.75	46.75	45.75	41.75	42.75
2.9	41.99	49.99	54.99	59.99	57.99	55.99
3	52.64	51.64	60.64	55.64	55.64	54.64
3.1	45.85	48.85	38.85	43.85	36.85	43.85
3.2	55.20	52.20	49.20	39.20	39.20	37.20
3.3	49.36	48.36	45.36	51.36	53.36	52.36
3.4	46.81	45.81	42.81	40.81	39.81	48.81
3.5	43.92	51.92	55.92	57.92	54.92	49.92
3.6	43.36	47.36	52.36	51.36	52.36	49.36
3.7	49.54	56.54	53.54	50.54	52.54	50.54
3.8	46.90	50.90	55.90	57.90	53.90	58.90
3.9	42.10	36.10	35.10	32.10	33.10	41.10
4	35.00	38.00	37.00	33.00	31.00	40.00
4.1	23.30	28.30	21.30	19.30	24.30	22.30
4.2	32.20	35.20	30.20	38.20	23.20	20.20
4.3	43.30	34.30	37.30	29.30	41.30	29.30
4.4	49.00	49.00	49.00	43.00	43.00	39.00
4.5	40.30	46.30	50.30	42.30	50.30	53.30
4.6	31.20	32.20	27.20	35.20	23.20	27.20
4.7	22.00	17.00	28.00	31.00	31.00	38.00
4.8	22.30	21.30	30.30	25.30	20.30	28.30
4.9	25.10	25.10	28.10	29.10	27.10	21.10
5	23.90	15.90	19.90	26.90	23.90	26.90
5.1	26.60	27.60	20.60	18.60	14.60	4.60
5.2	26.10	18.10	25.10	28.10	27.10	18.10
5.3	27.70	29.70	32.70	29.70	25.70	23.70
5.4	25.80	35.80	34.80	30.80	38.80	38.80
5.5	31.30	33.30	32.30	37.30	30.30	22.30
5.6	23.80	24.80	19.80	16.80	21.80	17.80
5.7	23.30	25.30	21.30	24.30	26.30	19.30
5.8	15.80	18.80	19.80	29.80	24.80	23.80
5.9	15.30	16.30	20.30	23.30	19.30	18.30
6	12.70	23.70	21.70	20.70	23.70	26.70
6.1	3.70	4.70	10.70	6.70	9.70	15.70
6.2	3.00	1.00	3.00	7.00	13.00	15.00
6.3	6.70	7.70	11.70	7.70	10.70	17.70
6.4	0.90	6.90	11.90	11.90	13.90	15.90
6.5	7.50	8.50	13.50	15.50	17.50	19.50

Time	PTP	PTP	PTP	PTP	PTP	PTP
6.6	5.90	4.90	10.90	11.90	12.90	16.90
6.7	7.70	2.70	6.70	10.70	7.70	11.70
6.8	2.00	5.00	10.00	7.00	11.00	19.00
6.9	5.00	7.00	12.00	16.00	15.00	13.00
7	5.30	11.30	9.30	10.30	12.30	7.30
7.1	6.50	8.50	15.50	13.50	20.50	20.50
7.2	2.50	9.50	11.50	15.50	19.50	20.50
7.3	4.30	1.30	3.30	4.30	15.30	18.30
7.4	7.50	4.50	10.50	6.50	9.50	7.50
7.5	1.90	8.90	8.90	8.90	12.90	10.90
7.6	2.70	3.70	8.70	13.70	13.70	20.70
7.7	3.70	5.70	1.70	6.70	10.70	12.70
7.8	4.50	7.50	11.50	12.50	11.50	15.50
7.9	3.90	11.90	12.90	11.90	13.90	16.90
8	6.10	2.10	5.10	9.10	9.10	11.10
8.1	8.10	7.10	15.10	12.10	14.10	18.10
8.2	3.70	11.70	9.70	14.70	18.70	20.70
8.3	8.20	11.20	14.20	20.20	22.20	21.20
8.4	12.70	10.70	12.70	16.70	13.70	13.70
8.5	13.00	16.00	20.00	22.00	21.00	21.00
8.6	8.40	10.40	10.40	10.40	10.40	12.40
8.7	5.80	12.80	9.80	13.80	15.80	22.80
8.8	2.70	9.70	8.70	14.70	14.70	12.70
8.9	3.10	4.10	7.10	5.10	15.10	14.10
9	5.10	6.10	12.10	17.10	15.10	13.10
9.1	8.20	7.20	7.20	9.20	14.20	11.20
9.2	10.40	3.40	12.40	6.40	12.40	6.40
9.3	3.90	5.90	10.90	7.90	13.90	6.90
9.4	9.10	8.10	6.10	3.10	5.10	6.10
9.5	13.30	10.30	7.30	15.30	11.30	15.30
9.6	16.80	12.80	11.80	16.80	19.80	13.80
9.7	12.90	22.90	25.90	22.90	21.90	10.90
9.8	15.70	13.70	9.70	4.70	6.00	5.70
9.9	19.70	17.70	15.70	18.70	12.70	6.70
10	16.60	10.60	23.60	15.60	21.60	22.60

C.7 Overload on Transmit Port Queue

C.7.1 Overload PM

Time	PM	PM	PM	PM	PM	PM
0.1	0.72	5.72	5.72	5.72	2.72	1.72
0.2	1.19	6.19	6.19	5.19	5.19	1.19
0.3	3.33	3.33	7.33	7.33	3.33	5.33
0.4	6.45	9.45	7.45	7.45	7.45	9.45
0.5	8.44	8.44	13.44	13.44	11.44	8.44
0.6	10.52	12.52	11.52	12.52	14.52	12.52
0.7	12.72	17.72	15.72	12.72	17.72	14.72

Time	PM	PM	PM	PM	PM	PM
0.8	15.95	15.95	16.95	19.95	20.95	19.95
0.9	16.61	17.61	16.61	17.61	21.61	17.61
1	18.25	20.25	18.25	21.25	19.25	21.25
1.1	20.47	21.47	25.47	21.47	22.47	20.47
1.2	23.29	23.29	28.29	26.29	24.29	26.29
1.3	25.43	25.43	25.43	28.43	28.43	30.43
1.4	29.00	30.00	30.00	29.00	30.00	32.00
1.5	32.66	35.66	34.66	34.66	34.66	34.66
1.6	37.20	40.20	38.20	42.20	39.20	38.20
1.7	38.88	38.88	39.88	41.88	39.88	41.88
1.8	42.52	47.52	42.52	47.52	46.52	47.52
1.9	45.89	45.89	48.89	48.89	48.89	46.89
2	48.93	51.93	48.93	52.93	51.93	50.93
2.1	46.42	47.42	47.42	51.42	48.42	47.42
2.2	48.10	49.10	50.10	50.10	51.10	52.10
2.3	45.98	46.98	46.98	46.98	50.98	47.98
2.4	44.35	44.35	47.35	48.35	47.35	44.35
2.5	47.90	47.90	49.90	52.90	50.90	51.90
2.6	49.27	51.27	50.27	52.27	54.27	53.27
2.7	48.78	50.78	49.78	51.78	48.78	53.78
2.8	43.75	43.75	47.75	46.75	48.75	48.75
2.9	46.99	46.99	51.99	48.99	50.99	49.99
3	48.64	53.64	49.64	49.64	48.64	53.64
3.1	46.85	49.85	49.85	47.85	47.85	47.85
3.2	50.20	51.20	52.20	51.20	51.20	50.20
3.3	51.36	52.36	56.36	54.36	53.36	55.36
3.4	49.81	52.81	50.81	49.81	50.81	54.81
3.5	48.92	48.92	50.92	51.92	53.92	49.92
3.6	45.36	45.36	49.36	47.36	45.36	47.36
3.7	49.54	53.54	52.54	51.54	54.54	53.54
3.8	49.90	50.90	52.90	54.90	50.90	53.90
3.9	48.10	52.10	49.10	49.10	53.10	51.10
4	44.93	48.93	49.93	45.93	48.93	46.93
4.1	42.87	46.87	47.87	43.87	43.87	45.87
4.2	46.37	48.37	46.37	46.37	48.37	50.37
4.3	48.71	51.71	50.71	51.71	49.71	50.71
4.4	46.89	51.89	50.89	51.89	48.89	48.89
4.5	51.50	54.50	55.50	52.50	53.50	56.50
4.6	49.16	53.16	49.16	51.16	53.16	51.16
4.7	48.04	48.04	53.04	48.04	48.04	48.04
4.8	46.18	48.18	46.18	46.18	50.18	48.18
4.9	47.57	49.57	52.57	51.57	47.57	50.57
5	50.63	54.63	51.63	50.63	50.63	53.63
5.1	51.40	56.40	55.40	52.40	56.40	54.40
5.2	53.61	56.61	57.61	58.61	53.61	58.61
5.3	49.73	52.73	50.73	52.73	53.73	52.73
5.4	47.44	50.44	49.44	47.44	48.44	52.44
5.5	51.80	54.80	56.80	54.80	56.80	53.80
5.6	52.39	53.39	53.39	54.39	52.39	56.39
5.7	46.83	51.83	48.83	49.83	50.83	51.83
5.8	51.32	54.32	56.32	51.32	53.32	55.32

Time	PM	PM	PM	PM	PM	PM
5.9	52.91	57.91	56.91	55.91	54.91	53.91
6	53.56	56.56	58.56	55.56	57.56	53.56
6.1	55.17	58.17	60.17	59.17	60.17	58.17
6.2	52.76	57.76	52.76	52.76	56.76	54.76
6.3	54.75	56.75	59.75	55.75	57.75	58.75
6.4	52.34	56.34	56.34	53.34	54.34	56.34
6.5	49.19	52.19	50.19	50.19	54.19	51.19
6.6	45.21	45.21	45.21	49.21	50.21	50.21
6.7	42.33	44.33	43.33	42.33	46.33	43.33
6.8	39.45	40.45	41.45	39.45	39.45	40.45
6.9	37.35	40.35	37.35	37.35	40.35	40.35
7	33.83	38.83	34.83	38.83	35.83	35.83
7.1	30.64	32.64	35.64	32.64	35.64	31.64
7.2	27.72	27.72	29.72	30.72	29.72	27.72
7.3	26.21	31.21	29.21	28.21	27.21	31.21
7.4	24.58	27.58	28.58	28.58	24.58	25.58
7.5	22.34	24.34	27.34	26.34	24.34	22.34
7.6	19.81	21.81	21.81	19.81	21.81	24.81
7.7	19.03	19.03	24.03	20.03	24.03	21.03
7.8	18.61	19.61	18.61	22.61	23.61	21.61
7.9	17.22	18.22	20.22	22.22	20.22	22.22
8	18.34	22.34	20.34	21.34	21.34	20.34
8.1	19.09	19.09	19.09	19.09	21.09	24.09
8.2	22.90	24.90	24.90	25.90	22.90	22.90
8.3	28.09	28.09	29.09	31.09	32.09	29.09
8.4	31.50	31.50	31.50	31.50	32.50	35.50
8.5	35.58	40.58	40.58	38.58	35.58	40.58
8.6	37.52	37.52	40.52	39.52	37.52	37.52
8.7	42.18	43.18	46.18	46.18	47.18	42.18
8.8	44.14	47.14	46.14	46.14	47.14	45.14
8.9	46.24	46.24	48.24	49.24	49.24	46.24
9	50.25	53.25	51.25	52.25	54.25	54.25
9.1	50.34	55.34	52.34	50.34	50.34	55.34
9.2	48.18	49.18	49.18	50.18	50.18	49.18
9.3	46.83	51.83	48.83	49.83	51.83	51.83
9.4	49.68	51.68	53.68	53.68	49.68	50.68
9.5	51.71	51.71	52.71	54.71	56.71	56.71
9.6	48.00	53.00	49.00	51.00	50.00	50.00
9.7	46.83	49.83	46.83	51.83	46.83	51.83
9.8	46.80	46.80	48.80	51.80	47.80	50.80
9.9	48.64	53.64	53.64	51.64	49.64	53.64
10	51.15	51.15	51.15	53.15	52.15	54.15

C.7.2 Overload PTP

Time	PTP	PTP	PTP	PTP	PTP	PTP
0.1	0.72	4.72	0.72	5.72	1.72	5.72
0.2	1.19	5.19	1.19	3.19	6.19	4.19
0.3	3.33	3.33	8.33	4.33	4.33	7.33

Time	PTP	PTP	PTP	PTP	PTP	PTP
0.4	6.45	7.45	9.45	8.45	9.45	8.45
0.5	8.44	12.44	8.44	10.44	9.44	12.44
0.6	10.52	12.52	14.52	15.52	13.52	12.52
0.7	12.72	16.72	17.72	12.72	16.72	14.72
0.8	15.95	15.95	17.95	16.95	19.95	19.95
0.9	16.61	17.61	20.61	18.61	17.61	21.61
1	18.25	22.25	19.25	19.25	23.25	23.25
1.1	20.47	24.47	20.47	23.47	24.47	20.47
1.2	23.29	26.29	26.29	23.29	24.29	27.29
1.3	25.43	28.43	30.43	29.43	29.43	27.43
1.4	29.00	30.00	34.00	32.00	34.00	33.00
1.5	32.66	32.66	33.66	32.66	36.66	35.66
1.6	37.20	39.20	40.20	40.20	42.20	40.20
1.7	38.88	40.88	39.88	39.88	39.88	41.88
1.8	42.52	42.52	47.52	42.52	42.52	46.52
1.9	45.89	45.89	46.89	47.89	49.89	45.89
2	48.93	49.93	50.93	52.93	52.93	53.93
2.1	46.42	49.42	46.42	48.42	50.42	48.42
2.2	48.10	51.10	48.10	52.10	52.10	48.10
2.3	45.98	48.98	50.98	50.98	47.98	50.98
2.4	44.35	48.35	44.35	45.35	48.35	47.35
2.5	47.90	48.90	50.90	47.90	50.90	49.90
2.6	49.27	49.27	53.27	49.27	51.27	53.27
2.7	48.78	53.78	50.78	51.78	50.78	51.78
2.8	43.75	43.75	44.75	45.75	47.75	44.75
2.9	46.99	51.99	51.99	50.99	48.99	46.99
3	48.64	48.64	50.64	51.64	48.64	48.64
3.1	46.85	49.85	46.85	51.85	49.85	51.85
3.2	50.20	53.20	54.20	52.20	54.20	53.20
3.3	51.36	51.36	54.36	56.36	54.36	54.36
3.4	49.81	49.81	52.81	52.81	49.81	53.81
3.5	48.92	50.92	53.92	50.92	49.92	48.92
3.6	45.36	49.36	49.36	48.36	49.36	49.36
3.7	49.54	51.54	53.54	53.54	54.54	52.54
3.8	49.90	54.90	53.90	49.90	54.90	49.90
3.9	48.10	52.10	53.10	49.10	53.10	52.10
4	44.93	44.93	49.93	48.93	47.93	49.93
4.1	42.87	43.87	47.87	43.87	45.87	47.87
4.2	46.37	46.37	51.37	47.37	50.37	47.37
4.3	48.71	53.71	53.71	48.71	49.71	50.71
4.4	46.89	51.89	49.89	48.89	48.89	48.89
4.5	51.50	53.50	53.50	51.50	56.50	51.50
4.6	49.16	49.16	54.16	53.16	53.16	53.16
4.7	48.04	50.04	52.04	49.04	51.04	48.04
4.8	46.18	47.18	46.18	49.18	50.18	50.18
4.9	47.57	48.57	50.57	50.57	48.57	52.57
5	50.63	51.63	50.63	55.63	50.63	51.63
5.1	51.40	51.40	53.40	52.40	51.40	55.40
5.2	53.61	55.61	58.61	53.61	54.61	55.61
5.3	49.73	50.73	49.73	53.73	49.73	50.73
5.4	47.44	50.44	48.44	50.44	48.44	51.44

Time	PTP	PTP	PTP	PTP	PTP	PTP
5.5	51.80	54.80	55.80	51.80	53.80	52.80
5.6	52.39	54.39	53.39	56.39	52.39	54.39
5.7	46.83	48.83	47.83	51.83	50.83	46.83
5.8	51.32	55.32	52.32	53.32	56.32	52.32
5.9	52.91	56.91	52.91	55.91	52.91	54.91
6	53.56	56.56	54.56	55.56	55.56	53.56
6.1	55.17	59.17	56.17	60.17	58.17	59.17
6.2	44.40	46.40	47.40	48.40	48.40	49.40
6.3	54.75	55.75	55.75	59.75	56.75	59.75
6.4	52.34	57.34	55.34	54.34	55.34	56.34
6.5	49.19	51.19	49.19	49.19	49.19	52.19
6.6	45.21	50.21	47.21	45.21	49.21	49.21
6.7	42.87	43.87	46.87	46.87	45.87	46.87
6.8	46.37	49.37	49.37	47.37	51.37	50.37
6.9	48.71	52.71	53.71	51.71	52.71	49.71
7	46.89	49.89	46.89	48.89	46.89	46.89
7.1	51.50	52.50	51.50	55.50	54.50	55.50
7.2	49.16	54.16	54.16	51.16	53.16	49.16
7.3	48.04	50.04	48.04	51.04	52.04	49.04
7.4	46.18	46.18	49.18	49.18	48.18	49.18
7.5	47.57	48.57	47.57	47.57	50.57	49.57
7.6	50.63	50.63	55.63	54.63	54.63	55.63
7.7	51.40	56.40	56.40	53.40	55.40	56.40
7.8	53.61	58.61	56.61	56.61	55.61	54.61
7.9	49.73	52.73	49.73	51.73	54.73	52.73
8	47.44	48.44	52.44	52.44	47.44	52.44
8.1	36.60	38.60	36.60	40.60	41.60	37.60
8.2	29.20	30.20	31.20	29.20	30.20	31.20
8.3	26.30	30.30	26.30	29.30	26.30	31.30
8.4	30.80	35.80	33.80	31.80	35.80	33.80
8.5	35.58	35.58	40.58	38.58	39.58	40.58
8.6	37.52	41.52	39.52	37.52	41.52	37.52
8.7	42.18	44.18	42.18	42.18	44.18	47.18
8.8	44.14	46.14	44.14	49.14	47.14	47.14
8.9	46.24	47.24	46.24	49.24	46.24	50.24
9	50.25	52.25	53.25	52.25	54.25	54.25
9.1	50.34	52.34	55.34	51.34	53.34	54.34
9.2	48.18	51.18	53.18	50.18	50.18	52.18
9.3	46.83	46.83	48.83	49.83	46.83	46.83
9.4	49.68	50.68	54.68	50.68	51.68	52.68
9.5	51.71	53.71	52.71	51.71	55.71	54.71
9.6	48.00	50.00	52.00	48.00	49.00	53.00
9.7	46.83	47.83	48.83	47.83	50.83	49.83
9.8	46.80	50.80	46.80	51.80	51.80	47.80
9.9	48.64	53.64	49.64	51.64	53.64	53.64
10	51.15	53.15	55.15	55.15	54.15	53.15

C.8 BD-B levels during trunk overload

C.8.1 BD-B levels PTP

Time	PTP	PTP	PTP	PTP	PTP	PTP
0	50	61	72	62	75	78
0.07	72.4	105	97	126	133	139
0.14	171	174	65	174	186	195
0.21	296	318	114	324	339	354
0.28	282	272	174	271	268	267
0.35	276	290	319	293	289	285
0.42	248	277	275	287	298	304
0.49	123	134	285	142	148	152
0.56	80	74	287	97	100	103
0.63	73	68	140	75	88	86
0.7	60	67	86	72	69	71
0.77	73	77	68	82	87	86
0.84	127	154	67	174	181	184
0.91	194	180	75	202	213	213
0.98	287	273	164	285	282	284
1.05	282	274	188	309	304	309
1.12	293	309	284	167	317	325
1.19	152	169	283	143	173	171
1.26	152	143	308	124	141	150
1.33	82	116	165	69	133	135
1.4	82	68	148	51	76	90
1.47	48	57	129	211	49	56
1.54	153	185	74	209	209	212
1.61	178	191	53	319	223	227
1.68	300	293	198	243	329	342
1.75	230	221	200	235	246	260
1.82	230	215	307	224	234	230
1.89	194	214	231	115	226	222
1.96	87	117	230	128	129	137
2.03	110	108	210	101	142	155
2.1	87	85	115	49	106	114
2.17	65	55	120	90	52	49
2.24	37	69	93	120	99	94
2.31	115	112	51	207	121	125
2.38	197	194	77	339	213	215
2.45	310	336	108	319	349	353
2.52	299	302	202	216	332	336
2.59	177	206	331	45	221	234
2.66	22	50	317	13	47	44
2.73	5	2	207	81	26	39
2.8	54	84	47	28	96	109
2.87	20	27	6	61	37	46
2.94	50	47	84	114	63	62
3.01	120	123	33	129	125	137
3.08	94	129	46	228	128	130
3.15	200	229	119	283	241	249

Time	PTP	PTP	PTP	PTP	PTP	PTP
3.22	299	287	132	246	283	295
3.29	260	250	224	268	255	265
3.36	240	255	282	222	277	282
3.43	220	227	245	207	219	234
3.5	176	179	254	107	213	215
3.57	82	117	227	86	120	117
3.64	62	95	194	97	100	105
3.71	87	102	112	159	105	114
3.78	120	134	90	279	172	182
3.85	228	257	101	289	284	280
3.92	299	286	146	319	285	282
3.99	293	297	270	284	317	318
4.06	287	284	282	285	282	296
4.13	256	272	307	75	293	293
4.2	68	66	283	186	81	94
4.27	186	189	286	149	197	208
4.34	143	142	79	72	144	159
4.41	37	56	190	57	75	70
4.48	56	45	150	75	58	55
4.55	50	72	61	151	84	87
4.62	110	140	50	282	157	152
4.69	293	278	78	307	296	296
4.76	310	299	140	279	310	306
4.83	260	275	281	308	293	300
4.9	282	293	301	263	322	319
4.97	225	256	273	193	267	268
5.04	177	174	307	67	188	184
5.11	50	43	267	66	81	86
5.18	34	66	189	47	70	80
5.25	28	34	57	133	60	72
5.32	124	124	61	212	134	131
5.39	206	210	44	291	221	227
5.46	240	275	135	309	298	296
5.53	276	307	215	217	310	310
5.6	177	204	278	151	213	215
5.67	110	124	307	78	157	154
5.74	60	70	207	85	89	87
5.81	56	67	139	54	94	107
5.88	30	30	65	93	58	53
5.95	73	83	82	103	93	99
6.02	68	76	41	154	98	112
6.09	124	154	92	219	153	168
6.16	225	221	91	314	229	230
6.23	276	302	156	275	320	318
6.3	245	262	218	111	278	289
6.37	90	101	305	132	121	116
6.44	120	112	273	132	131	126
6.51	113	134	114	172	133	128
6.58	115	147	126	135	173	178
6.65	104	131	135	184	146	145
6.72	163	173	157	271	195	193

Time	PTP	PTP	PTP	PTP	PTP	PTP
6.79	223	254	140	277	271	281
6.86	275	270	182	341	278	279
6.93	318	339	269	198	339	341
7	154	187	270	122	204	219
7.07	104	101	343	96	130	141
7.14	56	85	183	53	103	110
7.21	48	43	109	90	60	70
7.28	73	86	84	120	103	101
7.35	138	123	53	105	129	143
7.42	85	98	89	271	109	115
7.49	248	258	119	380	276	275
7.56	340	357	94	144	393	391
7.63	120	123	259	112	144	152
7.7	98	100	371	105	115	128
7.77	77	90	134	69	103	118
7.84	62	58	111	53	74	89
7.91	48	40	105	41	65	68
7.98	42	38	56	101	53	62
8.05	99	88	44	164	107	104
8.12	160	162	44	244	166	179
8.19	214	242	95	287	252	266
8.26	254	283	167	317	299	312
8.33	299	301	247	104	332	331
8.4	90	114	280	114	118	114
8.47	82	94	303	94	120	128
8.54	90	87	109	71	107	113
8.61	45	74	106	56	66	70
8.68	60	46	87	70	60	71
8.75	90	76	71	247	66	73
8.82	248	246	42	278	247	251
9.03	276	266	73	340	281	296
9.1	314	341	244	286	345	340
9.17	293	290	268	165	294	300
9.24	177	172	341	137	162	163
9.31	127	119	288	121	149	152
9.38	87	105	169	90	118	132
9.45	76	77	126	68	105	100
9.52	42	61	116	90	78	74
9.59	48	83	76	290	92	105
9.66	242	275	63	303	297	300
9.73	307	300	88	260	310	319
9.8	270	262	275	253	255	254
9.87	230	247	300	127	263	263
9.94	120	132	259	138	126	139
10.01	104	128	256	110	151	156

C.8.2 BD-B levels PM

Time	PM	PM	PM	PM	PM	PM
0	50	54	66	74	60	54
0.07	72	70	82	90	102	102
0.14	171	179	174	183	191	180
0.21	307	294	294	296	295	282
0.28	112	108	116	123	134	126
0.35	2	4	6	7	8	12
0.42	8	11	16	19	12	16
0.49	10	12	14	20	21	29
0.56	1	7	8	8	13	19
0.63	112	99	88	80	68	73
0.7	153	165	171	162	150	150
0.77	194	203	203	202	206	218
0.84	330	341	334	336	336	327
0.91	90	96	111	111	103	95
0.98	2	3	3	8	11	17
1.05	8	12	14	17	17	30
1.12	10	14	15	27	17	30
1.19	1	3	7	9	9	9
1.26	112	98	110	113	100	105
1.33	153	150	161	170	180	192
1.4	178	192	196	186	193	196
1.47	301	287	279	269	274	264
1.54	196	205	220	206	198	197
1.61	29	19	19	33	39	52
1.68	10	11	15	26	22	12
1.75	1	7	13	14	21	19
1.82	6	6	13	17	15	18
1.89	1	2	6	11	11	13
1.96	17	21	32	23	10	12
2.03	94	88	92	96	111	111
2.1	178	167	173	186	179	168
2.17	318	320	319	307	315	330
2.24	66	57	62	56	49	52
2.31	1	5	7	11	16	5
2.38	22	23	14	21	28	28
2.45	1	3	4	9	13	13
2.52	2	6	8	14	17	29
2.59	1	7	9	12	19	31
2.66	50	49	46	58	60	65
2.73	94	99	88	84	80	72
2.8	200	215	214	211	213	228
2.87	350	365	361	376	377	370
2.94	111	123	135	130	119	126
3.01	1	7	10	17	23	17
3.08	10	15	24	26	21	33
3.15	1	4	6	9	13	16
3.22	10	13	15	30	45	49
3.29	1	7	12	14	15	18
3.36	108	108	115	105	109	122

Time	PM	PM	PM	PM	PM	PM
3.43	167	160	149	149	145	158
3.5	220	231	229	238	230	227
3.57	312	307	311	317	328	328
3.64	256	269	262	249	259	253
3.71	1	5	6	7	14	17
3.78	6	7	8	8	10	11
3.85	4	8	13	20	6	8
3.92	4	4	7	13	19	13
3.99	1	4	11	16	2	3
4.06	50	47	39	31	41	38
4.13	90	101	91	77	74	60
4.2	173	184	188	177	187	178
4.27	334	349	349	353	345	343
4.34	153	160	159	161	162	172
4.41	60	52	60	49	57	58
4.48	10	10	10	16	17	11
4.55	4	10	16	17	23	25
4.62	1	7	9	14	15	20
4.69	28	21	30	40	33	48
4.76	75	81	94	86	85	95
4.83	122	136	149	142	132	138
4.9	210	211	201	195	199	190
4.97	284	297	295	293	306	292
5.04	131	124	110	97	87	89
5.11	1	5	11	13	16	4
5.18	3	9	10	13	19	26
5.25	1	5	9	13	15	18
5.32	4	10	17	14	20	14
5.39	1	3	7	10	13	18
5.46	50	55	57	58	66	81
5.53	149	138	138	130	135	123
5.6	269	263	253	255	250	249
5.67	345	341	332	323	335	342
5.74	90	95	97	91	106	117
5.81	1	8	11	13	16	12
5.88	4	5	8	14	18	22
5.95	1	4	10	12	13	15
6.02	1	2	3	5	10	11
6.09	1	8	12	16	16	20
6.16	123	134	120	114	110	96
6.23	227	216	230	237	227	213
6.3	318	330	321	329	338	353
6.37	165	164	166	163	166	171
6.44	55	48	46	57	49	50
6.51	1	4	7	13	16	15
6.58	10	11	15	23	30	29
6.65	1	2	8	15	19	29
6.72	50	40	52	65	65	62
6.79	85	82	83	80	78	84
6.86	184	192	206	221	220	234
6.93	350	339	332	334	335	350

Time	PM	PM	PM	PM	PM	PM
7	120	129	129	118	132	142
7.07	2	9	13	20	13	17
7.14	8	13	13	18	27	28
7.21	19	18	25	32	24	25
7.28	1	5	8	10	14	20
7.35	131	145	138	139	128	120
7.42	208	212	223	238	238	238
7.49	245	239	252	254	257	245
7.56	330	328	321	322	315	315
7.63	90	76	88	97	87	86
7.7	2	2	4	11	16	21
7.77	30	21	34	36	22	19
7.84	10	13	20	8	14	16
7.91	1	8	11	13	17	19
7.98	112	121	122	112	100	94
8.05	143	154	161	160	146	148
8.12	194	199	191	177	186	184
8.19	314	321	323	317	316	327
8.26	174	174	160	159	152	165
8.33	29	15	18	29	19	30
8.4	10	11	18	12	15	12
8.47	1	1	4	7	11	16
8.54	6	6	9	9	13	13
8.61	1	5	12	19	10	13
8.68	5	9	15	17	7	13
8.75	94	101	98	85	80	89
8.82	343	349	349	344	347	340
9.03	186	183	195	194	181	192
9.1	67	61	66	65	62	75
9.17	10	17	8	8	13	14
9.24	1	7	7	9	15	21
9.31	6	13	16	14	21	15
9.38	7	7	14	14	16	16
9.45	5	10	14	20	25	13
9.52	131	120	129	139	125	135
9.59	292	292	295	292	291	292
9.66	10	10	12	16	2	2
9.73	1	5	11	16	21	27
9.8	6	6	10	12	13	14
9.87	7	12	14	21	10	10
9.94	5	5	9	12	17	24
10.01	94	95	95	87	100	108

C.9 Autocorrelation Function for Packet Trains in Applications

Application 1		Application 2	
Lag	Correlation	Lag	Correlation
1	0.837184	1	0.84353
2	0.658955	2	0.633342
3	0.560084	3	0.5342
4	0.547465	4	0.51992
5	0.487493	5	0.4782
6	0.444675	6	0.42145
7	0.416466	7	0.41097
Application 3		Application 4	
Lag	Correlation	Lag	Correlation
1	0.64353	1	0.74353
2	0.633342	2	0.633342
3	0.5342	3	0.523154
4	0.494361	4	0.412966
5	0.439696	5	0.302778
6	0.385031	6	0.19259
7	0.330366	7	0.082402
Application 5			
Lag	Correlation		
1	0.802349		
2	0.6931		
3	0.571363		
4	0.559912		
5	0.520517		
6	0.467491		
7	0.443037		

Appendix D - Publications

Published Papers:

Traffic Profiles and Application Signatures,

by Georg Mueller, Peter Sanders, John Allen

Computer Communications

August 1999

Strategies for Content Migration on the World Wide Web

by M.P Evans, A.D. Phippen, G. Mueller, S.M Furnell, P.W. Sanders and P.L.

Reynolds

Internet Research

Volume 9, Number 1, 1999

Awaiting Publication:

Traffic Characteristics of WWW Sessions

by Georg Mueller , Peter Sanders and Garry Joyce

Electronic Letters - IEE

The following paper was published internally at AT&T and is submitted for publication

A New Methodology on Overbooking in Frame Relay Networks

by Georg Mueller, Peter Sanders, John Allen

AT&T Technical Journal USA

Conference:

Content Migration on the WWW

by M.P Evans, A.D. Phippen, G. Mueller, S.M Furnell, P.W. Sanders and P.L.

Reynolds

International Network Conference, Sponsored by IEE

1998, Plymouth

Traffic Profiles and Application Signatures

Georg Mueller, Peter Sanders*, John Allen***

** SECEE, University of Plymouth, UK*

*** Netscient Limited, UK*

Keywords: Traffic Profiles, Application Signatures

Our hypothesis is that data applications leave a signature when used over networks. Data that is sent from the original source and travels to the destination utilises the network in a very similar manner, if no strong congestion is experienced. Should the network experience congestion, the unavoidable frame losses will alter the signature of the application and change the statistical property of the data. However, certain properties remain the same, even after a network disruption and congestion. These properties can together be used to create an "application signature", which is unique to the specific application. This is due to the protocols used and the parameters set in the environment of the traffic source. We think that these findings could help to understand the traffic characterisation and the processes involved. The traffic profiles could then be used for the dynamic resource allocation (e.g. Resource ReServation Protocol) in future networks as well as with the simulation of such networks.

Introduction

Data from user equipment is typically broken into multiple frames and sent according to the protocol used, in a specified manner over the network. Traditional network theories imply that the packet size distribution is exponential and often work on average packet size distribution [1]. In practise however, it is known that the packet size distribution is directly related to the traffic source, its protocol and the application over the protocol. A function of these three parameters creates a specific sequence of packet sizes as they appear on the network and a unique packet size distribution.

Previous attempts to characterise Internet arrival processes have concentrated on traffic by component, e.g., telnet and ftp. Caceres et al. [2] provide evidence that characteristics of an instantiation of a specific TCP application do not depend on the environment, but that characteristics of the conversation arrival process itself do depend on the environment. They

state that they were “unable to form a realistic and network-independent model of conversation arrivals, since the arrival parameters depend on geographic site, day of week, time of day, and possibly other factors”.

Paxson and Floyd used fifteen wide-area traces to investigate the extent to which TCP arrival processes (session and connection arrivals, ftp connection arrivals within ftp sessions, and telnet packet arrivals) are Poisson. They find that user-initiated TCP session arrivals, e.g., remote login and file transfer, reasonably reflect Poisson processes with fixed hourly rates, but other connection arrivals are less convincingly Poisson. Furthermore, they find that modelling telnet packet arrivals as exponential inaccurately reflects telnet burstiness. Finally they determine that ftp connection arrivals within ftp sessions come bunched into “connection bursts”[3]. Our opinion is that these burst can be further analysed and grouped by various characteristics, e.g. size of the bursts and sequence of packet sizes.

Another different but useful model is by Jain. He offered a *packet train* model of packet arrivals to describe traffic on a local area network at MIT [4]. He defined a *packet train* as a burst of packets arriving from the same source and heading to the same destination. If the spacing between two packets exceeds some inter-train gap, they are said to belong to separate trains. In his model, the inter-train time is a user parameter, dependent on the frequency with which applications use the network. The inter-car interval for a train is a system parameter and depends on network hardware and software.

We are mainly interested in the existence of the gaps, not the size of them, because they make it possible to analyse how large a train is. In our experiment the inter-train gap, which identifies separate trains, does not reflect a user parameter. But it is a parameter, which is defined by the way protocols break up files into segments and then series of packets, which appear as bursts. These trains can be a reflection of the windowing mechanism used by the protocols. The sequence of packet sizes can be explained in a similar way as the header of the train knows exactly how large the cars inside the train are going to be.

Our findings show that if an application is loaded and traced on a LAN, the measurements show that the LAN has a minor influence on certain characteristics. Therefore these characteristics are again directly related to the before mentioned function, and not the LAN itself. We have not investigated characteristics of a specific protocol or a mechanism like

telnet or ftp, but the application or file which can be transferred by one of these protocols. Our research has instigated the loading of the same file under various conditions. As it is impossible to make any capacity reservations on a LAN, the LAN has to be viewed as a black box and therefore as the source for the application. However, as some of the characteristics of a specific transaction are known, the LAN should be able to provide the WAN with information about the data which is going to follow. This is from the perspective of a router or switch, which has to reserve dynamic capacity for the following transaction. From the view of a router, the LAN *is* the source of the data and its input can be predicted.

Traffic Patterns

The knowledge of various network properties is very important for modelling and simulating networks. Such detailed knowledge makes it easier to predict the capacity and delay characteristics of the network.

The packet size distribution is one of the characteristics of networks. In our process we investigate the properties of packet size distribution, inter-arrival time of packets to identify packet trains and the predictability of traffic patterns of individual applications. The model we propose is just an initial set of parameters that can represent together a statistical profile of an application. This set of parameters can be expanded if parameters are found which are not changed by the underlying LAN.

In our research we focus on the most popular protocol, the Internet Protocol Suite (TCP/IP). The general characteristic of a building block is that it would identify a sequence of packets travelling from the source to the destination, which is identifiable and can assume a particular type of service or program. These packets are identifiable and belong to the same application. The application can be either a WWW page or any another repeatable network download activity or object.

Breaking traffic up into components and analysing them is inevitable; as advanced technologies will require quality of service (QOS) parameters for the application. Examining the parameters of an application at various sites is also relevant because the same applications may differ when it is located on a different site. The arrival characteristics of traffic on a regional network may differ from the same application which is used over a WAN due to time-sensitivity of the protocol and its congestion control algorithms and various reasons for

delay, e.g. propagation and congestion delay. This is mainly due to capacity limitations, which cause packet drops and change have an influence on some characteristics. If a WAN reserves enough capacity for a transaction, then the transaction looks very similar. This however is not very economical at the moment, as the properties of a transaction are not known before it happens. Reserving too much capacity on the other hand will lead to wasted resources.

However, we feel it will be more and more important to characterise the aggregate arrival process in relation to transport protocol and application. This approach will be increasingly relevant as different types of Internet traffic proliferate, decreasing the proportion of traffic carried by traditional protocols. As a larger part of the data on the Internet, at the present time, results from WWW traffic, the tests were carried out on a WWW page.

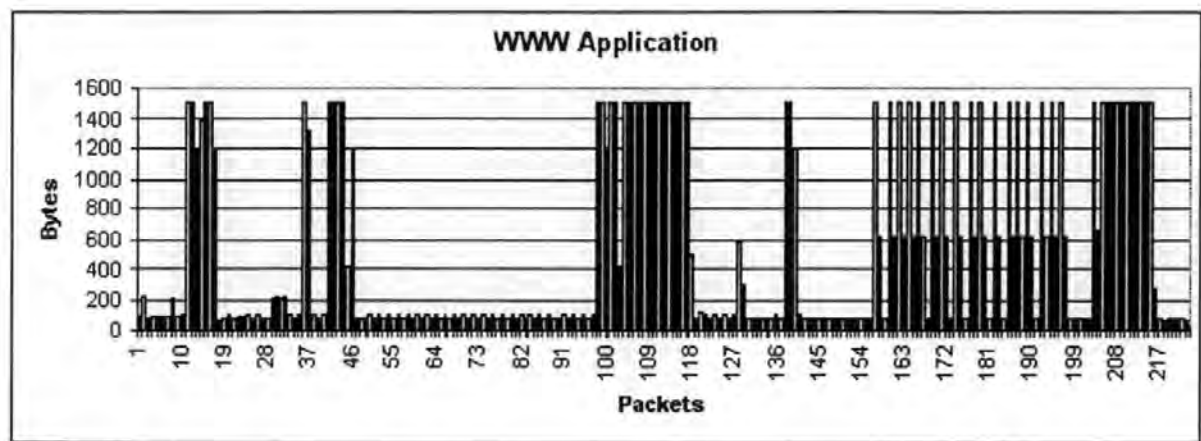
Our measurements indicate that, at least in the environments we studied, current IP traffic consists of more short transaction type traffic rather than longer-term flows. The short packets and short flows together shed doubt on a strategy of optimising for long flows that are in fact the minority case. However, we note that many new applications may change this characteristic, as these new applications may introduce traffic flows with different behaviour, particularly real-time continuous media flows, which tend to exhibit greater duration and flow volume. The principal we introduce can be used for different types of traffic, as long as the traffic is repeatable. This can be short transactions like downloading a specific Text file, graphics or stored movies. We think the movie on demand applications that will be used by television companies can use our method, as the stored movie will be transmitted to many places many times on different occasions.

Visual Study

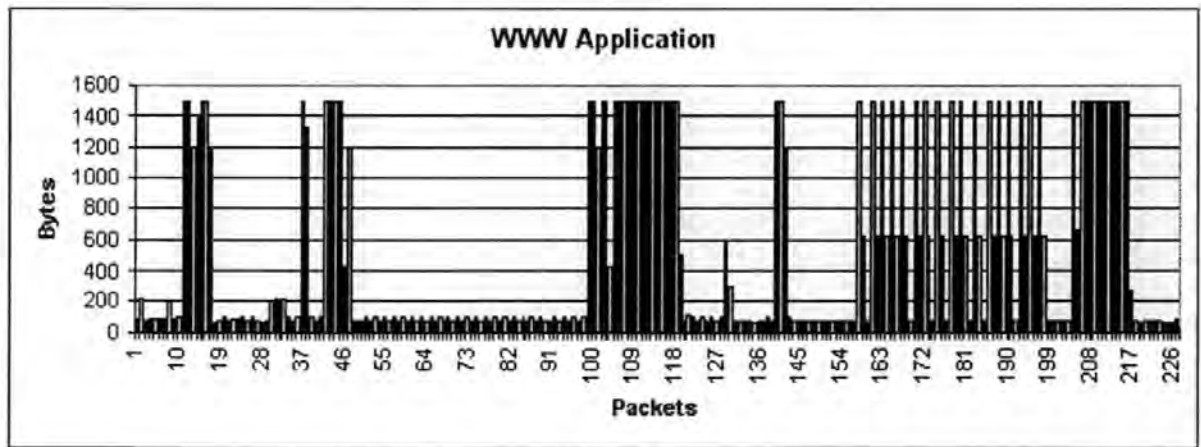
The aim in this work was to investigate the validity of claims for a 'data fingerprint', and to do so via a visual study of data traffic graphs and simple statistics of real life applications. To do this we have used different applications that have been downloaded over a LAN. As the traffic usually originates from a source connected to a LAN, the most original form of the traffic profile can be found very "close" to the source. Therefore "signatures" could be defined by parameters like packet size distribution, packet size sequence, and the responder/originator ratio. These parameters and therefore the fingerprints are in the purest form, when they are loaded and recorded on the LAN.

All the following graphs are representing the same WWW page, which has been downloaded under various traffic conditions on the LAN. The utilisation on the LAN varied from 5%, 20%, and 30%. It should be noted that a number of applications and WWW pages have been measured and have shown the same effects as the one represented below. Please note that the graphs look very similar, even under the various utilisation levels. This is one reasoning for the argument, that an application has a unique fingerprint.

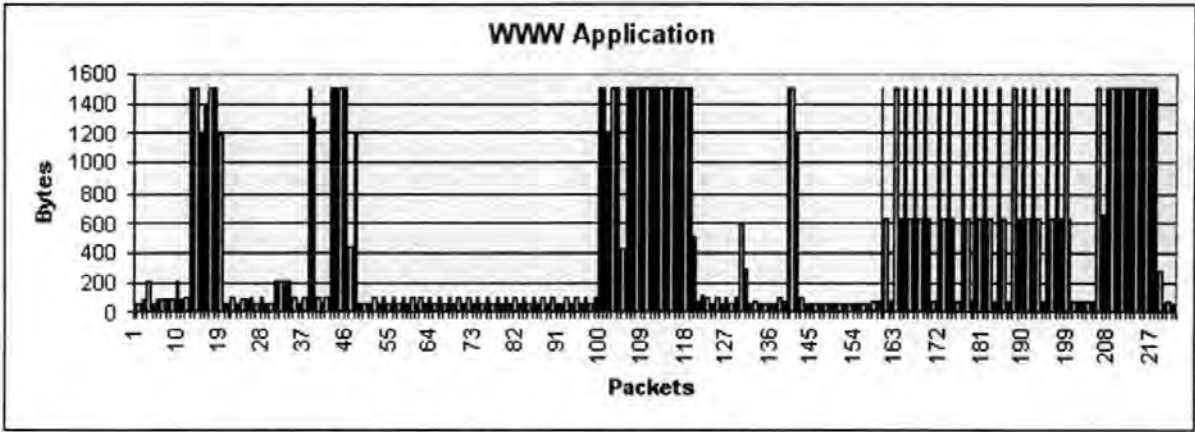
In the following, it can be seen how similar graphs 1,2 &3 look. Even with a big utilisation difference, no major difference in the graphical outlook can be found. The short utilisation variations obviously have little or no effect on the fragmentation of the packets.



Graph 1: Measurement at 5% Utilisation



Graph 2: Measurement at 20% Utilisation

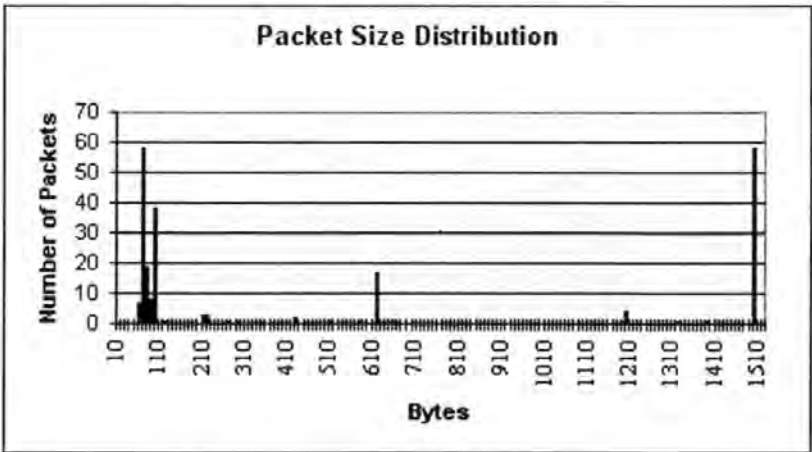


Graph 3: Measurement at 30% Utilisation

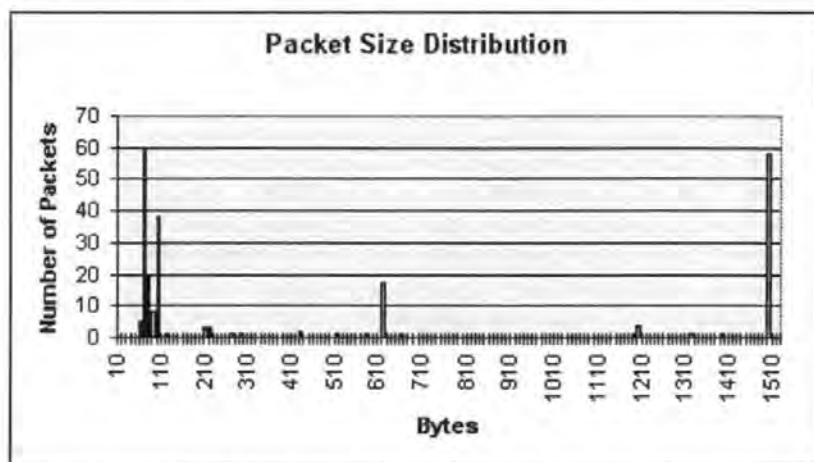
Graphs 1,2 and 3 show very similar traffic traces. All three graphs are parts of a fingerprint of the same application, which was downloaded three times, at various LAN utilisation levels.

Packet Size Distribution

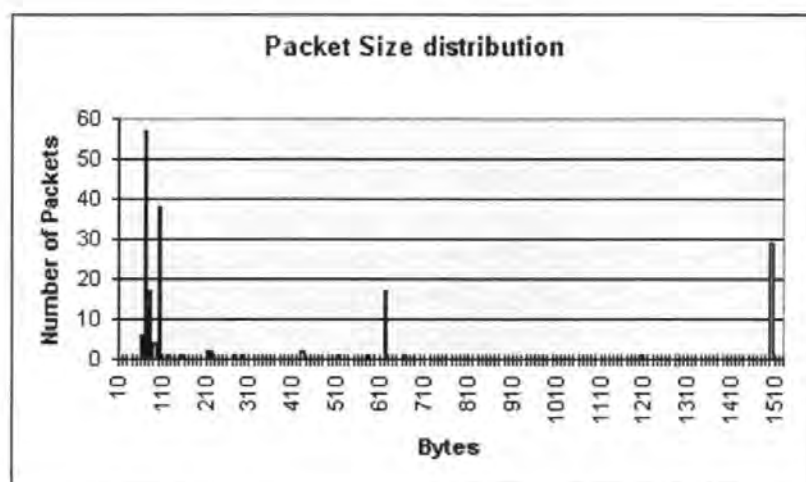
The packet size distributions are shown in graphs 4,5, and 6. It can be seen that the packet size distribution is very similar. One important point is that it is easy to assume that the number of bytes generated by the application originator equates to the number of packets generated. This assumption is generally erroneous. However, we found that the number of packets with the same length was very much the same when the test were conducted.



Graph 4: Measurement at 5% Utilisation



Graph 5: Measurement at 20% Utilisation



Graph 6: Measurement at 30% Utilisation

Graphs 4, 5 and 6 show the packet size distribution of the same application under various utilisation levels. The distributions are very similar, and applies therefore as a parameter for application signature.

Responder/ Originator Ratio

Using this model for traffic profiles for prediction of traffic we have to investigate the relationship between the distributions of the originating source and responder. We would often like to know how many responder bytes to expect given a particular application. Previous research of telnet session has shown a ration of 20:1 between the bytes generated by the computer in a remote login session and those generated by the user. Our research differs in the way that every application seems to have a distinct ratio, which is hardly changing. This ratio can vary by application from 5:1 to 40:1. In any case however, the same application created the same ratio in all trials.

Summary

We have presented a number of graphs and statistics, which confirm our hypothesis, that objects or applications have certain characteristics when downloaded on a LAN. This behaviour varies from application to application but finds very similar characteristics when one application is used at different utilisation levels or by different users.

We have investigated some possible parameters of what could make up an application signature. Obviously there are no general signatures for applications at present, which could be applied without knowing the exact makeup of the source, the downloadable object and the protocols involved. However, when the object is downloaded various times a picture is created in form of parameters that hardly change. We think that it should be possible to create a "scanner", which is able to perform a simulation on an object or file, to create these parameters, without loading and tracing it on the network. The results of the scanner could be stored with the file or object so it could be used by the capacity reservation protocol when the application is loaded. This scanner could be a running locally or over the network, storing the information on a central signature server.

The essence of the argument presented in this paper is that while traffic cannot be modelled exactly, some of its properties can be used for good approximations in predictions of traffic levels and therefore reduces congestion. Furthermore, these models can be used in simulations to look at other aspects of WANs such as congestion control.

Also some of our findings suggest that not only applications but also user-groups have specific network behaviour which influences network characteristics.

References

- [1] Kershenbaum, A. (1993) Telecommunications Network Design Algorithms, McGraw-Hill
- [2] Caseres R., Danzig P., Jamin S & Mitzel J. (1991) Characteristics of Wide Area TCP/IP Conversations, Proceedings of the 1991 ACM SigComm Conference
- [3] Paxson V. & Floyd S. (1995) Wide Area Traffic: The Failure of Poisson Modelling, IEEE/ACM Transactions on Networking No.3 June

[4] Jain, R. & Routhier R. (1986) Packet Trains, Measurements and a New Model for Computer Network Traffic, IEEE Journal on Selected Areas in Communications, SAC-4, No. 6.

Strategies for content migration on the World Wide Web

M.P. Evans

A.D. Phippen

G. Mueller

S.M. Furnell

P.W. Sanders and

P.L. Reynolds

The authors

M.P. Evans, G. Mueller, S.M. Furnell, P.W. Sanders and P.L. Reynolds are all at the Network Research Group, School of Electronic, Communication and Electrical Engineering, University of Plymouth, Plymouth, UK.

A.D. Phippen is at the School of Computing, University of Plymouth, Plymouth, UK.

E-mail: Mike/Evans@jack.see.plym.ac.uk

Keywords

Distributed data processing, Distribution, Internet

Abstract

The World Wide Web has experienced explosive growth as a content delivery mechanism, delivering hypertext files and static media content in a standardised way. However, this content has been unable to interact with other content, making the Web a distribution system rather than a distributed system. This is changing, however, as distributed component architectures are being adapted to work with the Web's architecture. This paper tracks the development of the Web as a distributed platform, and highlights the potential to employ an often neglected feature of distributed computing: migration. Argues that all content on the Web, be it static images or distributed components, should be free to migrate according to either the policy of the server, or the content itself. The requirements of such a content migration mechanism are described, and an overview of a new migration mechanism, currently being developed by the authors, is presented.

Introduction

The World Wide Web (the Web) is a platform for distributing software resources across the Internet, which are then presented as rich, consistent content by applications on the client (usually a browser). The three main standards which define the platform are:

- (1) the Uniform Resource Locator (Berners-Lee *et al.*, 1994);
- (2) HyperText Transfer Protocol (Berners-Lee *et al.*, 1996);
- (3) HyperText Markup Language (Berners-Lee *et al.*, 1995).

The Uniform Resource Locator (URL) is used to locate software resources; the HyperText Transfer Protocol (HTTP) is the protocol used to distribute the resources; and HyperText Markup Language (HTML) is used to present the information contained within the software resources in a consistent way across all computer platforms.

Consequently, today's Web is a large distribution system. The software resource is a single, self-contained unit of data (usually a binary or text file), which the Web can locate (using the URL) and distribute (using HTTP). It encodes content, which is presented on the client by applications according to the media type the content represents (e.g. images, video, etc.). Each media type must conform to its own universal standard, which is not part of the specification of the Web itself, but which contributes to its ubiquity and openness. The content is decoded from the software resource by the browser or its own application (generally termed a "viewer" or "plug-in"), and is presented consistently across all platforms according to the layout and style specified by the HTML page. For example, the graphics interchange format (GIF) standard, developed by CompuServe, is a standard format for compressing and encoding images. A GIF viewer is an application which works inline with the browser to interpret a GIF image file and display the image it contains. This GIF viewer essentially reads in a generic, platform-independent file (the software resource) which contains an encoding of the image, and converts the encoded data into content: platform-dependent information which can be displayed on the client's screen as the decoded image in a consistent way across all

platforms according to the layout and style specified by the HTML. The same can also be said for other content formats (e.g. JPEG, MPEG, AVI, QuickTime), each of which encodes a specific media type according to the media's own defined standard. In fact, for any type of content to proliferate on the Web, it must have its own platform-independent standard with its own platform-specific viewers generally available on every platform. To the Web's distribution mechanism (i.e. the Web servers and HTTP), everything is a generic software resource (see Figure 1). Only when the correct application receives it on the client does it become content.

Static and intelligent content

Web content has traditionally consisted of static files without functionality, and without the ability to interact with other software resources. A GIF file, for example, contains the information required to display the image it encodes with a suitable viewer, but there is no computational intelligence contained within it; consequently, the user cannot interact with it (the use of "image maps" within a browser, whereby a user can click on an image to navigate to another page, is controlled and formatted by the HTML in the Web page, not the image file). Currently, then, the Web is a distribution system, not a distributed system. However, this is changing. As the Web matures, its functionality is increasing, and, more important, the intelligence contained within the resources it is currently distributing is growing along with the Web itself. To distinguish between resources which contain some form of static media

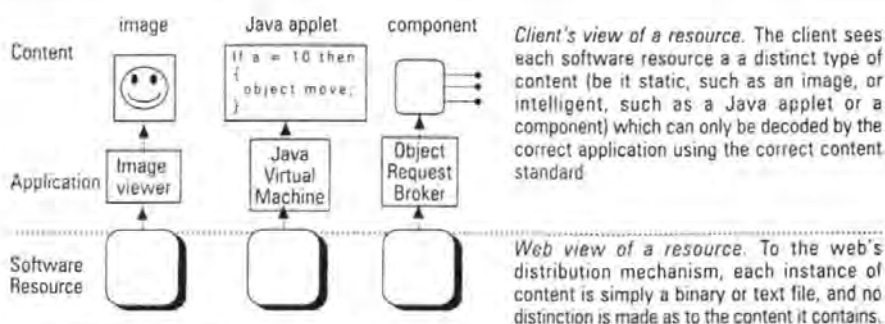
content (such as an image), and resources which have some form of computational intelligence as part of their content (such as a Java applet), this paper will define the terms static content and intelligent content, respectively.

Intelligent content currently consists of small self-contained blocks of code which reside on a server as software resources, and are downloaded onto a client machine, where they are executed by a suitable application, usually inline with an HTML page. Java applets are an example of such content, as are Microsoft's ActiveX controls. This type of content is limited, however, by its self-contained nature: a Java applet, for example, cannot communicate with other Java applets on machines other than the server it originated from. In order to distribute the intelligence of a large scale application, the components of the application must be able to interact with each other across a distributed environment; to achieve this, a distributed component architecture must be employed.

Distributed components

Component software develops on the potential of object-based software by constructing software from components which encapsulate functionality and data. This is similar to object orientation, but allows dynamic component interaction at runtime (known as "runtime reuse"). This is achieved through the use of a component architecture, which is a defined platform with rules for interaction between components. Any component developed on top of this platform will be able to interact with any other component built on the same platform. While a general component architecture enables

Figure 1 Relationship between content and the software resource



components on the same computer to interact, distributed component architectures add to the functionality by enabling interaction across a distributed network environment. A client component can use the services of components not only on its host machine, but also any other machine which supports the distributed architecture. Components within such architectures are also termed distributed objects, primarily because the architecture itself is based on the object-oriented paradigm. Currently, the distributed component field is dominated by two major architectures:

- (1) Microsoft's distributed component object model (DCOM); and
- (2) the Object Management Group's common object request broker architecture (CORBA).

DCOM is the distributed extension to Microsoft's component object model (COM), and extends the basic functionality to incorporate transparent network distribution and network security mechanisms into the architecture. Through DCOM, ActiveX controls can interact with one another, and with other COM-based components, across a network.

CORBA is a complete architectural specification developed by the Object Management Group (OMG, 1995) which specifies both a component architecture, and component services. CORBA is entirely generic, defining platform-independent data-types, protocols for communication across platforms, and a number of platform-independent services which provide the components with a number of useful services such as security, transaction processing, and naming and location services for finding components across a distributed system. CORBA's functionality is implemented through an object request broker (ORB), which provides the transparencies required by the architecture.

Both architectures offer the developer similar features and similar benefits. They both provide a component distribution mechanism employing network and location transparency, marshalling mechanisms, etc., and both expose functionality through language-independent interfaces. They are reliable distributed platforms on which large scale distributed applications can be built. Such distributed component systems are increasingly being incorporated into

the Web. Distributed components are becoming the next type of software resource to share server space with existing types of static and intelligent content. This allows the Web to become a true distributed system, being able to provide distributed applications and services via a client's browser. Netscape, for example, has integrated CORBA functionality into its Communicator 4.0 browser, allowing it to interact with CORBA components on CORBA-enabled servers. Equally, Microsoft's Internet Explorer 4.0 browser is DCOM-enabled, allowing it to communicate with DCOM components on DCOM-enabled servers. In this way, the Web is evolving into a complete distributed system, termed the "object Web" (Orfali *et al.*, 1996) to reflect the object-based nature of the distributed architectures being employed.

Content migration

An overview of migration in a distributed system

One of the benefits of a distributed system is the ability of an application to be distributed across multiple hosts across a network, in such a way that no one host has to execute the whole application on its own. With a fast enough network, this "load balancing" functionality can greatly increase the efficiency and performance of the application in a way which is entirely transparent to the client machine. However, the drawback to this distributed paradigm is the static nature of the location of each component. Once a component has been installed on a host, it cannot easily be moved to another host. Thus, should the host's, or its network's, performance degrade in any way, access to the component will be affected. Invocations on the component's interfaces will be slowed down, which in turn will affect the performance of the application as a whole. The component can be manually relocated to a different host, but this is time-consuming. Most distributed applications comprise many components, and it would be impractical to manually redistribute them all whenever necessary.

Consequently, various automatic component relocation mechanisms exist. These "migration mechanisms" can transparently move a component from one host to another in such a way that the client has no awareness of the move. These

mechanisms are provided by some (though not all) distributed architectures as a way of dynamically relocating components to provide load balancing and fault tolerance. Distributed component architectures can migrate entire components, including their functionality and data, and retain the state of the component from one machine to another.

The problems with distributed components and the WWW

The distributed component is a new type of intelligent content, which has the ability to interact with other content of the same type. However, components of different architectures cannot directly communicate with each other. Thus, Netscape's CORBA-compliant browser cannot use DCOM components, and Microsoft's DCOM-enabled browser cannot use CORBA components. Thus, neither component architecture provides its content (the distributed component) with true ubiquity across the Web in the way in which traditional content does.

This problem affects the architectures' use of migration. Most, including DCOM and Enterprise JavaBeans, do not support migration at all. However, even if they did, current distributed architectures cannot successfully employ a ubiquitous migration mechanism across the Web, because no matter how open they are, the type of resource that can be migrated is tied too closely to the architecture itself. The Web treats each software resource as a generic unit. The URL is used to reference it, and HTTP to distribute it, regardless of the resource's content type. In contrast, distributed architectures work only with their own content, and use their own reference formats to locate the components. Thus, only components created specifically to an architecture's specifications can be migrated, and only if both hosts involved in the migration support the architecture. Currently, however, the vast majority of content on the Web today consists of JPEG and GIF files, and Java applets, which have no concept of a distributed architecture, much less the services that one can provide. Equally, servers supporting CORBA or DCOM are uncommon, leaving very few places for a component to physically migrate to.

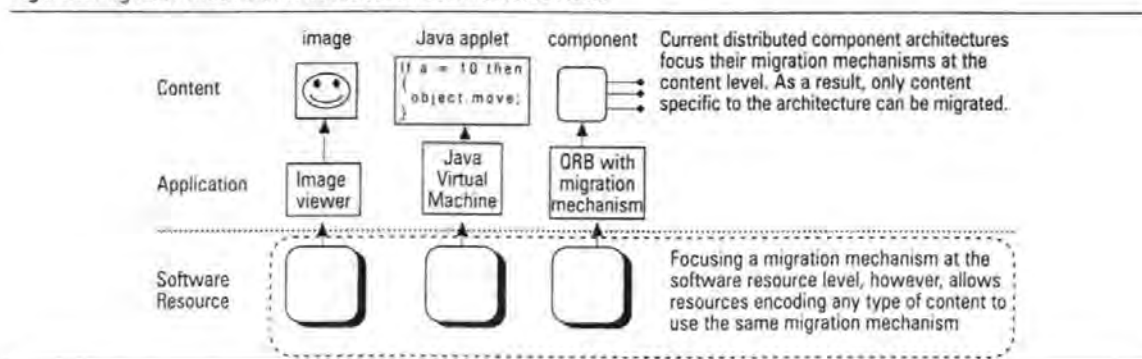
Requirements for a migration mechanism on the WWW

For a migration mechanism to be successful on the Web, then, it must recognise the diverse range of content that exists. Therefore, it must be completely decoupled from the content that it can migrate, and instead focus on the software resource: a generic unit of data which may or may not be aware of the mechanism (see Figure 2). Additionally, to truly be of benefit, the mechanism must fit in with the existing Web architecture, rather than build its own set of standards on top of the existing Web platform. In this way, it can be used by existing Web content as much as by intelligent content such as distributed components, and can provide services which distributed components can use to enable the Web to become a distributed platform.

True content migration, then, where content of any type can be freely migrated, relies on implementing migration at the resource level. In order to achieve this, the following set of requirements for a resource-level migration mechanism have been identified:

- *Universal client access.* The mechanism must be accessible to clients of any type and should not require clients to be altered in order to use it. Thus, existing software does not need to be rewritten, and future software will not require any extra facilities in order to use it.
- *Content neutrality.* A Web-based mechanism must be completely decoupled from the content it can migrate, enabling it to migrate all resources, no matter what type of content they encapsulate (see Figure 2).
- *Full integration with the Web's current architecture.* The mechanism must reuse as much of the Web's existing architecture as possible. Specifically, this means the reuse of HTTP and the URL. There is too much investment in the infrastructure supporting HTTP to change it overnight, and the URL is becoming accepted by the public as the only way to navigate to Web resources. With businesses now using the URL as part of their advertising campaigns, URLs can now be recognised even by people without access to the Web.
- *Practical design.* Resource migration can be technically achieved in many different ways, but adopting a practical approach means focusing on the requirements of Web

Figure 2 Migration mechanisms at the content and resource levels



developers, existing Web software, and (most important) Web users, rather than focusing on a technically optimal design. A practical design also means one that takes into account the dynamics and characteristics of the Web (and, by implication, the users of the Web); an approach that technically works will not achieve ubiquity if it results in the Web appearing to run more slowly.

In the next section, this set of requirements will be used to evaluate existing approaches to migration to see which is best suited to the development of a migration mechanism for the Web.

Developing a Web-based migration mechanism

Methods of resource migration

For any migration mechanism, there are four different methods through which a resource can be tracked once migration has occurred (Ingham *et al.*, 1996). These are:

- (1) Forward referencing.
- (2) Name service.
- (3) Callback.
- (4) Search.

Forward referencing

Forward referencing involves leaving behind a reference in place of the migrated resource which points to the resource's new location. Thus, an object leaves behind a chain of references on each host it visits. For example, the migration mechanism of the "W3Objects" system (Ingham *et al.*, 1996), and "Voyager", from ObjectSpace (an agent-oriented CORBA implementation), both adopt this approach.

When an object migrates in Voyager, a "virtual reference" is left behind to forward messages to the new location. As an object migrates, more virtual references are created, forming long chains which eventually resolve onto the object itself. The W3Objects approach is similar, in that "forward references" are created each time a resource migrates; however, to prevent long chains building up, "shortcuts" can be created which allow a reference holder (that is, a resource with a link to the migrated resource) to bypass the chain of references, and reference the resource directly.

Suitability for the Web

Voyager's mechanism is unsuitable for the Web as, like most other distributed architectures, it only migrates Voyager-aware content, and is therefore not content-neutral. Surprisingly, the mechanism used by W3Objects will also only work with its own, object-oriented resource (termed a "W3Object") and a specially-defined reference (termed a "W3reference"), and so it too is not content-neutral. Furthermore, in order to use the W3Objects system, each client's browser must be adapted to work with W3References rather than URLs.

However, the forward reference method itself is unsuitable for use on the Web. Each link in the chain of forward references adds another point of potential failure (Ingham *et al.*, 1996), and if the chain breaks, then the resource is lost completely. Further, the characteristics of the Web will make managing the chains unrealistic, as the number of forward references will increase with both time (some resources, such as autonomous agents, will migrate constantly) and space (every resource will require a chain of references to be maintained).

Name service

The name service method employs an external system to maintain references to registered resources at all times. Such mechanisms generally focus on the use of the name used to identify a resource, and attempt to abstract any location-dependent information out of the name itself. For example, the uniform resource name (URN) is a proposed standard by the Internet Engineering Task Force (IETF) for naming a resource independently from its location (Sollins and Masinter, 1994). Specifically, a URL is used to locate a resource, while a URN can be used to identify a resource (Berners-Lee *et al.*, 1994). The URN can then be mapped onto the URL through an external resolver discovery service (RDS), which maintains the location of the resource. Should the resource have migrated, the RDS will resolve the URN into a URL that points to another RDS which can resolve the URL. Thus, a chain of references is built up across the resolver service, rather than across each visited server.

Suitability for the Web

The URN identifies a resource independently from its location, and so subsumes the URL, treating it not as a name, but as a pointer to a location. Thus, while the URN has content-neutrality, it does not support full integration, as the URL cannot be used at the user level.

Also the name service method suffers from the same problems inherent with any "chain" of references, as described above. Further, the method is not practical, as it does not take into account the characteristics of the Web users: it requires, for example, that a resource's name remain invariant throughout its lifetime (which can be "for hundreds of years" (Sollins and Masinter, 1994)), but in real life, the ownership of a resource can change within its lifetime, and the new owner may wish to give the resource a new name.

Callback

The callback method relies on a resource to inform all other resources with references to it of any change in its location, in order to ensure referential integrity. The benefit of this approach is that there is no indirection, and so no chain of references need be maintained.

The Hyper-G system (Kappe, 1995) adopts this approach, maintaining a large database on

the references used between resources. Should a resource move, the database is informed, and all references are updated. This is similar to the name service approach, in that an external service is used, but it is the relationships between resources which are maintained by the service, rather than the resources' locations.

Suitability for the Web

This approach either requires each resource to know which other resource has references to it, or requires an external service to maintain the references. However, the former approach is unrealistic, as the Web is a federated system, with no central control: a resource has no way of knowing who or what is referencing it. Equally, the latter approach is unrealistic, as the size of the database of references would become impossible to manage, and many Web servers are frequently offline, resulting in the database being swamped as it must store pending reference updates until they are online again (Briscoe, 1997).

Search

The search method does not attempt to update the references between resources, or to maintain the location of a resource. Rather, it uses a sophisticated search mechanism to find the resource if it migrates. To ensure success, the entire system must potentially be searched, which involves flooding the network. This has the advantage that so long as the server hosting a particular resource is accessible, the resource can be guaranteed to be found, as the flood will eventually cover all servers. Thus, the search approach has perfect robustness. The Harvest information system (Mic Bowman *et al.*, 1995) uses this approach to catalogue and index a distributed system's collection of resources. However, the Harvest system is used to index and search for pertinent information within resources, and so is effectively a search engine which can index an entire distributed system.

Suitability for the Web

While flooding a network provides perfect robustness, it is also the most costly method in terms of messaging overheads (Ingham *et al.*, 1996). A flooding algorithm must be implemented which spans the entire Web. To prevent the network being overwhelmed with packets (which, unchecked, would increase

exponentially), attempts must be made to restrict the flood. This can be achieved by including time to live fields in any messages sent by such a mechanism, but this requires knowledge of the exact diameter of the Web (Tanenbaum, 1996).

Selecting a migration method

The callback approach

The callback service can be immediately ruled out. As has been said, a resource on the Web has no way of knowing who or what is referencing it, and so any implemented callback service simply cannot be used.

The chain approach

The forward reference and the name service approaches can be grouped together and termed the "chain approach", as both rely on a chain references to effect migration. The difference is simply that the forward reference approach leaves its references on the servers it has visited, while the name service approach relies on a separate service to store and maintain its chain of references. The concept of the chain approach, then, can be examined in its own right, but does not meet all of the requirements specified above. The very fact that a chain exists exposes the whole approach to the chain's weakest link; in this case, the weakest link is the most unreliable server within the chain, meaning that a resource may be lost because somebody else's server has crashed. Finding that server can be difficult; worse, the resource's owner will have no control over the maintenance of the crashed server, and if it goes down permanently, the resource may be lost permanently. This is not just impractical, it is unacceptable to a network such as the Web which is forming the platform for e-commerce: losing a resource can sever the relationship between an organisation and its customers.

The search approach

The search approach comes closest to meeting all of the requirements specified above. Because the search would be performed within the network, the client need not be aware that a search is being performed; it simply receives the resource once it is located. Thus, universal client access is achieved. The search process would be performed using the resource's URL; consequently, as long as the resource has a

URL, it can be located, regardless of its content type. This achieves the requirement of content neutrality. HTTP and the URL can remain. In fact, so long as the identifier is unique, it can be of any format, leaving the way open for future formats of identifier to be used with the same migration mechanism. Full integration with the Web's current architecture is, therefore, achieved. However, the message overhead used to locate a resource cannot be ignored. Because it uses a flooding algorithm, the messages will grow exponentially with the size of the network. This is, at best, impractical when considering a network the size of the Internet. Thus, the search approach fails the practical design requirement. If this can be resolved, however, the concept of the search approach is far more robust and scalable than the chain approach. With no chains of references to maintain, and the ability to visit all hosts in a network, there is no weak link in the system. Resources, by definition, cannot be lost. Therefore, adopting a different search algorithm for the search approach could result in a practical search-based migration mechanism on the Web.

Adapting the search approach

The problems described thus far relate to a search algorithm which is parallel in nature, generating exponential traffic as the search progresses, and works on unstructured data. Such an approach cannot be practical on the Web, because its latency overhead occurs at the wrong stage of the migration process. The process of migration can be divided into two stages: first, a resource migrates; then, it must be located whenever a client wishes to use the resource. Generally, the migration stage can cope with higher latency times than the location stage. This is because there is no user interaction with the resource during the migration stage, whereas a resource usually needs to be located because a user wishes to download it. Currently, there is no migration mechanism on the Web; locating a resource is simply a matter of connecting with the appropriate server. Any mechanism that is required to locate a resource will incur its own overhead, and this adds to the latency involved in actually accessing the resource. To the user, this latency is perceived as a slower response time of the Web. With the chain approach, the main overhead occurs

during the migration process. Location is simply a matter of following a chain of references, and so long as this chain is not too large, latency should not be appreciably increased. However, with the parallel search approach described above, all of the overhead occurs during the location process, with the latency increasing as the search continues. Worse, the message overhead also increases (exponentially) as the search continues, resulting in a network with more location traffic than resource traffic.

This, however, is simply one end of a spectrum of search algorithms. For example, another approach could involve constructing a look-up table, with the set of all URLs on the Web being mapped to each respective resource's actual location. The URLs can be ordered as appropriate, and a trivial search algorithm used to locate a specific URL within the look-up table. While this centralised approach is not fault tolerant, and could result in all resources being lost, it does illustrate how structuring the data can fundamentally change the performance of the search approach. What is required, therefore, is an approach which structures the data, but across a distributed system of migration-specific machines.

An overview of a Web-based migration mechanism

This is the approach that is currently being investigated by the authors. A migration mechanism is being developed which uses an external (distributed) service to keep track of the URLs and the actual location of the respective resources. This is similar to the resolver discovery service adopted by the URN approach, and provides the indirection required to retain the format of the URL while allowing the resource to reside on a machine with a different name. However, while the resolver discovery service uses a chain of references to keep track of the migrating resources, the new approach uses what is, essentially, a migration-specific distributed "database". This database is constructed and queried using Web-based technologies, such as Extensible Markup Language (XML). Rather than searching all of the resources on all of the servers across the Web, the set of all resources are represented within this distributed database by their URLs, and it is this database which is searched to locate a resource. Fault

tolerance techniques will be used to ensure no resources are lost, and load balancing will minimise the latency incurred. Because the database contains URLs, any content which can be addressed using a URL can safely migrate using this system. All that is required is for the system to be notified when a migration has occurred. This can be done by the server the resource has migrated from, or the server the resource has migrated to (or, for that matter, by the resource itself, if it contains intelligent content).

Development of this system is currently a work-in-progress, and results from the completed system will be published in a later paper. The next section discusses some of the new services such a system can provide to the Web.

Providing new Web-based services

How a migration mechanism can enable new services

Within a distributed system, much use is made of the term "transparency" (RM-ODP, 1995). This is used to convey the concept that the services performed by the distributed system (such as migration) happen without components being aware that anything has changed. Thus, a transparent migration mechanism is one in which components are migrated to another machine without the component, or a client wishing to access the component, being aware of the move. However, such a mechanism can be made "translucent"; that is, the components can be moved transparently, but if they require the service themselves, they can use it to initiate their own migration. In this way, the migration is controlled by the component rather than the server hosting the software resource. For example, static content has no intelligence, and so cannot make use of a migration mechanism. Therefore, if the resource encoding the static content is to be migrated, it must be at the server's discretion. The server is therefore able to migrate the resource without the resource or any other host being aware of the move. Intelligent content, however, has the ability to use any service the network can provide. Thus, a migration mechanism can be used by intelligent content to migrate itself. It may choose to do this for the purpose of network optimisation (for example, if it detects that the server's performance has degraded due to

excess demand), or it may do this to achieve a goal on behalf of a user. This would effectively enable the intelligent content to become a mobile autonomous agent (Franklin and Graesser, 1996); that is, software which can roam across a network, performing tasks on behalf of a user.

In this way, a translucent migration mechanism on the Web can provide a host of new and extended services. The same mechanism can be used by intelligent content (to autonomously roam the Web), and by Web servers (to optimise the network); it can solve the "broken link" problem typical of hypertext documents, whereby a URL embedded within an HTML document is rendered useless when the content it refers to is moved. It can also be employed on a company's intranet, allowing resources to migrate freely, either of their own volition, or transparently by the server hosting them. By providing its servers with the ability to monitor their own performance, a company can simply connect a new server to its intranet, and wait for resources to migrate to it from existing servers under strain. Using dynamic network configuration protocols, and wireless network technology such as wireless LAN, this facility can be extended so that a server need only be brought into range of a mobile basestation, and switched on: the server will connect to the network, and the resources will populate the server, automatically.

Mobile servers

Basing the migration mechanism on a search approach effectively provides a service which resolves the IP address of a machine given a specific resource. Thus, the same host can have many different IP addresses over time (for example, if the host is roaming) yet its resources will still be locatable (providing the host is accessible to the migration mechanism), because the mechanism ensures the resource can be located regardless of the current IP address associated with it. This implies that mobile servers can be developed with IP addresses which change according to the server's location, without affecting the accessibility of the resources being hosted.

Services for distributed component systems

Distributed component systems can use the mechanism to migrate components. Any type of content can use such a migration mechanism,

and this includes intelligent content such as distributed components. Thus the mechanism enables the Web to provide a generic migration service to such component systems. In this way, the Web can become a distributed platform, enabling distributed systems to build their own specific services on top of the Web's generic services. For example, system-specific messages between components can be routed to individual resources (components) irrespective of where the resources are located, using the generic services provided by the migration mechanism.

Optimising the network through network traffic profiling

Deciding which content to migrate and when can optimise both the performance of a server, and a network as a whole. Currently, certain network technologies and service level agreements (SLAs) with network providers insist on the network user specifying the expected quality of service of the network at certain times of the day. For example, Frame Relay can ensure a certain throughput to the user over a short period of time by guaranteeing a committed information rate (CIR). This CIR is the rate which is, on average, available to the user.

Determining the CIR is a difficult process and involves a good knowledge of the local traffic. The network manager has to plan for the expected traffic, keeping in mind that at very busy times he does not have the same throughput and availability of capacity above the CIR for "bursty" traffic. Traffic profiling is very important in such networks, whereby the traffic is monitored in order to determine the quality of service required. Research by the members of the author team is developing a methodology for profiling traffic in this way. It has been determined that while overall network traffic may be variable over the short term, over time it only increases. The SLAs, therefore, can provide the business case for introducing content migration as a means of balancing the network and staying within the CIR. With a transparent migration mechanism built into a company's intranet, software resources can be migrated to balance the load not just across servers, but across the network. A traffic profiling system can be used to monitor the traffic on a company's network. If network traffic has increased at a particular

point, resources can be migrated to ease the flow of traffic at the bottleneck. If the traffic is too great only at certain times of day, the profile will show this, and the resources can be migrated back and forth according to the time of day.

Conclusion

This paper has examined the various issues involved in developing a practical migration mechanism for the Web. It has identified the requirements of such a mechanism, and examined some of the different approaches that can be used to implement a migration mechanism with respect to these. However, no current migration system meets these requirements, largely because they are not content-neutral. Therefore, the authors are currently working on a migration mechanism that will meet these requirements, and thus could form part of the Web's infrastructure. A transparent, search-based, resource-level migration mechanism for the Web, combined with existing distributed component architectures and sophisticated network traffic profiling techniques should optimise both a server and the network, and can provide a new class of services to users. While the Web is currently a distribution system, the integration of a migration mechanism can provide the Web with the services it needs to offer to become a ubiquitous distributed system.

References

- Berners-Lee, T. and Connolly, D. (1995), "HyperText Markup Language – 2.0", RFC 1866, MIT/W3C, November.
- Berners-Lee, T., Fielding, R. and Frystyk, H. (1996), "Hypertext Transfer Protocol – HTTP/1.0", RFC 1945, MIT/UCS/UC Irvine, May.
- Berners-Lee, T., Masinter, L. and McCahill, M. (1994), "Uniform resource locators (URL)", RFC-1738, CERN/Xerox/University of Minnesota, December.
- Briscoe, R.J. (1997), "Distributed objects on the Web", *BT Technology Journal*, Vol.15 No.2, April, pp.158.
- Ingham, D., Caughey, S. and Little, M. (1996), "Fixing the 'broken link problem': the W3Objects approach", in *Fifth International World Wide Web Conference*, 6-10 May, Paris.
- Franklin, S. and Graesser, A. (1996), "Is it an agent, or just a program?: A taxonomy for autonomous agents", in *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, Springer Verlag, New York, NY, Berlin, 1996, <http://www.msci.memphis.edu/%7Efranklin/Agent-Prog.html#agent>
- Kappe, F. (1995), "A scalable architecture for maintaining referential integrity in distributed information systems", *JUCS*, Vol. 1 No. 2, February, pp. 84-104.
- Mic Bowman, C., Danzig, P.B., Hardy, D.R., Manber, U. and Schwartz, M.F. (1995), "The harvest information discovery and access system", *Computer Networks and ISDN Systems*, Vol. 28, pp. 119-25.
- OMG (1995), "The Common Object Request Broker: Architecture and Specification, Revision 2.0", Object Management Group.
- Orfali, R., Harkey, D. and Edwards, J. (1996), *The Essential Client/Server Survival Guide*, John Wiley & Sons, New York, NY, and Chichester.
- RM-ODP (1995), "Open distributed processing reference model (RM-ODP)", ISO/IEC DIS 10746-1 to 10746-4, 1995. <http://www.iso.ch:8000/RM-ODP/>
- Sollins, K. and Masinter, L. (1994), "Functional requirements for uniform resource names", RFC 1737.
- Tanenbaum, A.S. (1996), *Computer Networks*, 3rd ed., Prentice Hall, Englewood Cliffs, NJ.

Traffic Characteristics of WWW Sessions

*Georg Mueller *, Peter Sanders* and Garry Joyce ***

** SECEE, University of Plymouth, UK*

*** AT&T Unisource, UK*

The growth of the World Wide Web (WWW) and its use is causing a dramatic impact on networks. The use of the older protocols and applications are being replaced for WWW usage, by quasi-standard interfaces like Internet Explorer from Microsoft and Netscape's Navigator. The following examines whether existing models of traffic theory can be used for WWW session arrivals. The need to understand the underlying statistical properties of the distribution of WWW sessions, relative to time over a corporate network, influences the design and modeling of Wide Area Network (WANs). Such information can be important since the behaviour of the WAN and the corporate network will depend on the performance and Quality of Service (QOS) requirements of the network.

There are two distinct but equally important components to any session / time distribution:

- The session duration distribution, which is the distribution of the difference between the initiation and the end of a transaction of a session
- The inter-arrival time distribution, which is the distribution of the difference between the connect time between two consecutive sessions

In this paper, the practical circumstances and actual results of WWW transactions and sessions will be discussed. The following parameters will be analyzed: session duration and session inter-arrival times for groups of users and for a whole corporate network.

Measuring Conditions

In this measuring environment every WWW activity on a corporate LAN is recorded. For each transaction a record can be set up containing the start and end times of a WWW session, with the originating address and destination address. These records can then be analyzed to provide a more detailed profile of session distribution for individual addresses, for groups of addresses and for a complete corporate LAN. It is interesting to see the difference in the distributions between groups of addresses, which usually represent a particular department, and the whole corporate network.

The theory indicates that a sequence of events recorded with regard to the time at which they occurred are taken as independent and will give rise to a negative exponential probability distribution, with a probability density function given by:

$$f(t) = \theta e^{-\theta t}$$

where t represents the time between any two consecutive events. In the study of traffic theory and telephone networks two events are often examined. They can be either the start and finish times of a call or the start times of two successive calls. Either the duration times or the inter-arrival times of these sessions can then be represented as having negative exponential distributions.

If the inter-arrival times follow the negative exponential distribution described above and C represents the number of sessions arriving in a given unit of time, then C will follow the Poisson distribution given by:

$$C = \frac{(\lambda \cdot s)^x e^{-\lambda s}}{x!}$$

For this reason, phenomena such as sessions within a communications network found to have negative exponential distributions of inter-arrival time and duration are commonly referred to as having Poisson characteristics or obey the Poisson model. The widely presumed independence between sessions has meant that considerable sections of the theoretical work carried out to date, with regard to data networks, have been based on the Poisson assumption.

Empirical Results

The negative exponential and related Poisson distributions provide typical examples of theoretical distributions. Even in instances where the real network may be modeled in this way, there will always be areas in which observed data will deviate from the perfect form. Variations like this are inevitable in a situation such as the AT&T network in which session patterns seems to change from one moment to the next, from day to day and from week to

week. A slight lack of precision must be taken into account when testing for distributional fit – perfect exponential curves and straight-line logarithms are unlikely to be found.

For any recorded network session, the number of sessions will of course be finite while the theory states an infinite number. It is, therefore, unlikely for the inter-arrival times and duration to have occurred with exactly the frequency predicted by the Poisson model. However, the amount of data traced should be enough to give an impression of the distributions. In a perfect negative exponential distribution, sessions of any positive length, no matter how long, occur with non-zero frequency and no two inter-arrival times can occur with the same frequency. Other network factors peculiar to the session of interest will demonstrate the inaccuracy of the real world examples. There may be, for example, at a particular time, good reason why sessions within a certain range of lengths occur more often than others. This is especially true for short measurements or extremely long sessions. Often sessions with just a few bytes are sent, which represent polling of equipment or time-dependent updates of WWW pages. Automated WWW session which access the same sides and poll for new information show this occurrence, as one accessed WWW page only changes its traffic profile when strong congestion occurs.

There are a number of interesting points, which can be extracted from the data:

- The data is not a totally random nature
- It is likely that the delay distribution data follows a Poisson distribution form:

It is interesting to note that some, but not all, users seem to have a distinctive “signature” when establishing a session. Usually all users connect to an initial WWW page, which is set up as the “home” page on the Internet browser. From there on users tend to go to either a location which they usually use, or a search engine, which acts as an Internet directory.

Typical results are given in the Figs. 1 and 2, which show the inter-arrival time frequencies, in both standard and logarithmic form. The approximation for the negative exponential frequency distributions were noticeably better for larger samples than for small samples,

presumably since individual peculiarities are more likely to affect experiments with fewer measurements and group behavior mentioned earlier. In general, the fit of the distribution was good in the majority of cases for which the sample was at least large enough.

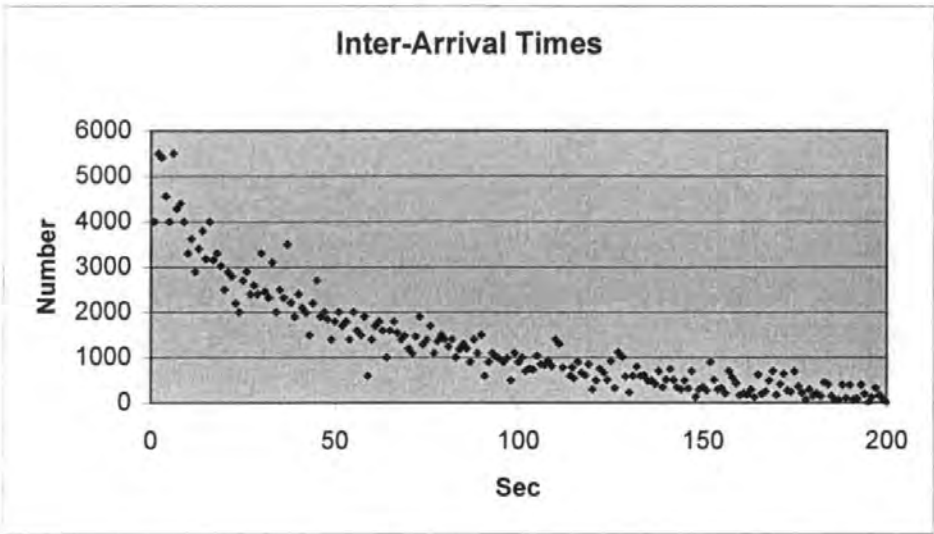


Figure 1

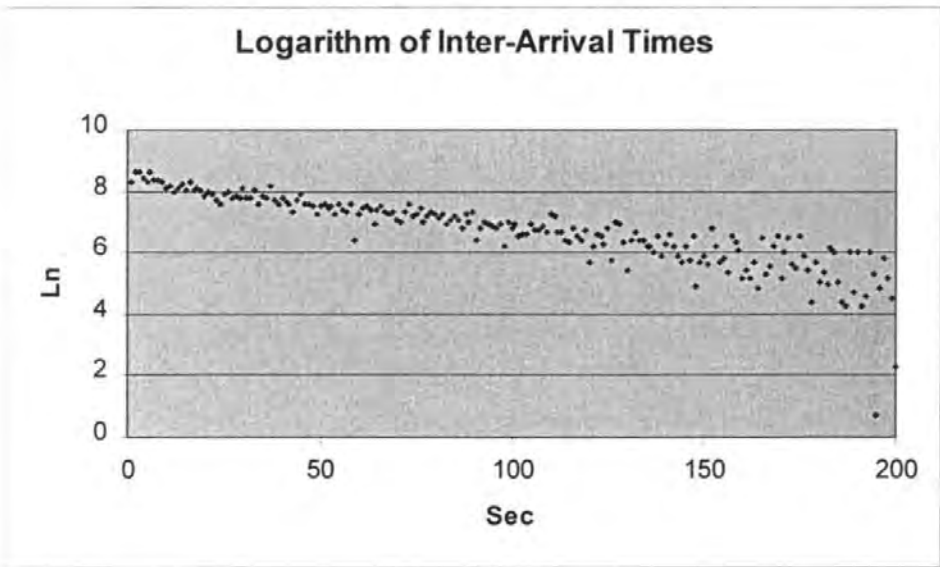


Figure 2

Conclusions

The conclusions that can be drawn from the study are as follows:

- There is indeed a pattern to the sessions made by specific WWW groups on a LAN. The pattern consists of user groups that tend to access the same WWW pages over and over again, e.g. stock quotes or news. This would imply that user sessions are not totally independent in their choice of WWW sites and this specific user behavior creates unique probability distributions. Usually these distributions are not exponential and can vary greatly. Maybe the reason for this is that user groups often have similar interests found in their job or hobbies.
- If the whole corporate network is analyzed, the specific distributions of the individual groups tend to merge and produce a negative exponential distribution, because many addresses and users tend to use many different locations.
- It is possible to quantify the distributions and it would appear that the distributions remained relatively constant over the period of the study.
- There is a strong suggestion that the delay distribution and possibly all the parameters follow simple distributions. The most likely candidate for the inter-session delay for a whole corporation is the Poisson distribution.

Acknowledgments

This paper results in part from the long collaboration of the University of Plymouth and AT&T UK. The work was carried out at AT&T Capacity Planning in Redditch. We especially thank Stephen Nicholas and Steve Willis.

A New Methodology on Overbooking in Frame Relay Networks

Georg Mueller, Peter Sanders*, John Allen***

** SECEE, University of Plymouth, UK*

*** Netscient Limited, UK*

1. Introduction

The explosive growth of computer networks and the massive interconnection of heterogeneous network infrastructures in recent years have created the need for new analysis methods and modelling tools. The expenditure on Information Technology goes up every year, and securing these investments is crucial to companies. The estimation of future network traffic is as important as the influence and impact of different design rules and their impact on performance.

The number of applications which require a large amount of network capacity for a short period of time steadily increases. These are served by Frame Relay systems and because Frame Relay is a statistical multiplexing based protocol, it allows the access circuits and core trunks of the AT&T network to be better utilised for bursty communications. In theory it allows the allocation of an entire access link to a single Permanent Virtual Circuits (PVC) for the duration of a burst and is, therefore, very suitable for Local Area Network (LAN) interconnections. To ensure a certain throughput to the user over a short period of time, the network guarantees a Committed Information Rate (CIR). This CIR is specified in Kbps and is the rate which is, on average, available to the user. If a user transmits data at a rate exceeding CIR a control bit is set to indicate this frame can be selectively discarded when there is congestion on the network.

Frame Relay policy considerations and network analysis must begin to interact in a way not previously recognised or implemented. One way of using the full potentials of this technology is by over-subscription of capacity. To be compliant with the commonly used terms in frame relay, we will call them Overbooking Factors (OBF). By overbooking the available capacity the provider takes advantage of the fact that not all PVCs use their full CIR at the same time. PVCs which are silent are providing busy PVCs with their capacity and consequently a sharing of trunk resources.

The aim of this project was to develop a methodology, which allows a more accurate design of frame relay networks by using traffic profiles of every individual PVC and utilise trunk resources in the best possible way. The hypothesis is, that individual OBF can be set to gain both advantage in the service provided to the user as well as economy to the network provider. The basic idea is to change the nominal booking factors from a static value only dependent on CIR to individual overbooking of PVCs dependent on recent actual usage. PVCs with a high overbooking will be mixed with PVCs which have a low overbooking. This wasn't done previously, as all PVCs with the same CIR were treated equally.

In the face of the current evolution of global information infrastructure, vastly expanding both in complexity and sophistication of applications, measurements and experience offer evidence to support our hypothesis.

The results in this paper have been obtained over a period of 6 months from the AT&T Unisource Frame Relay network in Europe. The network is based on Stratacoms IPX/IGX Frame Relay technology. In data networks it is important to plan for the busy period, which is defined as that period of the day which is utilising the network most and has the greatest risk of losing data.

The introduction of new overbooking rules will improve customer service, increase reliability of the network and thereby reduce costs. Changing the overbooking factors is a crucial component and must, therefore, be implemented in the existing planning process. To demonstrate our theory, we will utilise these results by showing how

- (1) to balance the Frame Relay Network, and
- (2) utilise trunks to a specific threshold by using traffic profiles.

Our methodology is based on a "rule of thumb" technique as bursts and utilisation levels vary according to user levels, time of day, seasonality and introduction of new applications and service levels expectations to the users. The method differs from the existing AT&T design rules [1], which treats all PVCs the same way, independent of their real utilisation.

The strength of this method is that the revised rule applications can be automated and there are nearly no maintenance costs for capacity planning and redesigning for new routings. However, several other factors, described in section 2, motivate to an alternative approach.

Most notably, not all PVCs can be treated in the same way, as they use different protocols and applications and, therefore, show a different traffic profile. We do not assume that the dedicated capacity rate stays within the limits, but rather ground the definition on real utilisation and peak levels of the individual PVCs. A principal objective is finding the balance and granularity of reconfiguring intervals. This approach can address some fundamental Stratacom Frame Relay opportunities, including performance requirements of IPX/IGX switches and quality of service levels, e.g. packet losses.

Initial work in traffic profiling has been done before by Kleinrock [5], where the growth of the ARPAnet in 1971 was discussed. Kleinrock found that traffic grows exponentially with time, but slows down after a certain period of time. More recent work has been done by Klaffy et al [6] who were investigating the backbone traffic of the NFSnet and Caseres et al [4] investigating characteristics of Wide Area TCP/IP conversations.

Most of the work however is focused on the Internet growth and the TCP/IP protocol suite and not on the carrying technology. We will focus on the Frame Relay technology, which is widely used for the interconnection of LANs of large corporations and their offices and services.

2. Common Practice

In this section we describe the existing design rules and the technical parameters in relation to Stratacom's frame relay equipment. The Stratacom switches have a parameter, which allows the setting of overbooking factors. This parameter is called the Percent Utilisation Factor (%UTIL) and indicates what percentage of the CIR will actually be reserved. This factor is the inverse of the overbooking factor.

The switch determines

$$S_T \geq \sum_i CIR_i * \%UTIL_i * B_{iT} \quad \text{for all trunks } T$$

where

S_T is the trunk speed

$$B_{iT} = \begin{cases} 1 & \text{if PVC } i \text{ is carried on Trunk } T \\ 0 & \text{otherwise} \end{cases}$$

At the start of Frame Relay it was understood that some network providers had Overbooking Factors (OBF) of 10:1, where it was assumed that users only used 10% of their actual CIR. With time, however, this OBF gradually decreased, as users started to use more of their available resources and Internet and its applications like the World Wide Web (WWW), and other TCP/IP connections started to increase [3].

One extreme would be an overbooking of 1:1, and in some cases it is certainly required, dependent on the Quality of Service (QOS). As a general case it is, however, uneconomical, as the price for this service is much greater and Time Division Multiplexing technologies become more appropriate. Experience shows that an overall overbooking of around 3:1 is a practical compromise for network providers that satisfies most QOS requirements of the users. The question of the overbooking threshold is an economic question, and is driven by the profit margin from the trunk cost, maintenance cost etc. and the asking price from the customer.

To determine the CIR is a difficult process and involves a good knowledge of the local traffic. The network manager has to plan for the expected traffic, keeping in mind that at very busy times he doesn't have the same throughput and availability of capacity above the CIR for bursty traffic. Depending on the use of protocols, this can have major influence on the choice of the CIR. Time sensitive protocols like SNA need more consideration than TCP/IP applications.

Very often the choice of CIR is not influenced by the traffic volume itself, but by the budget a company is able to afford. In cases where the CIR is chosen too low and the general OBF is applied, we have two inaccurate parameters. In such cases the measured dynamic UTIL is sometimes much larger than the 100 %. This not only causes congestion, data loss, time-outs of protocols and poor throughput but also is a financial loss to the network provider.

However, the goal of finding the “optimum network design” is most likely not achievable, as the traffic sources have a level of uncertainty in their profile. Reynold’s approach [2] settles therefore for a “near optimum design”, in which he states that “slight deviations from the optimum conditions are of little importance” and not of great impact.

Examples for actual existing design rules could be as the following:

The %UTIL for PVCs are set at

- 10 % for 16kbit/sec
- 33% for all other CIR.

This means that all PVCs with a CIR of 16kbit/Sec will be overbooked by 10:1 and all other PVCs by 3:1, regardless of their traffic profile.

This categorising should enable the network provider to maintain a required quality of service. It is assumed that the lower loaded PVCs will allow the higher loaded PVCs to use the available capacity on a trunk, as not all users are using their PVCs at the same time. This assumption is supported by looking at the overall dynamic utilisation of a network. The existing simple overbooking rule results in some trunks being overloaded whilst others are relatively lightly loaded at peak hours.

Hence practice shows that certain trunks tend to have more frame losses than others. As the network uses fixed routing and all parameters are set manually, it is, therefore, only able to react to these overloading problems by congestion control.

The Stratacom nodes allow two possible modes of routing:

- preferred routing
- non-preferred routing

Preferred routing allows the network provider to manually set up a route for a PVC. All frames on this PVC will then follow this route. Only in case of trunk failure, is an alternative route taken. It will return to the preferred path when the outage is cleared.

Non preferred routing is an algorithm in the Stratacom nodes which sets a path through the frame relay network. This path will exist till a trunk failure occurs. The algorithm is very simple and works by calculating the minimum number of hops from a start node to a destination node and then chooses the path with the largest unreserved capacity. On failure of a trunk the algorithm is re-run to choose a route from the trunks available, but the problem is that the path is not altered when the failure is cleared. This is a big limitation of the algorithm as large delay paths may arise and continue to persist until manually cleared by operator intervention..

AT&T in Europe decided to use preferred routing in their network to have direct control over the routing and thus avoid using nodes in America as intermediate hops for “European” traffic. Another reason is that price differences for trunks, which vary a lot in

Europe, are not taken into account by *non-preferred routing*. By usage of preferred routing the predictability of the operation of the network was possible.

3. Profile

One limitation in profiling is the availability of detailed statistics over a greater period of time. Usually this has to do with memory restrictions and performance problems, as gathering statistics is a processor intensive procedure. The available useable statistics in our case are limited to the number of bytes transmitted per hour. We think, however, that this figure is giving us a reasonable granularity for our methodology and planning purpose.

Figure 1 shows the overall utilisation of the Frame Relay network in Europe. The utilisations have been taken over five working days in a week from 6 a.m. to 8 p.m. It can be seen that these periods are showing a relative constant traffic profile. Looking at the network growth in terms of utilisation over a longer period of time, where new PVCs were excluded, the overall network utilisation grew by 2% per month.

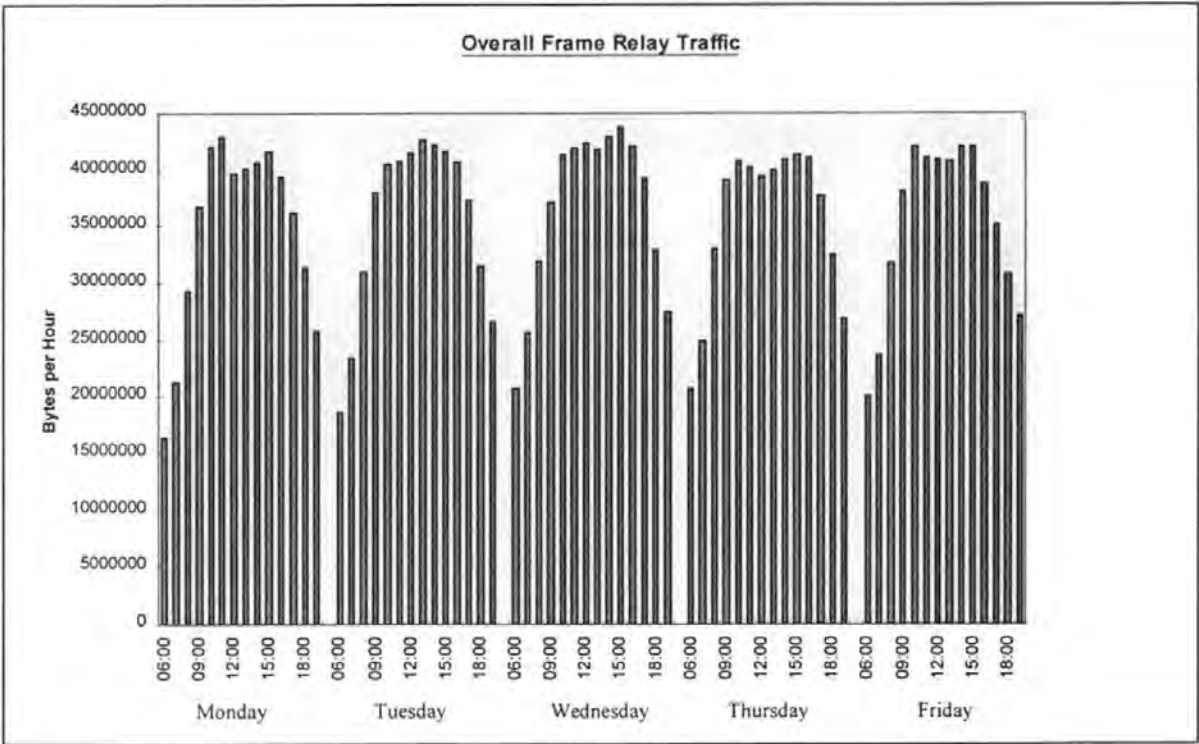


Figure 1

Figures 2 and 3 show examples of two different PVC dynamic UTIL. These PVCs show hourly figures (from 6 a.m. to 7 p.m.) for 5 consecutive days. It can be seen that the "busy" hours are usually between 9 a.m. to 3 p.m.. There are a few peaks which are much greater than the busy period, but these peaks are not crucial as they change at every PVC and would comply with the theory of statistical multiplexing. We are not especially concerned regarding peaks occurring from time to time, as we are only interested in the trended OBF.

Even in the worst case when peak traffic from many PVCs occur at the same time and cause congestion, our planning strategy would still outperform the general OBF procedure.

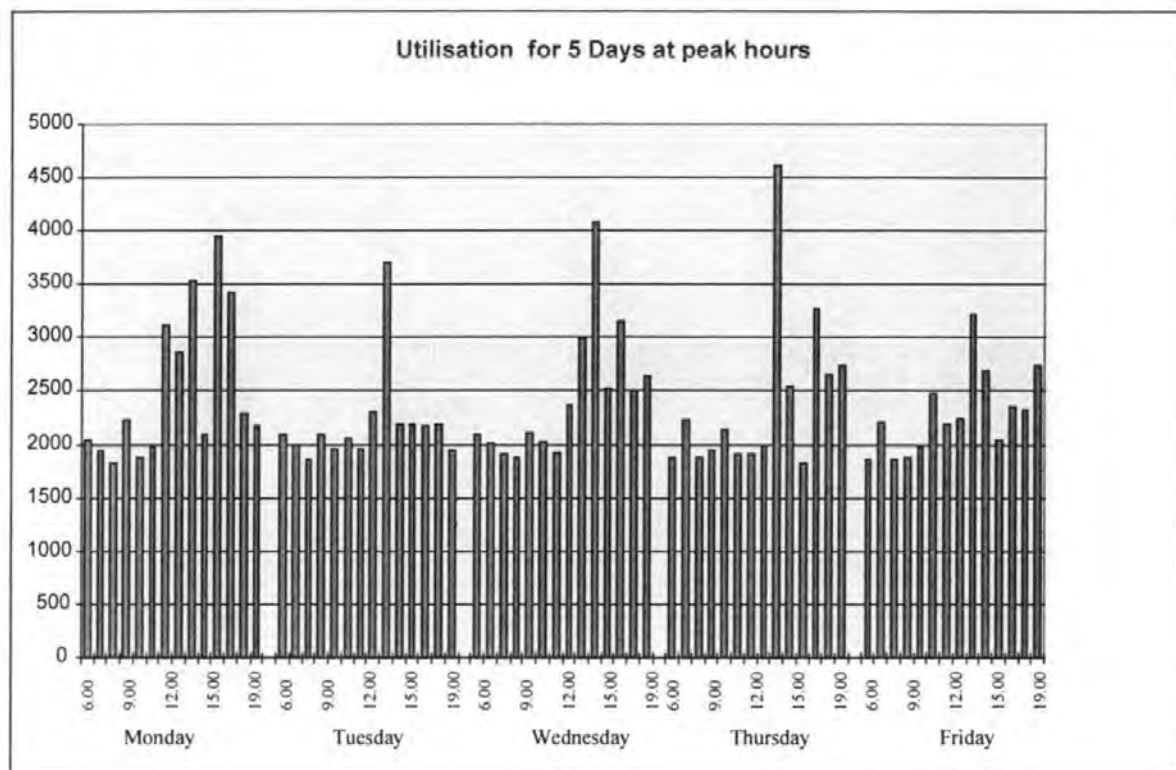


Figure 2

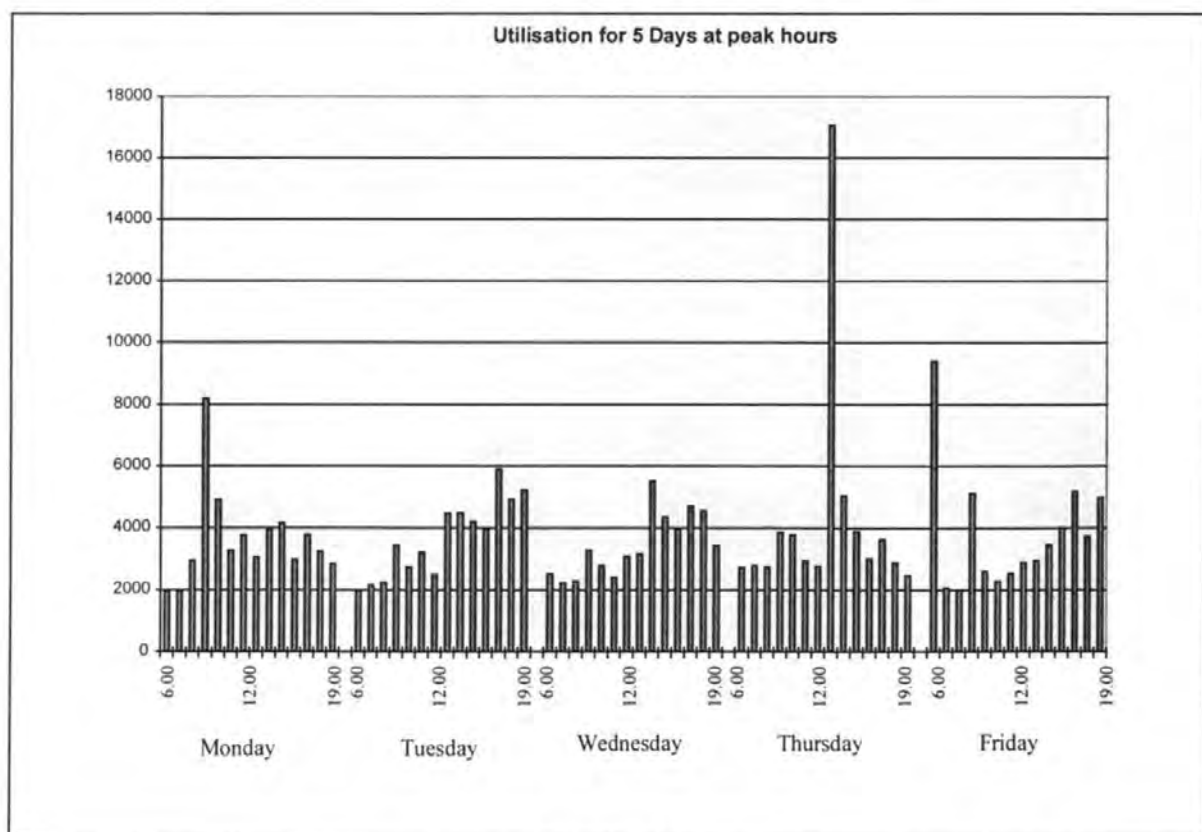


Figure 3

4. Distribution

Our research has shown that the static utilisations are generally set higher than busy hour actual utilisation for a great number of PVCs, while a few PVCs are using more capacity than allocated. Using existing utilisation values for each PVC to design the network we would get forecast trunk utilisations in line with recent actuals. The network would then be more balanced, enabling the trunk to be loaded by low utilised and high utilised PVCs. At the moment, certain trunks are running at nearly 100% dynamic utilisation causing packet losses and re-transmitting of frames and messages. The model and the static load on a trunk appears to be correct, however, the use of the general overbooking rules does not take into account the imprecise setting of OBF.

It can be seen in Figure 2 how the dynamic UTIL is distributed. To show it graphically we selected the dynamic utilisation of all PVCs with a CIR of 256 Kbit/sec and sorted them in descending order. The distribution of PVCs on this graph are compliant with the 80/20 Pareto rule. This not only applies to 256 Kbit/sec PVCs but also for every other PVC size. This means that by individual overbooking AT&T could save capacity and money, whilst giving better service. The capacity allocation for the whole network based on actual dynamic utilisation is less than that required by the existing rule. Naturally, preferred routes have to be adjusted to reflect the required individual PVC capacity allocation.

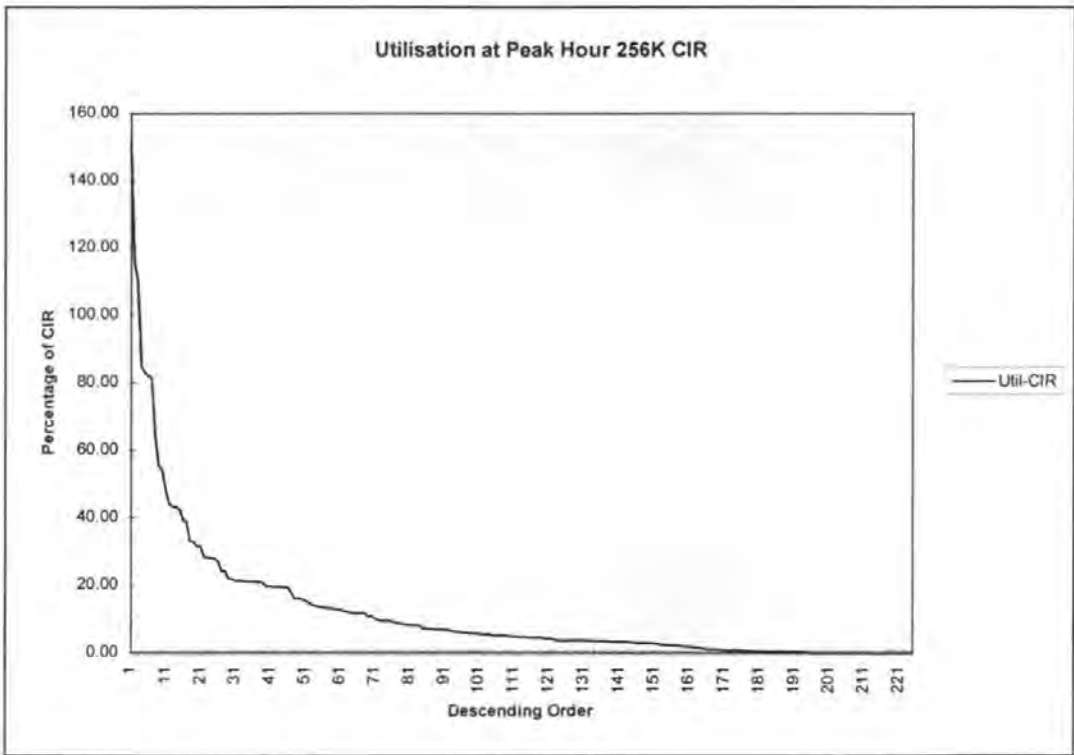


Figure 4

5. Variability

We assume that by using real dynamic utilisation, the model will be very accurate; however, if the dynamic utilisation readings put into the model are changing too often and are too high or too low, the model will not represent the network. It is therefore important to have a certain continuity of dynamic utilisation levels over the measured weeks. As we are not trying to fill the trunks completely, we can measure the utilisation of each trunk in the network at peak hours to see how close to the predicted utilisation they are and adjust the dynamic utilisation of each PVC that we feed into the model accordingly.

PVCs which change their utilisation at peak hours dramatically in upwards and downwards directions from one month to next are very difficult to categorise, but by monitoring these changes, actions can be taken very soon after this occurs. In Figure 5 we show the change in dynamic utilisation over the period of a month. The results in the graph are for the month of May 1997.

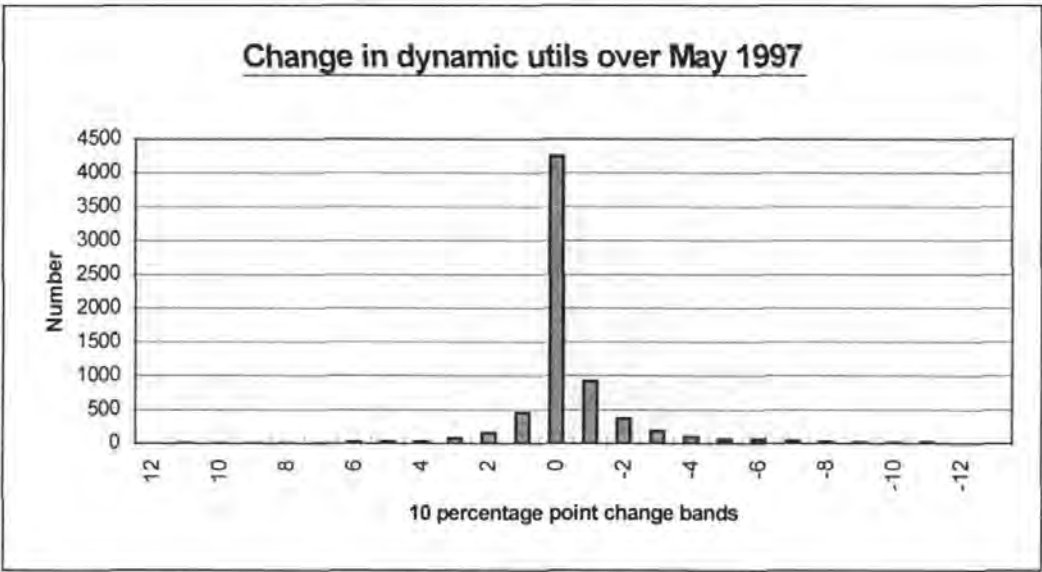


Figure 5

The graph shows how stable the average dynamic %UTIL is in practice. Nearly 90% of all PVCs change by less than 10% and, therefore, give a good level of confidence for our methodology, and hence predictability.

Our investigation has discovered that PVCs peak utilisation tend to grow with a steady linear rate, interrupted by a sudden jump. This "step" upwards is explained by the introduction of new software or the use of new servers and services by the users. In the case of new Internet connections, some PVCs jump by 100 % and more, over one week, and stay on this level. This has to be recognised as these PVCs need special treatment and should be watched more closely. Averaging the busy period in a rolling window and not recognising this jump could result in a big error and in capacity limitations. One major contributor to these traffic jumps are the Intranets and the use of WWW browser. It is difficult to predict these steps, but from experience on some of the AT&T networks, it can be seen that the appearance of WWW servers create these sudden increases of traffic. Providing the step changes occur to a small number of PVCs in a month, their extra capacity needs will be met from the contingency provided by the allocation to the rest of the PVCs on a trunk.

Experience indicates that mixing traffic from many users on a trunk reflects these individual step changes into a general overall expansion of use. Monitoring and revising utilisation weekly is expected to adequately accommodate these movements.

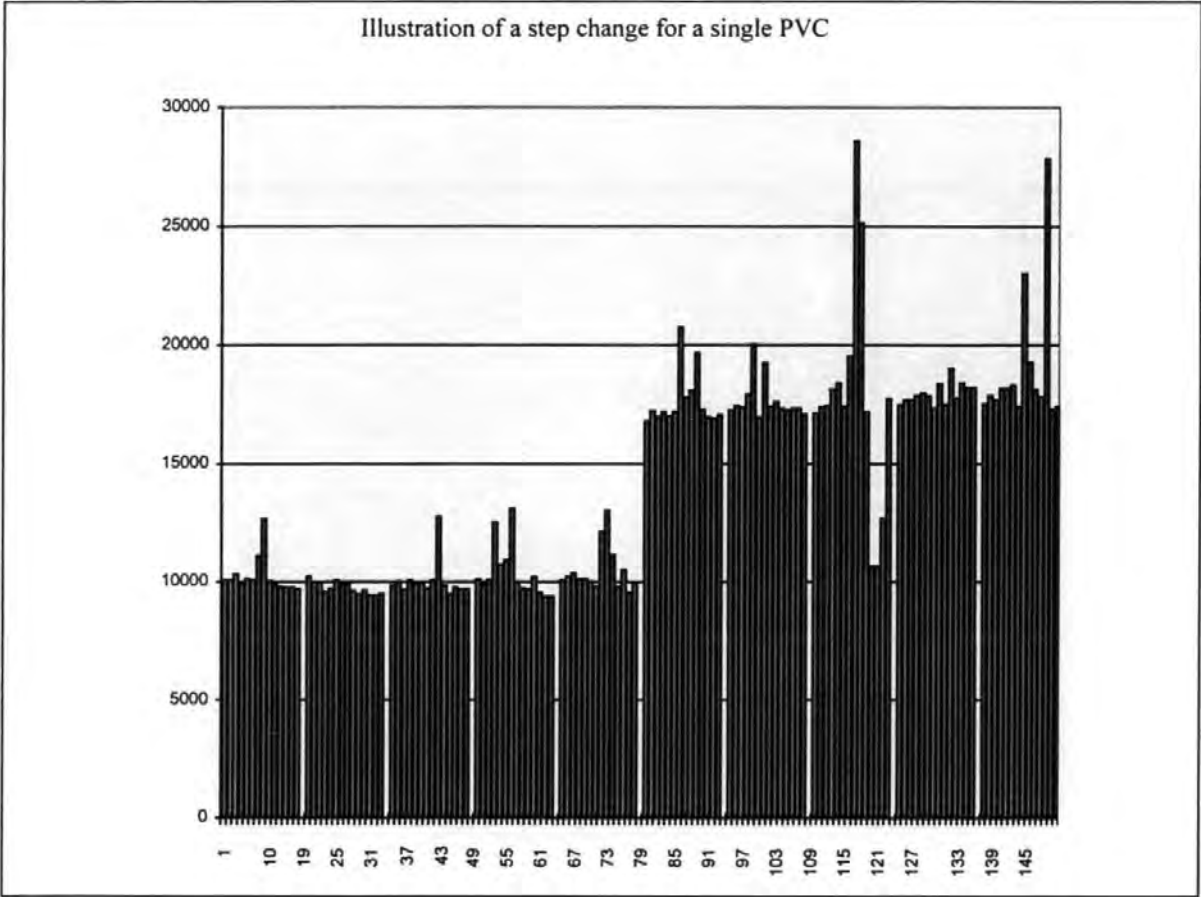


Figure 6

6. Methodology and Implementation

The following design rules for the %UTIL for PVCs are currently applied at AT&T:

CIR	%UTIL	CIR	%UTIL
4	64	64	24
8	32	96	32
12	21	128	32
16	32	192	32
24	21	256	32
32	24	512	32
48	26	768	32
56	22	1024	32

Instead of using the general OBF the dynamic utilisation of each PVC will be fed into the model. These are obtained using the introduced methodology:

1. Identifying the busiest day of the week for the whole network in terms of total traffic. It seems that the middle of the week, or usually between Tuesday and Thursday, the traffic levels are at their highest.
2. The average traffic of the peak hours in terms of total traffic has to be determined. As can be seen on the previous charts, the busy period are between 10:00 and 16:00 and are used for each PVC on Tuesdays, Wednesday or Thursdays.
3. These busy day figures are recorded for a rolling window of 13 weeks. The 13 weeks was chosen, as this gives us a period where the granularity is small enough for identifying changes and still large enough for the overall trending and it gives a good level of confidence for predictions.
4. These figures then give a trend of the dynamic utilisation for each PVC. For an optimisation of the routes in an existing network, the trended utilisation of the PVC to 2 weeks is fed into the model. Two weeks are chosen to enable the network planning for reaction time, e.g. capacity management.
5. For some of the PVCs the statistics show utilisation levels greater than 100%. This is because they are running over trunks that are not congested. For design and configuration purpose these PVCs are limited to 100 %UTIL as we want to design the network for a maximum of up to the CIR for each PVC. The reason for setting the PVC to a maximum of 100 %UTIL and not higher, is economic, as AT&T is committed in the contracts to deliver a service set by the CIR. By setting the utilisation at a level higher than 100% the company would be giving capacity away, and therefore losing money.
6. PVCs which are not used by the customer, or where the dynamic utilisation is low, are set to a minimum level.
7. PVCs which are permanently loaded at 100% dynamic utilisation or higher should be upgraded to a larger CIR, as the customer will have a better performance and the profit margin for AT&T will increase. Customer Service and Sales will negotiate revising CIR with the customer.

-
8. When adding new PVCs into the model, their expected %UTIL will have to be specified. These can be obtained by trending the average utilisation of each band of PVCs in a particular CIR range and averaging. The average is then used and the new %UTIL is applied when the real dynamic utilisation levels have been monitored and can be applied into the model. Alternatively the old rules can be used for the initial overbooking level.

7. Conclusion

We have proposed a methodology which is based on traffic profiles and individual PVC traffic behaviour. Implementing the methodology delivers better service and save potential long term costs. There is a lot of scope for more investigations into this area, as these ideas can be used and further developed for future technologies. It is important to create traffic libraries and associated statistical models for planning purposes as well as traffic management and billing.

8. References

- [1] AT&T-Unisource StrataCom Design Rules(Release 8.1.71)
- [2] Reynolds P. L., Expert System Approach To Private Telecommunications Network Design, BT Technology Journal, Vol.11 No.4 October 1993
- [3] Vern Paxson, Growth Trends In Wide- Area TCP Connections, IEEE Networks, 8(4) July/August 1994
- [4] Caseres R., Danzig P., Jamin S & Mitzel J. (1991) Characteristics Of Wide Area TCP/IP Conversations, Proceedings Of The 1991 ACM Sigcomm Conference
- [5] Kleinrock L. (1976) Queueing Systems, Volume II: Theory, John Wiley & Sons
- [6] K.Claffy, G. Polyzos, And H.W. Braun, "Traffic Characteristics Of The T1 NSFNET Backbone", Proceedings Of INFOCOM '93, San Francisco, March, 1993

Copyright Statement

The copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the author's prior consent.