INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI[®]

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Traffic Engineering of Narrowband Networks

by
Melanie Fortin, B.Sc.
Laurentian University

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario

September 2000
©2000, Melanie Fortin

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

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

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

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

0-612-57726-0

Canada

The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis

Traffic Engineering of Narrowband Networks

submitted by Melanie Fortin, B.Sc.
in partial fulfillment of the requirements for
the degree of Master of Engineering

Thesis Supervisor

Chair, Department of Systems and
Computer Engineering

Carleton University

September 2000

date

Abstract

In this era of broadband and high speed networks, there are still some (e.g. the Armed Forces) who require models for narrowband networks. Narrowband links can be a satellite link, a radio frequency link or even a terrestrial link. The rate of error appearing on these links is different from one link to another. For this reason, this thesis evaluates the effects that increasing the error rate has on the traffic of the network. In order to achieve this, a test network is set-up for the purpose of collecting data when different error rates are injected into the link. The first- and second-order statistics as well as the statistical distributions are analyzed for the traffic measurements. The results show that the statistics vary with the rate of errors, however the distribution type does not. The resulting distributions also indicate that the current traffic models, which are developed for broadband networks, might not be appropriate for narrowband networks.

Acknowledgments

I wish to thank my thesis supervisor, Professor Mike Devetsikiotis, for his guidance and encouragement. The time he devoted to my research as well as the financial support he provided are deeply appreciated. I am also thankful to Dr Gerard Nourry and his staff of the Communications Research Centre Canada for their helpful suggestions and expertise. I also wish to thank them for the equipment they provided which allowed me to set up the test network.

I am sincerely grateful to my fiancé, Mario, for his support and understanding during the course of my studies. I am especially indebted to him for his assistance throughout the data collection phase of this research.

I would like to acknowledge the funding provided by the Natural Sciences and Engineering Research Council in support of this research.

Contents

Acceptance Sheet	ii
Abstract	iii
Acknowledgments	iv
Table of Contents	v
List of Tables	viii
List of Figures	x
List of Acronyms	xii
1 Introduction	1
1.1 Traffic Modelling	2
1.2 Thesis Objectives and Contributions	4
1.3 Thesis Organization	7
2 Background	8
2.1 Stochastic Processes	8
2.1.1 Stationary Process	9
2.1.2 Traffic Characterization	10
2.2 Generic Traffic Models	12
2.2.1 Poisson Models	13
2.2.2 Markov Models	13
2.2.3 Autoregressive Models	15
2.2.4 TES Models	15
2.3 Network Traffic as Fractals	17

2.3.1	ON/OFF Models	17
2.3.2	Monofractal or Self-similar Models	19
2.3.3	Multifractal Models	21
2.3.4	Self-similar vs Multifractal models	23
3	Methodology	25
3.1	Experimental Methodology	25
3.2	Techniques in Distribution Fitting	27
3.2.1	Parameter Estimation	27
3.2.2	Ranking of Fits	28
4	Experimental Configurations	30
4.1	Network Prototype	30
4.1.1	Network Configuration	31
4.1.2	Link Configurations	35
4.2	Network Monitoring	37
4.2.1	TCPdump	38
4.2.2	WinPharaoh	38
4.3	Data Collection	39
4.3.1	Applications	39
4.3.2	Measurements	44
5	Results	47
5.1	Interarrival Time	48
5.2	Packet Length	56
5.3	Data Rate	63
5.4	Distributions of the Data Rates	70
6	Conclusion	78
	Bibliography	81

A Network Architecture	85
B TCPdump	90
C Detailed Measurements	94

List of Tables

4.1	Nodes and IP Addresses	35
5.1	Results of the Interarrival Times on the Link	49
5.2	Results of the Interarrival Times at the source	49
5.3	Results of the Packet Lengths on the Link	56
5.4	Results of the Packet Lengths at the source	57
5.5	Results of the Data Rates on the Link	64
5.6	Results of the Data Rates at the source	64
5.7	Distributions of the Data Rates	71
A.1	The functions of each OSI layers	87
A.2	The functions of each TCP/IP layers	88
C.1	Statistics of Interarrival times for Aggregate Data	95
C.2	Statistics of Interarrival times for Audio Data	96
C.3	Statistics of Interarrival times for GCCS-M Data	97
C.4	Statistics of Interarrival times for HTML Data	98
C.5	Statistics of Packet Lengths for Aggregate Data	99
C.6	Statistics of Packet Lengths for Audio Data	100
C.7	Statistics of Packet Lengths for GCCS-M Data	101
C.8	Statistics of Packet Lengths for HTML Data	102
C.9	Statistics of Data Rates for Aggregate Data	103
C.10	Statistics of Data Rates for Audio Data	104

C.11 Statistics of Data Rates for GCCS-M Data	105
C.12 Statistics of Data Rates for HTML Data	106

List of Figures

4.1	Network Prototype	31
4.2	Ethernet, PPP and LAPB in a Complete System	32
4.3	Set Up of the Distributed Collaborative Planning Tool	40
4.4	Set Up for the Common Operational Picture Exchange	42
4.5	Set Up for the Web Services	43
4.6	Set Up of all Three Applications	44
5.1	Autocorrelation for Interarrivals of Aggregate and Audio Link Traces	52
5.2	Autocorrelation for Interarrivals of GCCS-M and HTML Link Traces	53
5.3	Autocorrelation for Interarrivals of Aggregate and Audio Source Traces	54
5.4	Autocorrelation for Interarrivals of GCCS-M and HTML Source Traces	55
5.5	Autocorrelation for Packet Lengths of Aggregate and Audio Link Traces	59
5.6	Autocorrelation for Packet Lengths of GCCS-M and HTML Link Traces	60
5.7	Autocorrelation for Packet Lengths of Aggregate and Audio Source Traces	61
5.8	Autocorrelation for Packet Lengths of GCCS-M and HTML Source Traces	62
5.9	Autocorrelation for Data Rates of Aggregate and Audio Link Traces .	66
5.10	Autocorrelation for Data Rates of GCCS-M and HTML Link Traces .	67
5.11	Autocorrelation for Data Rates of Aggregate and Audio Source Traces	68
5.12	Autocorrelation for Data Rates of GCCS-M and HTML Source Traces	69
5.13	Distribution Fit of the Data Rates for Aggregate Traces	73

5.14	Distribution Fit of the Data Rates for Audio Traces	74
5.15	Distribution Fit of the Data Rates for GCCS-M Traces	75
5.16	Distribution Fit of the Data Rates for HTML Traces	76
A.1	Protocol Architecture	86

List of Acronyms

AR	Autoregressive
ARMA	Autoregressive Moving Average
ASCII	American Standard Code for Information Exchange
ATM	Asynchronous Transfer Mode
COP	Common Operational Picture
CSMA/CD	Carrier Sense Multiple Access with Collision Detect
DCE	Data Computer Equipment
DCP	Distributive Collaborative Planning
DTE	Data Terminal Equipment
EIA	Electronic Industries Association
FDDI	Fiber Distributive Data Interface
FTP	File Transfer Protocol
GCCS-M	Global Command and Control System - Maritime Extension
HP	Hewlett Packard
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IID	Independent Identically Distributed
IP	Internet Protocol
ISDN	Integrated Services Digital Network
K-S	Kolmogorov-Smirnov
LAN	Local Area Network
LAPB	Link Access Procedure Balanced

LCP	Link Control Protocol
MLE	Maximum Likelihood Estimator
MMPP	Markov Modulated Poisson Process
NCP	Network Control Protocol
OSI	Open System Interconnection
PC	Personal Computer
PDF	Probability Density Function
PPP	Point to Point Protocol
REPEAT	Repeatable Performance Evaluation and Analysis Tool
RX	Receiver
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
TES	Transform Expand Sample
TX	Transmitter
UDP	User Datagram Protocol
VBR	Variable Bit Rate
WAN	Wide Area Network
WSS	Wide Sense Stationary

Chapter 1

Introduction

A model is defined as a representation of a system for the purpose of studying the system. In the case of communication networks, an important element of a model is the random flow of traffic produced by LAN/WAN data. These random traffic models are then utilized as input to the network model in order to evaluate their effects on the network performance, usually to aid in explaining, understanding or improving the network. Also known as stochastic simulation models, these types of models are analyzed numerically, usually with the aid of a computer and have random inputs which lead to random outputs. Thus the results obtained can only be considered as estimates of the true characteristics of the system. Simulation is widely used when systems become too complex for theoretical techniques such as analytical queuing theory.

In this thesis, traffic from a narrowband network is examined in order to identify if current broadband network traffic models are still applicable. Since narrowband links often consist of radio frequencies or satellites, their error rate can differ from one link to another. For this reason traffic traces are collected from a link with different rates of error to identify if their statistics (variance, autocorrelation function, etc.) change as the errors of the link are augmented. Although in this era of broadband and high speed networks they are often forgotten, narrowband networks are still of interest to some

individuals, e.g. defence scientists. For instance, most of the canadian naval ships, when at sea, are unable to communicate at rates greater than 64Kbps. Currently research is being conducted in order to upgrade naval communication systems. This task could be simplified with the creation of appropriate simulation models.

1.1 Traffic Modelling

Traffic modelling is a key element in simulating communication networks. The success of modelling networks lies in understanding the nature of its traffic flow as well as selecting an appropriate random traffic model. However, in practice, it is difficult to develop an exact analytical traffic model. So a model must be described by statistically based measurements of one or more characteristics of the process. Only the steady-state, second-order and higher-order statistics of traffic are measurable. Steady-state statistics consist of the distribution function while second- and higher-order statistics are described by autocorrelation functions in the time-domain or by spectral functions (i.e. power spectrum) in the frequency domain. For more details on these traffic statistics the reader is referred to Section 2.1.

In today's networks, traffic "burstiness" seems to be of a major issue. Video and file transfers are among many services which tend to be bursty. Burstiness is present in traffic flow when runs of several relatively short interarrival times are followed by a relatively long one. Because of such a property, traditional models like the Poisson model, where aggregate traffic becomes smoother (less bursty) as the number of traffic sources increase, are no longer appropriate for traffic modelling. Current research is gearing into the direction of fractal-like models for describing the traffic flow of communication networks, i.e. self-similar and multifractal models.

In the past few years, self-similarity has been a main interest in modelling modern packet network traffic. The self-similar nature of traffic is explained by the high variability of individual connections that contribute to aggregate traffic. More recently,

web applications and ATM traffic have been offering a more complex scaling behavior, which has been explained by local traffic characteristics, such as end-to-end congestion control mechanism. These local irregularities operate over small-time scales. Since self-similar models work well over large-time scales but not over small ones, multifractal models, which do cover small-time scales are thus becoming a greater interest.

Self-similar traffic displays structural similarities across a very wide range of time scales. For instance on Ethernet LAN traffic, self-similarity is revealed in the absence of a natural length of a “burst”. At time scales ranging from a few milliseconds and above, bursts consist of bursty subperiods separated by less bursty subperiods. The Hurst parameter describing the degree of self-similarity measures the burstiness of LAN/WAN traffic [17, 24]. There has been significant progress in developing appropriate mathematical and statistical techniques for observed self-similar scaling behavior of measured data traffic over large-time scales (from hundreds of milliseconds to seconds and beyond). These techniques explain, describe and validate the reported large-time scaling phenomenon in aggregate network traffic at the packet level in terms of more elementary properties of traffic pattern operated by individual users and/or application [27].

An issue of interest is whether for more accurate description of traffic traces, a broader class of model, namely that of multifractal processes, has to be considered. The multifractal analysis is particularly appropriate in investigating the local fluctuations such as protocols and congestion control mechanism, of traffic traces. However, modelling for modern networks will avoid describing such localized features and will instead concentrate on more large-scale behavior. Taqqu et al. [28] investigate the application of multifractal models versus self-similar models. They suggest that with their sample means subtracted, traffic traces are well modelled by random processes that are either exactly or asymptotically self-similar and that it is not necessary to

increase the model class to that of general multifractal processes. Yet to completely describe network traffic, networking researchers argue that the local irregularities should not be ignored [7].

Studying the small-time scale behavior or local characteristics of data traffic is closely related to understanding the complex interactions that exist in data networks. Feldmann et al. [6] demonstrated that because of the increasing popularity of the Web, a scaling property of WAN traffic at the transport layer can be found. Thus traffic characteristics at the transport layer have gone through important changes that reveal the self-similar method proposed by Paxson and Floyd [24] to be inadequate for WAN traffic at the transport layer nor for small-time scales at the packet level. However, self-similar methods seem to be satisfactory for the LAN setting.

Multifractals suggest a class of models that provide a more complete and accurate description of actual traffic. Multifractals are appealing because of their close connection to certain multiplicative processes or cascade models. Networks act as cascades by fragmenting for example a web-session into a number of TCP connections which in turn are further fragmented into several HTTP requests which in turn are fragmented into IP flows and so on [6]. A cascade framework allows physical explanation of observed multifractal-scaling behavior of data traffic and allows refinement of the previously observed self-similar nature of data traffic to account for local irregularities in WAN traffic [7]. More precisely, a cascade is a fragmentation mechanism, which preserves the mass of the initial set or does so in expectation [7, 12]. Recent research of multifractals have been oriented in finding tools and adequate models for multiplicative processes [5, 6, 7, 11, 12, 13, 25, 26].

1.2 Thesis Objectives and Contributions

Today's research in network traffic modelling is mostly based on broadband and high speed networks. Fractal-like traffic models such as self-similar and multifractal models

have been successfully tested for these types of networks. However, some organizations, for instance the Canadian Navy, require research to be more developed in the area of simulation and modelling for narrowband networks. I refer to narrowband networks as in networks performing at low rates, e.g. 128Kbps and below. Because of the ongoing field trials and experiments being conducted to test possible upgrades for these communication networks, there is a need for the development of narrowband models. These models can be more cost effective and less time consuming than the current field tests.

One difference that exists between broadband and narrowband links is the elements which constrict the rates at which the traffic is transmitted, i.e. the bottlenecks. In broadband and high speed links the traffic rates are limited by the rates at which the devices (workstations and Personal Computers) can transmit their data. On the other hand, in narrowband links the traffic rates are limited by the links maximum rate capacity.

Does such a change in bottlenecks have an effect on the traffic models? For example, would the traffic model of audio based on measured traffic from a high speed link be the same if the measurements were done on a 64Kbps link? Also, since narrowband links often consist of radio frequencies or satellites their error rate can differ from one link to another. Are the traffic models independent of the links error rate? If the error rate on a link is increased, will a different traffic model be required?

The objectives of this thesis are to provide answers for the above questions. Therefore, this thesis consists of the following:

- To study the effects that increasing the error rate of a narrowband link has on the statistics of the traffic on the link as well as the traffic at the source level.
- To identify if the statistics and model of the aggregate traffic on a narrowband link differ from those of a broadband link.

The above goals were achieved as a result of the following accomplishments:

- A test network was built using a data link simulator to create a narrowband link from which error rates can be programmed.
- A network monitor called WinPharaoh was connected to the narrowband link enabling us to capture its traffic. As for the traffic at the source level, it was captured using a different monitor called TCPdump.
- A number of traffic traces were collected for different rates of error as well as for different types of data.
- A comparison study was conducted on the first- and second-order statistics of the traffic measurements (interarrival times, packet lengths, and data rates).
- A distribution fit was performed on the data rate measurements.
- A comparison study was conducted on the distributions of the measurements.

Contributions

As previously mentioned, despite the hype of high speed networks, narrowband networks are still being used and they will continue to be used in the future. The contributions of this thesis to research and development are as follows:

- This work shows that the distribution of narrowband traffic is not affected by the different rates of error found on narrowband links. This means that it will be possible to build narrowband traffic models regardless of the error rates.
- This work also shows that different traffic models are needed for narrowband networks than the ones currently used for broadband and high speed networks.

1.3 Thesis Organization

Chapter 2 starts off by giving a brief introduction to stochastic processes and network traffic characterization. It then continues on by providing a survey of the traffic models used in research from Poisson to fractals, including self-similar and multifractal models. In Chapter 3 the methodology followed to achieve the experiments is established. Also discussed are the techniques that are used to fit statistical distributions to the sampled data.

Chapter 4 covers in detail the set-up of the test network from the protocols to the configurations of the narrowband link. Also included in this chapter is a description of the schemes used for collecting the data as well as the details on the different types of data being measured. The results obtained from the analysis conducted on the measurements during the course of this research are presented in Chapter 5. Concluding remarks are found in Chapter 6.

Chapter 2

Background

2.1 Stochastic Processes

In the real world, most physical processes involve a random or stochastic element in their structure which increase the interest of studying stochastic processes. A stochastic process can be described as a statistical phenomenon that evolves in time according to probabilistic laws. It is a family of time indexed variables $X(\omega, t)$, where ω belongs to a sample space and t belongs to an index set.

For a given t , $X(\omega, t)$ is a random variable while for a given ω , $X(\omega, t)$, as a function of t , is called a sample function or a realization. With proper understanding that a stochastic process, $X(\omega, t)$, is a set of time indexed random variables defined on a sample space, the variable ω is usually suppressed and $X(\omega, t)$ is written as $X(t)$ or X_t .

A time series is a realization or a sample function from a certain stochastic process. Thus, a time series is a collection of observations made sequentially in time. There are various objectives for studying time series which include the understanding and description of a generating mechanism, forecasting of future values, and optimal control of a system. The essential property of a time series is that its observations are, in general, dependent or correlated, and the order of the observations is, therefore,

important. The body of statistical methodology available for analyzing time series is referred to as time series analysis.

Time series analysis is essentially concerned with evaluating the properties of the probability model which generated the time series. A simple way to represent a stochastic process is to give the moments of the process. More particularly the first and the second moments which are called the mean, variance and autocovariance functions. The mean function of the process is defined by

$$\mu_t = E(X_t), \quad (2.1)$$

the variance function of the process is defined by

$$\sigma_t^2 = E[(X_t - \mu_t)^2] \quad (2.2)$$

and the autocovariance function between X_{t_1} and X_{t_2} is defined by

$$\gamma(t_1, t_2) = E[(X_{t_1} - \mu_{t_1})(X_{t_2} - \mu_{t_2})]. \quad (2.3)$$

2.1.1 Stationary Process

Stationary processes are an important class of stochastic processes. A time series is said to be stationary if there is no systematic change in mean and in variance over time and if strictly periodic variations have been removed. Most of the probability theory of time series is concerned with stationary time series. For this reason, time series analysis often requires non-stationary series to be transformed as stationary so as to use the above theory. There are two types of stationary processes:

1. Strictly Stationary Process

A time series is said to be strictly stationary if the joint distribution of $X(t_1), \dots, X(t_n)$ is the same as the joint distribution of $X(t_1 + \tau), \dots, X(t_n + \tau)$ for all $t_1, \dots, t_n, \tau$. Thus, shifting the time origin by an amount τ has no effect on the joint distributions. The above definition must hold for any value of n .

More particularly, if $n = 1$, it implies that the distribution of $X(t)$ is the same for all t so that $\mu(t) = \mu$ and $\sigma^2(t) = \sigma^2$ are both constants that do not depend on the value of t . If $n = 2$, the joint distribution of $X(t_1)$ and $X(t_2)$ depends only on $(t_2 - t_1)$ which is called the lag. Thus, the autocovariance function, $\gamma(t_1, t_2)$, also depends only on $(t_2 - t_1)$ and may be written as $\gamma(\tau)$, where

$$\gamma(\tau) = E[(X(t) - \mu)(X(t + \tau) - \mu)] \quad (2.4)$$

is called the autocovariance coefficient at lag τ .

2. Weakly Stationary Process

It is often convenient to define stationarity in a less restrictive way than the one mentioned above. A process is said to be n^{th} order weakly stationary if all its joint moments up to order n exist and are independent of time origin. For instance, a second order weakly stationary process will have a constant mean and variance, with the covariance being a function of the time difference alone. This process is also known as wide sense stationary (WSS).

2.1.2 Traffic Characterization

Stochastic network traffic is often interpreted as the flow rate function of the data with respect to time, e.g. bits/sec. In discrete-time $\{X_n\}$ represents the quantity of data produced in one time unit and in continuous-time $\{X_t\}$ being the instantaneous rate of flow at time t . However, in some papers a discrete-time process $\{X_n\}$ represents the interarrival time between the k^{th} and the $k + 1^{th}$ discrete data unit. In this Thesis, both interpretations will be utilized to measure and model traffic. In order to not bring confusion to the reader, the interarrival time process will be denoted as $\{A_n\}$ and the quantity of data per unit time as $\{X_n\}$.

Another topic that is being discussed in the traffic modelling field is that of traffic “burstiness”. Burstiness is present in a traffic process if the $\{A_n\}$ process tends to give rise to runs of several short interarrival times followed by a relatively long one.

With typical network traffic exhibiting patterns and bursts that co-exist over many magnitude of time scales (from minutes to hours to days) comes the notion of time scale invariance [17]. Time scale refers to the change or immunity to change of the process structure upon scaling of time axis. A process $\{X_t\}$ can be defined as scaling invariant if for some $\alpha \in [a, b]$ the process is equal in distribution to its scaled version $\{X_{at}\}$. If a traffic is not scale invariant then when studying its behavior as time scale increase, it will show that the bursts and random fluctuations degenerate toward a white noise, non bursty type of traffic. This time scale characteristic, will be addressed further in Section 2.3.

In practice, it is difficult to develop an exact analytical traffic model thus, a different approach must be taken. One of the most established methods is that of identifying the steady-state first-order and second-order statistics of the traffic [3, 16, 17]. In other words, by establishing the marginal distribution and the autocorrelation function.

Marginal Distribution

The marginal distribution of a process $\{X_t\}$ captures the steady-state first-order distribution of X and is considered the primary characteristic in describing network traffic. Assuming that the process is WSS, the marginal distribution becomes invariant to time and is then defined by the one-dimensional probability density function (PDF)

$$f_X(x) = f_{X_t}(x) = \frac{d}{dx} Pr[X_t \leq x]. \quad (2.5)$$

The PDF describes the probability that the data will assume a value within some defined range at any instant of time.

Autocorrelation Function

The autocorrelation function of a process $\{X_t\}$ captures the second-order measurement of the process and it is used as a supplement to the marginal distribution.

The autocorrelation function for network traffic describes the general dependence of the values at another time. Assuming the process is WSS, then the autocorrelation between the data values at times t and $t + k$ is defined as follows:

$$\rho(k) = \frac{E[X_t X_{t+k}] - (E[X_t])^2}{E[(X_t - E[X_t])^2]}, \quad (2.6)$$

where k is called a “lag”, the difference between two relative time points. If the autocorrelation function $\rho(k)$ of $\{X_k\}$ is equal to zero for all values of $k \neq 0$, then $\{X_k\}$ is an independent and identically distributed (IID) process, e.g. Poisson process. Markovian and other models of Section 2.2 have a correlation structure that is characterized by an exponential decay. These processes are said to have a short-range dependence, thus $\sum_k \rho(k) < \infty$. On the other hand, the fractal models of Section 2.3 are defined to have long-range dependence. For these processes, the autocorrelation decays slowly in a way that it is not summable, thus $\sum_k \rho(k) \rightarrow \infty$. It is assumed to decrease polynomially as opposed to exponentially [23].

2.2 Generic Traffic Models

In its simplest form, network traffic consists of single arrivals of packets, cells or bytes. The arrival process, $\{T_n\}_{n=0}^{\infty}$, is a sequence of arrival instants $T_1, T_2, \dots, T_n, \dots$ where $T_0 = 0$. Counting processes and interarrival time processes are two other processes that are common in traffic modelling. A counting process, $\{N(t)\}_{t=0}^{\infty}$, is a continuous-time, non-negative, integer-valued stochastic process, where $N(t) = \max\{n : T_n \leq t\}$ is the number of event occurrences in the time interval $(0, t]$. An interarrival time process is a non-negative random sequence $\{A_n\}_{n=1}^{\infty}$, where $A_n = T_n - T_{n-1}$ is the length of the time interval separating the n^{th} arrival from the previous one. In the following subsections, generic models that can be used to generate simple traffic will be described. For further details on the following models the reader is referred to [10, 18, 21].

2.2.1 Poisson Models

Poisson models are one of the oldest traffic models. In the past, they were formally used to describe the packet flow in computer networks, telephony, etc. A Poisson process is a counting process with rate parameter λ and distribution

$$P[N(t) = k] = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad k = 0, 1, 2, \dots \quad (2.7)$$

On even terms, the interarrival times of this process form an IID sequence of exponential random variables with mean $\frac{1}{\lambda}$, and distribution $P[A_n \leq t] = 1 - \exp(-\lambda t)$ [18].

The Poisson process was a successful traffic model because of its unique analytical properties. Beside having independent and stationary increments, it consists of the following. The superposition of independent Poisson processes N_i , $i = 1, 2, \dots, m$ with rate λ_i results in a new Poisson process whose rate is $\lambda = \lambda_1 + \dots + \lambda_m$. Similarly, a Poisson process can also be split into multiple independent Poisson processes [21]. Such properties have had many repercussions in modelling computer systems. However, the use of Poisson models has led to numerous problems where customers complain of poor performance even for low traffic levels. These problems are caused by the fact that Poisson models do not account for the burstiness which is now a major part of network traffic. Consequently, different traffic models which do consider burstiness are now utilized in performance analysis methods [9, 17, 24].

2.2.2 Markov Models

A Markov process is a stochastic process that has a limited form of “historical” dependency. For instance, a random process, $X(t)$, is a Markov process if the future of the process depends on the current state and not on previous states, that is if it satisfies

$$P[X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k, \dots, X(t_1) = x_1] = P[X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k] \quad (2.8)$$

for any k , any arbitrary instants $t_1 < t_2 < \dots < t_k < t_{k+1}$ and any $x_1, x_2, \dots, x_k, x_{k+1}$. This is known as the Markov property [18, 21]. Markov processes can be defined in continuous-time, $\{X(t)\}$, or in discrete-time, $\{X_k\}$, referred to as a Markov chain. In a simple Markov traffic model, each state transition represents a new arrival. So the interarrival times are exponentially distributed and their rates depend on the state from which the transition occurred. This results in dependence among the random sequence $\{A_n\}$. Consequently, there is a possibility that they can capture the burstiness of network traffic [10].

The most general form of a Markov process is a semi-Markov process. This generalization is accomplished by allowing the holding time (time between state transitions) to follow an arbitrary probability distribution. However, this destroys the Markov property if times are not exponential. A semi-Markov process first selects its next state and then selects a value for its holding time. The sequence of states visited over time period $[t_k, t_{k+1})$ is denoted as follows:

$$(i_0, t_0), (i_1, t_1), (i_2, t_2), \dots, (i_k, t_k).$$

If the holding time distribution is ignored then the sequence of states will be a discrete time Markov chain, and is referred to as an embedded Markov chain [9, 21].

Markov Modulated processes are another important class of models. It consists of an underlying Markov process in which the current state of the Markov process determines the probability distribution of the traffic. The most commonly used model of this type is the Markov Modulated Poisson Process (MMPP), which utilizes Poisson processes as the modulated mechanism. In this case, while in state k , the event arrivals occur according to a Poisson process with rate λ_k [1]. MMPP can be used to model a mixture of voice and data traffic.

2.2.3 Autoregressive Models

The autoregressive model of order p , is a process, $\{X_t\}$, where its current value of the process is expressed as a finite, linear combination of previous values of the process and a white noise process $\{\epsilon_t\}$. A white noise process is a purely random process with mean zero and variance σ^2 . Mathematically it is represented as follows:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \epsilon_t, \quad (2.9)$$

where ϕ_i are constants and X_{t-i} are past realizations at time $t - i$. This is similar to a regression model except that the observations are dependent and the nature of this dependence is of interest in itself. An autoregressive model of order p will be abbreviated to an AR(p) model. The recursive form of this model makes it a popular candidate for modelling and generating autocorrelated traffic, e.g. VBR Video traffic. However, the autoregressive models cannot generally fit the empirical marginal distribution of video type traffic.

A second autoregressive model which is popular in traffic modelling is that of the Autoregressive Moving Average model of order (p, q) , ARMA(p, q). This process is of the following form.

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \dots - \theta_q \epsilon_{t-q}. \quad (2.10)$$

The ARMA model can be used to model VBR traffic. The estimation of its parameter are more involved then that of the AR models. For instance, to obtain an estimation of the θ_i 's, it is required to solve a set of nonlinear equations or to use spectral factorization techniques. It is also difficult to obtain the analytical solutions.

2.2.4 TES Models

Transform-Expand-Sample (TES) models are another commonly used method for modelling autocorrelated time series, such as network data. These models aim to capture both autocorrelation and marginal distribution of empirical data. The TES

model was developed by B. Melamed to model stationary, correlated time series and then to generate synthetic streams of realistic traffic to drive simulations of communications networks [10, 20]. Making use of stationary empirical time series, such as traffic traces, TES models attempt to specify the following three requirements:

- The marginal distribution of the model should match its empirical counterpart, i.e. histogram,
- The autocorrelation function of the model should approximate its empirical counterpart, and
- The sample paths generated by a simulation of the model should “resemble” its empirical counterpart.

Note: The first two are quantitative requirements while the last is considered a qualitative one. This third criteria can be use in addition to the other two in order to enhance our confidence in the model. In addition, there is a software package called *TEStool* which provides a visual interactive environment for fitting TES models to empirical data. It provides services to construct and modify TES models in a heuristic search mode, to generate simulated sample paths and to examine their statistics, as well as drive algorithmic searches [20].

The TES models consist of two major TES processes: TES^+ and TES^- . TES^+ produces a sequence which has positive correlation at lag 1, while TES^- produces a negative correlation at lag 1. In [20] Melamed et al. implied that the TES^+ version is most suitable for modelling network traffic while TES^- is more appropriate for time series with a zigzag sample path. For this reason only the TES^+ process will be defined in this thesis. For more information on the TES^- process the reader is referred to [1, 10, 20].

In order to define the TES^+ process, the *modulo-1* operation must first be introduced. The *modulo-1* of a real number x , denoted as $\langle x \rangle$, is defined as $\langle x \rangle = x - \lfloor x \rfloor$,

where $\lfloor x \rfloor$ is the maximum integer less than x . The recursive construction of the background TES^+ process is defined as follows:

$$U_n^+ = \begin{cases} U_0^+, & n = 0 \\ \langle U_{n-1}^+ + V_n \rangle, & n > 0 \end{cases}, \quad (2.11)$$

where $\{V_n\}$ is a sequence of IID random variables referred to as *innovations* and U_0^+ is uniformly distributed on $[0, 1)$ and independent of $\{V_n\}$. The resulting sequence $\{U_n^+\}$ has a $[0, 1)$ uniform marginal distribution, and an autocorrelation function determined by the probability density function $f_V(t)$ of V_n . The choice of $f_V(t)$ will result in a different correlation structure of the resulting process. From this background sequence the output process of the model referred to as the foreground sequence, $\{X_n^+\}$, is created by distorting each U_n^+ by $X_n^+ = F^{-1}(U_n^+)$, where F is the marginal distribution of the empirical data.

2.3 Network Traffic as Fractals

Recent research performed on LAN traffic [17] and WAN traffic [24] have challenged the commonly assumed models for network traffic, e.g. the Poisson distribution. If traffic were to follow a Poisson or Markovian arrival process, it would have a characteristic burst length which would tend to be smoothed by averaging over a long enough time scale. Rather, measurements of real traffic indicate that significant traffic variance (burstiness) is present on a wide range of time scales. This type of traffic can be described statistically by using long-range dependent and fractal models, e.g. self-similar and multifractal models.

2.3.1 ON/OFF Models

The fractal nature of network traffic is explained by the behavior of the individual connections that produce the aggregate traffic stream [19, 27, 30, 31]. In WAN traffic, individual connections correspond to “sessions”, where a session starts at a random

point in time, generates packets or bytes for some time and then stops transmitting. On the other hand, in LAN traffic, individual connections correspond to an individual source-destination pair. Individual connections are generally described using simple traffic models such as the ON/OFF sources [27, 30]. These models assume that a source alternates between an ON-period (active period), where packets or bytes are being transmitted and an OFF-period (idle period), where no packets or bytes are being transmitted. Strictly alternating ON/OFF sources are models where ON/OFF-periods strictly alternate and where ON-periods are IID, OFF-periods are also IID and ON/OFF-period sequences are independent from one another.

The traditional ON/OFF models assume finite variance distributions for the duration of the ON and OFF periods. The aggregation of a large number of these models result in very small correlations. However, an extension to these traditional models was presented by Likhanov et al [19] and Willinger et al [31] by allowing the ON or both the ON and OFF periods to be implemented using heavy-tailed distributions with infinite variance, e.g. Pareto distribution. The superposition of many such sources produces aggregate traffic that exhibit long-range dependence.

Heavy-tail property: Let Y be a positive random variable, then Y is called heavy-tailed with tail index α , if it satisfies

$$P[Y > y] = 1 - F(y) \approx cy^{-\alpha}, \quad y \rightarrow \infty, \quad 0 < \alpha < 2 \quad (2.12)$$

where $c > 0$ is a finite constant independent of y . This distribution has infinite variance; and if $1 < \alpha < 2$ then it has a finite mean.

Long-range dependence: Let $\rho(k)$ denote the autocorrelation function of a process $\{X_k\}$ as defined in Section 2.1.2. Then the process $\{X_k\}$ in which the autocorrelation takes the form

$$\rho(k) \approx ck^{-\beta}, \quad 0 < \beta < 1 \quad (2.13)$$

for large k and a constant $c > 0$, are said to be long-range dependent. This implies

that the autocorrelation function decays slowly in a way that is not summable, thus $\sum_k \rho(k) \rightarrow \infty$. Note that long-range dependence implies and is implied by asymptotic (second-order) self-similarity.

2.3.2 Monofractal or Self-similar Models

Self-similarity in a process indicates that some aspect of the process is invariant under scale-changing transformations, such as “zooming” in or out. In traffic, this is observed when traffic becomes bursty (significant traffic invariance) on many or all time scales. The convenience with the self-similar models is that the degree of self-similarity of a series is expressed using only one parameter. The Hurst parameter, defined as $H = 1 - \frac{\beta}{2}$, expresses the speed of decay of the series autocorrelation function. For self-similar series the value of H is between 0.5 and 1. The degree of self-similarity increases as the Hurst parameter approaches 1.

In the case of traffic traces, self-similarity is used in the distributional sense: when viewed at varying scales, the objects distribution remains unchanged. When distribution is used as the base of the equivalence between two processes, it is noted by placing a ‘ d ’ atop the equal sign. For instance, $X \stackrel{d}{=} Y$ indicates that the processes X and Y have the same distribution. If $X \stackrel{d}{=} aY + b$ is used, then it means that their distributions differ by the scale and location parameters, a and b respectively.

The following self-similar processes are defined as they are adopted by Tsybakov et al [29]. The definition of a self-similar time-series involves a second-order-stationary sequence $X = \{X_t : t = 1, 2, 3, \dots\}$. The mean of X is $\mu = E[X_t]$, the variance is $\sigma^2 = \text{var}(X_t)$ and the autocorrelation function is $r(k) = \frac{E[(X_{t+k} - \mu)(X_t - \mu)]}{\sigma^2}$

Let

$$X^{(m)}(t) = \frac{1}{m} (X_{tm-m+1} + \dots + X_{tm}) \quad m, t = 1, 2, 3, \dots \quad (2.14)$$

be the corresponding aggregated sequence with level of aggregation m , obtained by

dividing the original sequence X into non-overlapping blocks of size m and averaging over each block. The index t labels the block. For each $m = 1, 2, 3, \dots$ let $X^{(m)} = \{X^{(m)}(k) : k = 1, 2, 3, \dots\}$ denote the averaged process with autocorrelation function $r^{(m)}(k)$.

Second-order exact self-similarity: A process X is called exactly second-order self-similar with parameter $H = 1 - (\frac{\beta}{2})$, $0 < \beta < 1$ if its correlation coefficient is

$$r(k) = \frac{1}{2} \left[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta} \right], \quad k = 1, 2, 3, \dots \quad (2.15)$$

Exact self-similarity: A strict-sense stationary process X is called strictly self-similar with parameter $H = 1 - (\frac{\beta}{2})$, $0 < \beta < 1$, if

$$X \stackrel{d}{=} m^{1-H} X^{(m)}. \quad (2.16)$$

Note: If X is strictly self-similar, then it is exactly second-order self-similar. The opposite statement is not true, except for the case of a Gaussian process.

Second-order asymptotic self-similarity: A process X is called asymptotically second-order self-similar with parameter $H = 1 - (\frac{\beta}{2})$, $0 < \beta < 1$, if

$$\lim_{m \rightarrow \infty} r^{(m)}(k) = \frac{1}{2} \delta^2(k^{2-\beta}), \quad k = 1, 2, 3, \dots \quad (2.17)$$

where

$$\delta^2(f(x)) = f\left(x + \frac{1}{2}\right) - f\left(x - \frac{1}{2}\right).$$

Asymptotic self-similarity: A strict-sense stationary process X is called strictly asymptotically self-similar if

$$X^{(m)} \stackrel{d}{=} X, \quad m \rightarrow \infty \quad (2.18)$$

Note: In these cases, a strictly asymptotically self-similar process is not necessarily asymptotically second-order self-similar.

The additive property: The asymptotically self-similar behavior of aggregate network traffic is an additive property where individual connections exhibit certain heavy tailed characteristics [27]. Traffic self-similarity is revealed at time scales ranging from a few milliseconds and above. The number of packets or bytes per time unit collected off a link over large-time scales, from hundreds of milliseconds to seconds and beyond, originate from all individual connections that are active on this link. For $m \gg 1$, $X^{(m)} = \{X^{(m)}(k) : k \geq 0\}$ denotes the total number of packets or bytes per time unit (measured at time scale m) generated by all connections. Then

$$X^{(m)}(k) = \sum X_i^{(m)}(k), \quad k \geq 0, \quad (2.19)$$

where i is an active connection at time k and where $X_i^{(m)} = \{X_i^{(m)}(k) : k \geq 0\}$ represents the total number of packets or bytes per time unit (measured at time scale m) generated by connection i . Thus equation (2.19) describes the additive property of all network traffic, which expresses the total traffic rate $X^{(m)}$ as the superposition of the individual connection rate $X_i^{(m)}$.

Willinger et al [30, 31] provide a method of constructing self-similar processes. Their method states that by superimposing multiple heavy-tailed processes with infinite variances, one can construct self-similar processes. Therefore, due to its additive property and heavy-tailed characteristics, aggregated network traffic expresses self-similarity.

2.3.3 Multifractal Models

Currently researchers have been studying the possibility of modelling aggregate WAN traffic with multifractal processes [5, 6, 11, 27]. It appears that even though measured WAN traffic is consistent with asymptotic self-similarity, it also exhibits small-time

scaling features that differ from those observed over large-time scales. This small-time scaling behavior has been related to protocol specific mechanism and end-to-end congestion control algorithms that operate at those small-time scales (below a few hundreds of milliseconds). Modelling network traffic with multifractals has the potential of capturing the observed scaling phenomena at large as well as small time scales [6]. Thus, it naturally extends and improves the fractal or self-similar behavior of measured network traffic.

Multifractal traffic: To quantify the local variations in traffic at a particular point in time t_0 , let $Y = \{Y(t), 0 < t < 1\}$ denote the traffic rate process representing the total number of packets or bytes sent over a link in an interval $[t_0, t_0 + t]$.

The traffic has a local scaling exponent $\alpha(t_0)$ at time t_0 if the traffic rate process behaves like $t^{\alpha(t_0)}$ as $t \rightarrow 0$.

- $\alpha(t_0) > 1$ relates to instants with low intensity levels or small local variations, and
- $\alpha(t_0) < 1$ is found in region with high level of burstiness or local irregularities.

Note: If $\alpha(t_0)$ is constant for all t_0 , then the traffic is *monofractal* or if $\alpha(t_0) = H$ for all t_0 , then the traffic is exactly *self-similar*. H being the Hurst parameter as described in Section 2.3.2. On the other hand, if $\alpha(t_0)$ is non constant (varies in time), then the traffic is *multifractal* [27].

Multiplicative Cascades: WAN traffic is multifractal because certain multiplicative cascades lie in the background [7]. Multifractal processes are well modeled using multiplicative processes or conservative cascades. Conservative cascades is a fragmentation mechanism, which preserves the mass of the initial set or does so in expectation. The generator of the cascade is called the fragmentation rule and the mathematical construct that describes the way the mass has been redistributed is called the limiting object or multifractal. Modern data networks together with their protocols and

controls can be viewed as specifying the mechanisms and rules of a process that fragments units of information at one layer in the networking hierarchy into smaller units at the next layer, and so on.

The observed multifractal nature of network traffic over small time scales and the notion of an underlying cascade mechanism implies that over fine time scales network traffic is multiplicatively generated. In other words, the traffic rate process at those fine time scales has an approximately lognormal shape because it is the product of a large number of more or less independent “multipliers” [7, 27].

2.3.4 Self-similar vs Multifractal models

Multifractal processes are a generalization of self-similar processes. This implies the assumption that self-similar processes are multifractal, however the reverse can not be assumed. So how do we know which of the two types of models is appropriate to characterize the measured traffic traces? Taqqu et al [28] presented a method for distinguishing between the two models. In their paper, they concluded that traffic traces from the LAN environment were well modelled using self-similar models and that more sophisticated models such as multifractals were not needed. On the other hand, in the WAN environment, there are some cases where self-similar models are not adequate and where multifractal models may be more appropriate.

Self-similarity in Section 2.3.2 was defined through the equality of finite-dimensional distributions. In this section let it be defined through the behavior of the absolute moments. First let, $X = \{X(i), i \geq 1\}$ be a stationary sequence. Then the absolute moments are defined as follows:

$$\mu^{(m)}(q) = E|X^{(m)}|^q = E \left| \frac{1}{m} \sum_{i=1}^m X(i) \right|^q, \quad (2.20)$$

where $X^{(m)}$ is the aggregated sequence with a level of aggregation m . $X^{(m)}$ is obtained by dividing the original sequence, X , into non-overlapping blocks of size m and by

averaging over each block. If X is self-similar, then $\mu^{(m)}(q)$ is proportional to $m^{\beta(q)}$, that is, $\log \mu^{(m)}(q)$ is linear in $\log m$ for a fixed q :

$$\log \mu^{(m)}(q) = \beta(q) \log m + C(q). \quad (2.21)$$

In addition, the exponent $\beta(q)$ is linear with respect to q . In fact, since

$$X^{(m)}(i) \stackrel{d}{=} m^{H-1} X(i),$$

from Section 2.3.2, then we have

$$\beta(q) = q(H - 1). \quad (2.22)$$

Thus, the definition of self-similarity is simply that the moments must scale as in (2.21) and that $\beta(q)$ satisfies (2.22). This definition can also be generalized to that of multifractal processes. A non-negative process $X(t)$ will be called multifractal if the logarithm of the absolute moments scale linearly with the logarithm of m as in equation (2.21). Unlike self-similar processes equation (2.22) does not need to be satisfied. If the multifractal can take positive and negative values, then it is called a *signed multifractal* or *multiaffine*. In order to identify if a process is multifractal or self-similar, higher moments than the second order must be analyzed. Taqqu et al [28] suggest that only moments up to the 4th order (i.e. $q = 1, 2, 3, 4$) should be analyzed since higher order sample moments are not well-behaved. In summary, their method states that if equation (2.21) shows the moments to be scaling then the process is a (signed) multifractal. However, if equation (2.22) is also satisfied then the process is self-similar.

Chapter 3

Methodology

3.1 Experimental Methodology

In order to study the effects that increasing the error rate has on network traffic, we must be able to inject different rates of errors into a network link. The ideal solution to this problem is to have access to the links of a real network where it is possible to connect the necessary equipment to inject the errors into the links. However, adding errors into a link is not transparent to the end users and thus it can hinder the users' every day activities. A more desirable solution is to build a test network. Therefore, the first step in this thesis is to set up a network consisting of a narrowband link where errors can be easily introduced. This process is described in detail in Section 4.1.

Once the network is set up, the next stage is to collect traffic traces to be analyzed. In order to obtain traffic traces, we must have the means of monitoring the network. There exist various tools that are able to capture the packets of a network. The network monitors used in this thesis are TCPdump and WinPharaoh. TCPdump is a software-based monitor that is used on the local area network (LAN). On the other hand, WinPharaoh is a hardware-based monitor and it is used at the wide area network (WAN) level. These two monitors and their specifications are described in Section 4.2. Using these monitors, traffic traces are collected from three different

locations on the network, i.e. on the narrowband link and on the two Ethernets. The most important aspect in collecting data is to verify that the required information is present in the traces. For instance, in this work, the interarrival times between packets, the size of the packets and the data rates are analyzed. These measurements can be generated from the timestamps and the sizes of the packets being captured and therefore they must be recorded in the traffic traces. The details of the data collection are established in Section 4.3.

The analysis of the collected traces is a significant part of this thesis. It consists of evaluating the different traces for the first and second moments that are described in Section 2.1. For each set of measurements, the mean, the variance and the autocorrelation function are calculated using the Matlab package from The Math Works Inc. The measurements are then visually inspected to identify how they are affected when the errors on the narrowband link increase. These results are presented in Sections 5.1 to 5.3.

The last phase of this thesis is to identify the statistical distribution of the networks data rate. The data rate is defined as the number of bytes per second that are being transmitted on the network. In order to identify the distribution, the histogram of the data rates must be fitted to different distributions. This process is going to generate more than one possible fit and thus in some way from these possibilities the most appropriate fit must be selected. Fitting distributions to a set of data is a lengthy process. Therefore, in this work, a distribution fitting software called BestFit by Palisade Corporation is used. The BestFit software and the techniques it uses are described in the following section while the resulting distributions for the data rates are reported in Section 5.4.

3.2 Techniques in Distribution Fitting

BestFit is a software package that runs on Microsoft Windows 95 or higher which allows us to fit probability distributions to our data. Fitting is done when there is a set of collected data that needs to be described using a theoretical distribution function. BestFit offers many features for calculating best fits and analyzing the results. These features include:

- Twenty-eight available distribution functions,
- Parameter selection using maximum-likelihood estimators,
- Input up to 100,000 data points or pairs, and
- Complete statistics report, including goodness-of-fit, critical values and target values.

To get more information on the features of the BestFit package the reader is referred to its User Guide [22].

3.2.1 Parameter Estimation

For each of the distributions, BestFit tries to find the set of parameters that make the closest match between the distribution function and the data set. For sample data, distribution parameters are estimated using Maximum Likelihood Estimators (MLEs).

Maximum Likelihood Estimators

The MLEs of a distribution are the parameters of that function that maximize the probability of obtaining the given data set. For any density distribution $f(x)$ with one parameter α , and a corresponding set of n sampled values X_i , an expression called the *likelihood* may be defined:

$$L = \prod_{i=1}^n f(X_i, \alpha) \quad (3.1)$$

To find the MLE, simply maximize L with respect to α :

$$\frac{dL}{d\alpha} = 0 \quad (3.2)$$

and solve for α . This method can be easily generalized to distributions with more than one parameter.

3.2.2 Ranking of Fits

BestFit ranks all the fitted distributions using one or more fit statistics. These statistics measure how good the distribution fits the input data and how confident we can be that the data was produced by the distribution function. For each of these statistics, the smaller the value, the better the fit. For continuous sample data, the fits can be ranked by their chi-squared statistic, Anderson-Darling statistic, or Kolmogorov-Smirnov statistic. In this thesis, we decided to base our decision using both the chi-squared statistic and the Kolmogorov-Smirnov statistics. Each of these two statistics is discussed in turn.

Chi-Squared statistic

The chi-squared statistic is the most popular goodness-of-fit statistic. To calculate the chi-squared statistic, the x-axis domain is first broken up into several “bins”. The chi-squared statistic is then defined as:

$$\chi^2 = \sum_{i=1}^K \frac{(N_i - E_i)^2}{E_i} \quad (3.3)$$

where K is the number of bins, N_i is the observed number of samples in the i^{th} bin, and E_i is the expected number of samples in the i^{th} bin.

A weakness of the chi-squared statistic is that there are no clear guidelines for selecting the number and location of the bins. In some situations, different conclusions can be reached from the same data depending on how the bins are specified.

Kolmogorov-Smirnov statistic

The Kolmogorov-Smirnov statistic is defined as:

$$D_n = \max |F_n(x) - \hat{F}(x)| \quad (3.4)$$

where n is the total number of data points, $\hat{F}(x)$ is the fitted cumulative distribution function, and $F_n(x) = \frac{N_x}{n}$, where N_x is the number of X_i 's less than x .

The K-S statistic does not require binning, which makes it less arbitrary than the chi-squared statistic. A weakness of the K-S statistic is that it does not detect tail discrepancies very well.

Chapter 4

Experimental Configurations

4.1 Network Prototype

A generic prototype is created to simulate a narrowband network. Figure 4.1 is a diagram of the network utilized to obtain the traffic measurements. The test network is designed to simulate a simple wide area network (WAN) which consists of two local area networks (LANs) connected through a narrowband link. The two LANs are each composed of a workstation running TCPdump, a desktop PC or a HP station for generating traffic across the network and a router connected through a 10Mbps Ethernet Hub. The narrowband channel consists of a serial connection between the two routers. This connection includes a Data Link Simulator (ADTECH) used to establish the data rate of the link as well as to introduce errors into the link. A desktop PC running GN Nettests WinPharaoh is also connected on this link in order to monitor its traffic. The following sections consist of details on the configuration of this network. It defines the protocols that are selected to describe the network functions at the network and data link layers of the system. For a brief introduction on the network architecture and its layering concept the reader is referred to Appendix A.

Figure 4.1: Network Prototype

4.1.1 Network Configuration

The network testbed utilized in this thesis, see Figure 4.1, is composed of two different physical mediums. The LAN technology consists of the Ethernet protocol which is implemented on a 10BaseT cable. The “10” in 10BaseT means that the network operates at 10Mbps, “Base” refers to the fact that the cable is used in a *baseband* system, and “T” stands for twisted pair. On the other hand, the WAN technology consists of a point to point link implemented on a serial line running PPP and LAPB protocols. The serial line is connected through a RS-232 interface. “RS” stands for recommended standard. It is a standard interface approved by the Electronic Industries Association (EIA) for connecting serial devices which can operate as fast as 230.4Kbps. The serial line is needed in order to connect the ADTECH data link simulator to create the required narrowband link. The need for the link simulator is discussed further in Section 4.1.2. Figure 4.2 illustrates how the Ethernet, PPP and LAPB protocols are connected in a system using TCP/IP.

Figure 4.2: Ethernet, PPP and LAPB in a Complete System

Ethernet Protocol

The Ethernet is a working example of the more general Carrier Sense, Multiple Access with Collision Detect (CSMA/CD) local area network technology. Thus, a set of nodes send and receive frames over a shared link. The “carrier sense” in CSMA/CD means that all the nodes can distinguish between an idle and a busy link, and “collision detect” means that a node listens as it transmits and can therefore detect when a frame it is transmitting has collided with a frame transmitted by another node. When the adaptor has a frame to send and the line is idle, it transmits the frame immediately. If the line is busy, it waits for the line to go idle and then transmits immediately. If a collision is detected, the adaptor first makes sure to transmit at least 512 bits of the frame and then stops the transmission. It must send 512 bits (delay $\times$ bandwidth) in order to make sure that the two transmitting adaptors on opposite ends of the Ethernet can both detect the collision. Once the adaptor has detected a collision and stopped its transmission, it waits a certain amount of time and tries again. Each time it tries to transmit but fails, the adaptor doubles the amount of time it waits before trying again. This technique is known as *exponential backoff*.

Each frame transmitted on an Ethernet is received by every adaptor connected to that Ethernet. Each adaptor recognizes those frames addressed to its address and passes only those frames on to the host.

Point to Point Protocol (PPP)

PPP is designed to transport datagrams from multiple protocols over point-to-point links. It runs on virtually all media that are full duplex in nature. The two principal means of communication on a serial line are *asynchronous* and *synchronous*. Asynchronous line hardware can send and receive one character at a time. Synchronous line hardware can send or receive a variable length block of bytes at one time. The network prototype in this research uses a synchronous line. Before user information can be sent across a point-to-point link, each two endpoint systems must test the link and negotiate an agreement regarding the parameters under which the link will operate. These functions are performed using the Link Control Protocol (LCP). Once the link is established, PPP must send Network Control Protocol (NCP) packets to negotiate the network layer protocol (i.e. IP) that will be supported in this link. Once the link is established and both peers have agreed to support a given network layer protocol on this link, datagrams from that network layer protocol may be sent over the link. PPP assumes that datagrams arrive in the order that they are sent, therefore does not provide any reliability. If a datagram is received corrupted, it will simply drop it. However, if reliable transmission is required, then the LAPB protocol can be added to enhance PPP, see Figure 4.2.

Link Access Procedure Balanced (LAPB)

LAPB is a bit-oriented synchronous protocol that provides complete data transparency in a full-duplex point-to-point operation. A minimum of overhead is required to ensure flow control, error detection and recovery, i.e. LAPB adds a sequence number to the PPP datagram. The concept of a frame window is used to send multiple frames before receiving confirmation that the first frame has been correctly received.

This means that data can continue to flow in situations where there may be long “turn-around” time lags without stopping to wait for an acknowledgment. This kind of situation occurs, for instance in satellite communication. The window sizes vary, but are typically 7 frames for most terrestrial lines and up to 128 frames for satellite links. LAPB must receive the frames in the order that they are sent. Thus if a frame is corrupted, dropped or is out of order, LAPB will request a retransmission of the next expected frame and all the other frames that were sent after it. Because the sender has to backtrack by some number of frames (potentially as many as the window size), this scheme is known as *go-back-n*. In the network prototype, LAPB is run on a serial line, thus all packets should be received in the order they were sent. However, since errors will be introduced to the link, see Section 4.1.2, LAPB will be used to identify if a frame is corrupted or lost and to request its retransmission.

Addressing

In order to provide successful host-to-host connectivity each node must be able to say which of the other nodes on the network it wants to communicate with. This is done by assigning an *address* to each node. An address is a byte string that identifies a node. When a source node wants the network to deliver a message to a certain destination node, it specifies the address of the destination node. If the sending and receiving nodes are not directly connected, then the switches and routers of the network use this address to decide how to forward the message toward the destination.

The IP service model provides an addressing scheme in which IP addresses are *hierarchical*, meaning that they are made up of several parts that correspond to some sort of hierarchy in the internetwork. Specifically, IP addresses consist of two parts, a network part and a host part. The network part identifies the network to which the host is attached; all hosts attached to the same network have the same network part in their IP addresses. The host part then identifies each host uniquely on the particular network. By convention, IP addresses are written as four decimal integers

IP Address	Node
Yellow LAN	
192.168.2.1	3COM Router
192.168.2.5	HP Station
192.168.2.10	Personal Computer(PC)
192.168.2.50	Workstation
Green LAN	
192.168.3.1	3COM Router
192.168.3.5	HP Station
192.168.3.10	Personal Computer(PC)
192.168.3.50	Workstation

Table 4.1: Nodes and IP Addresses

separated by dots. Each integer represents the decimal value contained in 1 byte of the address. For example, 192.168.2.10 is the address of one of the Personal Computers (PCs) used in the network prototype. It is host 10 of the local area network 192.168.2.

The network prototype, as described at beginning of Section 4.1 consists of two LANs. Instead of identifying the two LANs by a numerical name (e.g. LAN1 and LAN2), they are distinguished by the color of their Ethernet cables, thus Yellow LAN and Green LAN. This color scheme is also properly illustrated in Figure 4.1. The Yellow LAN is illustrated using a yellow Ethernet and the Green LAN using a green Ethernet. The IP addresses used for these two LANs and their corresponding nodes are identified in Table 4.1.

4.1.2 Link Configurations

The purpose of this research is to identify if the characteristics of a narrowband data link affects the statistics of the traffic going through it. A narrowband data link

can be any link that operate at a data rate which is lower than 128Kbps, e.g. radio frequency links, satellite links and even terrestrial links such as ISDN lines. All these links operate at different data rates and have different error rates. Does the traffic of a satellite link with a 10^{-6} error rate have the same statistics (i.e. autocorrelation, variance) as the traffic of a satellite link with a 10^{-5} error rate? In order to be able to answer this question, traffic measurements are needed from a link with different error rates. To accomplish this, the ADTECH model SX/12 Data Channel Simulator is connected to the serial link of the network illustrated in Figure 4.1. This is the reason why a serial link is needed between the two 3COM routers.

The Data Channel Simulator creates a controlled environment for testing, tuning, or demonstrating data communication equipment. It can also be added to real data channels to insert additional delays and errors to test their effects on the on-line communications equipment. Among its functions it contains a data rate generator, a delay generator and an error rate generator. This work makes use of the data rate and error rate parameters. The delay parameter remains set at zero. The SX/12 contains two data channels which corresponds to the two directions the data is transmitted on the link.

The SX/12 contains an error generator that has four bit-error sources where error rates can range from 1×10^{-9} to as high as 1 error per bit. Two of the bit-error sources are used to inject random errors into each channel while the other two are used to generate burst errors for each channel. The link utilized in this research only uses the random errors and thus only this feature is described. For more information on the burst errors the reader is referred to the SX/12 Operating Manual [2].

The random error rates are individually set for each of the channels. Both the mantissa and the exponent of the error rate are entered. An exponent of zero gives an error rate of 1 error per bit. A mantissa of zero turns off the random errors. The

random errors are used mainly to simulate background errors caused by Gaussian noise. The following rates are used to inject errors into the link in order to obtain the measurements:

- 0,
- 5×10^{-6} ,
- 1×10^{-5} ,
- 5×10^{-5} , and
- 1×10^{-4} (not always possible).

The error rate of 1×10^{-4} can not always be successfully injected in the link since it can cause some of the applications to perform unsatisfactorily. For instance, with audio the quality at this rate is generally poor. It is very difficult to understand and sometimes the voice is truncated. Another application, which can not be achieved at this error rate, is the web application. It takes between 10 to 15 minutes to simply view a complete web page which is residing on the opposite LAN. In these cases, the traffic measurements are omitted for this rate.

Each of the error rates are tested on a 64Kbps channel. The 64Kbps link is achieved by setting the transmission rate of the serial port on the 3COM routers to that specific rate. The SX/12 data rate generator is also set up to transmit at 64Kbps. The different types of data that are measured and the methods used to obtain their measurements are described in the following sections.

4.2 Network Monitoring

In order to model and analyze network traffic, a network must first be monitored to capture all packets being transmitted. The data captures are referred to as traffic traces. In this thesis, a software package called TCPdump, developed by the

Lawrence Berkeley Laboratory, is used to achieve this goal on the local area network. In addition, a hardware-based analyzer called WinPharaoh from GN Nettest is used to monitor at the wide area network level.

4.2.1 TCPdump

TCPdump is a Unix software base network monitoring and data acquisition tool. It prints out the headers of packets on a network interface that matches a boolean expression. The expression selects which packets are captured. If no expression is given, all packets on the network are captured. Otherwise, only packets for which the expression is 'true' are captured. The TCPdump command consists of the following:

```
tcpdump [-e n tt][-r file][-w file] expression
```

The components of the above command are described in details in Appendix B. Examples as well as the commands output format are also included.

For analysis purposes the timestamps and packet lengths of each packet need to be extracted out of the TCPdump output. By using the -tt and -e options this can be achieved with the awk 'print \$*n*' command. Where *n* is 1 for the timestamp, the first element of the output lines, and 5 for the packet length, the fifth element of the output lines. By redirecting this information to an ASCII file, it can then be used by any analysis tool desired.

4.2.2 WinPharaoh

WinPharaoh is a high performance, hardware based analyzer capable of testing both sides of interconnection devices simultaneously and accurately timestamping the captured data. The WinPharaoh analyzer has a Windows based interface running on a desktop PC which makes it easier to handle than other analyzers, e.g. TCPdump, snoop. The WinPharaoh hardware-based interface adapters are designed for the industry standard plug-in ISA bus. Adapter cards can be installed individually or in

different configurations such as multiple ATM/LAN/WAN interfaces, which make multi-segment analysis possible. In this thesis, a single-card interface is used to capture data at the WAN level of the network. WinPharaoh is very practical in that it provides an adapter card capable of monitoring a serial line. Recall from Section 4.1 that the physical medium of the narrowband link in the network is a serial line.

WinPharaoh has been designed to take advantage of the Windows Graphical User Interface which render most common activities to be configured or tested with just a few clicks of the mouse. The WinPharaoh software provides a complete network analysis solution. It comprises many features and utilities that will not be utilized or described in this work. For more details on these functions the reader is referred to the WinPharaoh Users Guide [14]. In this thesis, WinPharaoh is used to monitor and record all the packets going through the narrowband link, including the layer two generated packets. From the measured traffic trace, it is then possible to export the required information (i.e. timestamps and packet sizes) into a more suitable format.

4.3 Data Collection

4.3.1 Applications

In this section, the applications used to create the different data measurements are briefly described. Three type of data applications are used to create the traffic on the network. They are first examined individually in order to determine their traffic characteristics and their statistical behavior related to the errors introduced on the link. Then, the three data applications are aggregated together to create the last set of traffic measurements. The three applications are selected based on a multi-national naval exercise. It is an annual exercise which involves real and simulated ships, shore stations and Marine forces linked together over a multi-national Coalition Wide Area Network. In these exercises the applications are used and tested on

Figure 4.3: Set Up of the Distributed Collaborative Planning Tool

various narrowband link ranging from 9.6Kbps to 64Kbps. Therefore, by using the same applications, we are creating a testing environment which is realistic in nature.

Distributed Collaborative Planning (DCP) Tool

One of the DCP tool used in the naval exercises and therefore used in this research is Microsoft Netmeeting which runs on Microsoft Windows 98 or NT. This tool provides a number of features that can be used for collaborative planning: chat, whiteboard, video and voice. In order to simplify matters, only the audio conferencing tool is used. Netmeeting uses UDP, (an un-reliable connectionless but efficient protocol) to transmit voice. It only transmits data when the microphone picks up sounds above the threshold that is set within the application, by the user.

The Netmeeting software is installed on the PC of the Yellow LAN as well as the PC of the Green LAN. The set up of this scenario is illustrated in Figure 4.3. A Netmeeting call needs to be established between the two PCs in order to have communication. From there an audio session can be conducted between the two nodes.

To be able to repeat the voice exchange between the two users a script containing the dialogue is used as the basis of the conversation.

Common Operational Picture (COP) Exchange

The delivery of the COP in the naval exercises as well as in the network prototype used in this thesis are exchanged using the Global Command and Control System - Maritime Extension (GCCS-M) which runs on a HP-TAC-4 compatible workstation. This system periodically receives text only, position report, messages which are then processed and displayed in a graphical representation. To simulate these periodic reports, a tool developed by SPAWAR (Space and Warfare System Center) San Diego, called "REPEAT" is used. The Repeatable Performance Evaluation and Analysis Tool (REPEAT) software which is loaded on a PC uses database files and plays them back in a random fashion simulating real life.

The GCCS-M station receiving information from REPEAT is considered the master station. It is the station in charge of delivering the COPs to all the other GCCS-M stations in the network. The master station is referred to as the transmitter and the other stations as the receivers. There is only one transmitting station and there is no limit on the receiving stations. The REPEAT software sends its data to the GCCS-M software through a serial connection, see Figure 4.4. In the set up, the transmit station resides on the Yellow LAN and a receive station is on the Green LAN. The REPEAT database files that are used are the same ones that were used in the naval exercises of July 1999.

Web Services

Accessing the World Wide Web via HTTP servers is not a problem, in general, when access to the Internet is available. However, for a military network, the WAN must be completely separate from the Internet for security reasons. In an event such as this, the web servers must be provided internally (on the military WAN). These mil-

Figure 4.4: Set Up for the Common Operational Picture Exchange

itary web servers contain information that is pertinent to their missions. The data includes intelligence data, imagery, operational status data, etc. To be able to access this information, a web client is needed to allow them to browse the web pages found on the servers.

The network prototype in this thesis is set up in the same manner. The corresponding diagram is displayed in Figure 4.5. The Apache HTTP web server, created by the Apache Software Foundation, is installed on the workstation of the Green LAN which is also running the TCPdump software. Approximately 100 web pages are downloaded from the Internet and installed on the web server for viewing purposes. On the Yellow LAN, the PC, which is running the Microsoft Internet Explorer web client, is used for browsing the pages found on the web server. In order to automate the web browsing during the experiments, Frederic Massicotte of the Communication Research Center Canada, provided a Visual Basic program that accesses the web pages of the server in a random order using the Internet Explorer web client. The program completely downloads the web page then waits a constant amount of

Figure 4.5: Set Up for the Web Services

time before accessing another page. That way, by removing the constant wait time (OFF period) from the measurements, only the active download time of the pages (ON period) are statistically characterized and studied.

Aggregated Traffic

By running the three applications mentioned above simultaneously, the traffic measurements taken are a mix of the three types of traffic. This is referred to as aggregated traffic. This scenario is used because on a typical network there are multiple users and therefore multiple sessions running between the nodes. Figure 4.6 is a diagram of the set up used with the aggregate traffic. The difference between this set up and the ones of the individual applications is that the REPEAT software is set up on a Laptop computer instead of a PC. The reason for this is that the REPEAT software is not based on Windows but rather it is based on DOS. This required the software to be the only application running on that particular computer or else the time clock of REPEAT gets slowed down and the information is not sent to the GCCS-M station at the right time. Since the two PCs in the network are being used for the Netmeet-

Figure 4.6: Set Up of all Three Applications

ing Session, another computer is required, hence the laptop. The rest of the set up resembles what was previously presented.

4.3.2 Measurements

There are four different types of measurements. They are referred to as audio, GCCS, HTML and aggregate. Each set of measurements is taken in three different places on the network. On the Green LAN and on the Yellow LAN each being collected using the TCPdump monitor as well as on the serial (narrowband) link using the WinPharaoh monitor. The traffic traces taken on the LANs consist of only the layer four packets while the traces of the link are composed of both the layer four and the layer two packets. The reason for this is that the measurements taken on the Ethernets are to be as close to the source generators as possible. The reason why the measurements of the link need to include the layer two packets is because if the statistical characteristics of the traffic change as the errors increase, it can be caused by the retransmissions, which are only occurring at layer two.

Each set of measurements are performed twice (except for the aggregate set) and are referred to as version 1 and version 2. The sets consist of traffic traces where the different error rates described in Section 4.1.2 are injected into the link. There is a trace for each error rate except for that of 1×10^{-4} which is only measured for the GCCS traces and version 1 of audio. The 10^{-4} measurements are not repeated for the second set of audio traces because the voice quality is too poor. The voice is often choppy and sometimes it is even completely cut off (lost). As for the HTML traces with 10^{-4} , it takes too long to load a web page, i.e. in 30 minutes only 3 web pages have been viewed. Thus it is not possible to obtain a good trace to compare to the other traces.

Because there are a lot of different traces being collected, they need to be well identified. The following format is being used to create the filenames:

[Position][Data rate][Type][Error rate][Version].dmp

Position: indicates at which place in the network the data is collected. The choices are 'y' for the Yellow LAN, 'g' for the Green LAN and 'l' for the link.

Data rate: indicates the transmission rate of the link. '16' and '64' for data rates of 16Kbps and 64Kbps respectfully.

Type: describes which set of measurements the trace is from. The options are the following: 'aud' for audio, 'agg' for aggregate, 'htm' for HTML, 'gtx' for the GCCS transmitter (on the Yellow LAN), 'grx' of the GCCS receiver on the Green LAN and 'gcc' for GCCS traffic on the link.

Error rate: identifies which error rate is injected in the link for that trace. They are defined as follows: '0e0' for no injected error, '5e6' for 5×10^{-6} , '1e5' for 1×10^{-5} , '5e5' for 5×10^{-5} , and '1e4' for 1×10^{-4} .

Version: distinguishes between the two sets of traces, 'v1' for version 1 and 'v2' for version 2. This part of the filename is omitted for the aggregate measurements

since there is only one set of them.

For example, the filename of a traffic trace for the web services taken on the Green LAN with the narrowband link consisting of a 64Kbps data rate and an error rate of 1×10^{-5} is g64html1e5v1.dmp for the version 1 of the measurements.

From these traces any data set that is required can be extracted. In order to analyze the statistical characteristics of the traffic traces, data sets are created containing the packet arrival times, the interarrival times between packets as well as the packet lengths. Each data set has the same name as the traces they are extracted from. The only difference is the extension of the filename. For instance, data sets of packet arrival times have the extension .tim, interarrival times have the extension .int and packet lengths have the extension .dat.

In the case of the traffic traces taken on the link, a little more work needs to be done since the traces consist of duplex traffic, i.e traffic is collected from both directions on the link. Consequently, when the data sets are extracted from the traces they are divided into two files, one for each direction. In this case the Version identifier of the filename will be changed to include the direction, e.g. v1dte is the first set taken from the direction DTE to DCE and v1dce is the first set taken in the direction DCE to DTE. The DTE and DCE terminology identifies the connectors for a RS-232 serial cable. In this work, the direction of DTE to DCE represents the data going from the Green LAN to the Yellow LAN and vice versa.

Chapter 5

Results

Using the experimental set-up described in Section 3.1, many experiments were conducted on the Test Network of Chapter 4. The results of these experiments are categorized in the following manner:

- Trial 1: The statistics of the interarrival times between the packets.
- Trial 2: The statistics of the packet lengths.
- Trial 3: The statistics of the sources transmission rate and the links data rate.
- Trial 4: The statistical distribution of the transmission and data rates.

The data collected during the experiments is analyzed using the Matlab package from The Math Works Inc. The monitoring agents described in Section 4.2 recorded the measurements in several different formats. From these recorded traces the packet arrival times (timestamps) and the packet lengths are exported into two text files. These text files are then imported into Microsoft Excel in order to manipulate the data into the desired format. For instance, the timestamps from the WinPharaoh monitor are converted from the form minutes:seconds.microseconds to seconds.microseconds.

A simple Matlab routine is used to create, from the timestamps, text files containing the interarrival times between the packets. Using a similar routine, text files

of data rates are created from both the timestamps and the packet lengths. The measurements that are required for Trial 1 to Trial 3 are obtained by analyzing the above text files using Matlab. The results are presented in turn in the following sections. To obtain the distributions of Trial 4, a distribution fitting software called BestFit 4.0 from the Palisade Corporation is utilized, see Section 3.2. These results are presented in Section 5.4.

5.1 Interarrival Time

As explained previously, the goal of this thesis is to identify the effects that increasing the error rate of the link has on the traffic. The first aspect of the traffic studied is that of the interarrival times between packets. The measurements of the interarrival times are collected for each error rate injected into the link. Each set of measurements is then compared to one another in order to identify if any change occurs due to the increase in errors. The comparison is done for both the traffic on the link as well as the traffic collected close to the source. The detailed measurements of the mean and variance from these experiments are supplied in Tables C.1 to C.4 of Appendix C.

Along with a summary of the measurements, this section also includes a few comparison graphs of the autocorrelation functions. Since including all the autocorrelation graphs would be too cumbersome, only the graphs of the Version 1 traces and of that taken in the direction of the Yellow LAN to the Green LAN are reported. Table 5.1 summarizes the changes on the interarrival times of the packets that were captured on the narrowband link. As for Table 5.2, it summarizes the changes on the interarrival times of the packets captured close to the sources, i.e. on the Ethernets. These changes are observed by comparing the set of measurements obtained when the different error rates were injected into the narrowband link, see Appendix C.

In both cases, on the link and at the source, the mean and variance of the interarrival times generally tend to increase as the error rate increase, except for the

Interarrival Times on Link				
	Error Rate	Mean	Variance	Autocorrelation
Aggregate	↑	↑	↑	≡
Audio	↑	↑	↑	≠
GCCS-M	↑	I	I	≠
HTTP	↑	↑	↑	≠

↑ - Increase, ↓ - Decrease, I - Inconclusive
 ≡ - No Significant Change, ≠ - Changes Observed

Table 5.1: Results of the Interarrival Times on the Link

Interarrival Times at Source				
	Error Rate	Mean	Variance	Autocorrelation
Aggregate	↑	↑	↑	≠
Audio	↑	↑	↑	≠
GCCS-M	↑	↓	↓	≡
HTTP	↑	↑	↑	≡

↑ - Increase, ↓ - Decrease, I - Inconclusive
 ≡ - No Significant Change, ≠ - Changes Observed

Table 5.2: Results of the Interarrival Times at the source

GCCS-M data. When viewing the detailed measurements of Appendix C, it is observed that the increase is more apparent in the measurements taken at the source. This is caused by the fact that when an error occurs the retransmission of the packet is done at the layer 2 level and therefore, it is transparent to the source. This means that only a delay can be noticed at the source. This delay is what causes the increase in the interarrival times between the packets. On the other hand, in the case of the link, the delays are not apparent because the layer 2 packets are captured and thus the retransmission is reported. There is not much of an increase in the first few measurements where the error rates ranged from 0 to 1×10^{-5} . The effects of the errors are more apparent when the error rate is high, i.e. from 5×10^{-5} to 1×10^{-4} . This can be explained by the delays found at the source. As the errors increase so do the delays since the more there are errors, the more there are retransmissions and sometimes retransmissions are corrupted and so on. Because there are larger delays at the source, what might happen is the TCP/IP flow control starts to throttle back and thus the source is now transmitting at a slower rate. This causes the interarrival times of the packets to increase and thus it is reflected on the packets captured on the link. This does not happen in the case of the smaller error rates because the delays at the source are not big enough to cause the TCP/IP flow control to take action.

As for the GCCS-M traces, the effects caused by the increase of error is different than those of the other traces, see Table C.3. The mean and variance of the interarrival times between the packets captured close to the source decreases as the error rate increase. This is not unexpected because of the way the system operates. The REPEAT software was setup to send data to the transmitting GCCS-M station at a constant interval of time, approximately every 60 seconds. This means that there is approximately 60 seconds between the start of one ON period and the start of the next ON period. It does not however, mean that the OFF period is 60 seconds since the ON periods are not of equal lengths. This created large interarrival times between the last transmitted packet of a set and the first packet to be transmitted

from the next set, i.e. this is the length of the OFF period. However when error is introduced into the link, and thus delay is experienced at the source, it causes the GCCS-M station to be transmitting for a slightly longer period than if there were no delay. This means the gaps between the sets of transmissions, the OFF periods, are getting shorter as the ON periods are getting longer and therefore, there is a decrease in the measurements as the amount of errors increase.

Figures 5.1 to 5.4 shows some of the comparison graphs obtained from the autocorrelation functions of the interarrival times for the different rates of error. What these graphs, along with those that are not included, imply is that there can be some variations in the autocorrelation function of the interarrival times when errors are introduced into the link. Tables 5.1 and 5.2 suggest that some changes are observed in the autocorrelation of the interarrival measurements taken on the link for the audio, GCCS-M and HTML traces as well as those taken at the source for the aggregate and audio traces.

In the case of the traffic traces captured on the narrowband link, if some periodicity is occurring in the autocorrelation it seems to disappear as more errors are injected into the link. Also, as the errors increase, the autocorrelation appears to decrease towards zero. These trends are mostly seen in the traces of the GCCS and HTML graphs, see Figure 5.2. This was also the case for the aggregate traces of the link for the traffic going from the Green LAN to the Yellow LAN, not shown in this work. As for the audio traces, see Figure 5.1, there is no periodicity in the autocorrelation functions, however it is quite evident that as the error rates are high the autocorrelation is zero.

On the other hand for the traces captured close to the source, the autocorrelation functions are not as affected from the increase in errors, see Figures 5.3 and 5.4. However a slight change can be found in the graphs of the aggregate and audio traces

Figure 5.1: Autocorrelation for Interarrivals of Aggregate and Audio Link Traces

Legend: Yellow $\rightarrow 0$, Green $\rightarrow 5 \times 10^{-6}$, Magenta $\rightarrow 1 \times 10^{-5}$,
 Blue $\rightarrow 5 \times 10^{-5}$, Red $\rightarrow 1 \times 10^{-4}$

Figure 5.2: Autocorrelation for Interarrivals of GCCS-M and HTML Link Traces

Figure 5.3: Autocorrelation for Interarrivals of Aggregate and Audio Source Traces

Figure 5.4: Autocorrelation for Interarrivals of GCCS-M and HTML Source Traces

Packet Lengths on Link				
	Error Rate	Mean	Variance	Autocorrelation
Aggregate	↑	≡	↑	≠
Audio	↑	≡	≡	≡
GCCS-M (TX)	↑	↑	↑	≠
GCCS-M (RX)	↑	≡	≡	≠
HTTP Browser	↑	↓	↓	≠
HTTP Server	↑	↑	↑	≠

↑ - Increase, ↓ - Decrease, I - Inconclusive

≡ - No Significant Change, ≠ - Changes Observed

Table 5.3: Results of the Packet Lengths on the Link

for lags that are less than 10. As the error rates increase the autocorrelation function in those two cases tend to decrease to zero at a slightly slower rate. No significant change is observed for the GCCS-M and HTML traces.

5.2 Packet Length

In this section, the effects of increasing the error rate on the packet lengths are identified. The detailed measurements are also available in Appendix C, in Tables C.5 to C.8. The summary tables presented in this section are divided differently than the tables of Section 5.1. The measurements for the GCCS-M and web traces are arranged into two categories relating to the direction in which the traffic was flowing. For instance, the measurements for the GCCS-M transmitter (TX) and receiver (RX) are analyzed separately. The same goes for the HTTP browser and the HTTP server. The reason for not generalizing the applications as one category is because they are not responding to the errors in the same manner. Table 5.3 summarizes the effects of increasing the error rates on the size of the packets being captured on the narrowband

Packet Lengths at Source				
	Error Rate	Mean	Variance	Autocorrelation
Aggregate	↑	↓	I	≠
Audio	↑	≡	≡	≡
GCCS-M (TX)	↑	↑	↑	≠
GCCS-M (RX)	↑	≡	≡	≠
HTTP Browser	↑	≡	≡	≡
HTTP Server	↑	↓	↑	≠

↑ - Increase, ↓ - Decrease, I - Inconclusive

≡ - No Significant Change, ≠ - Changes Observed

Table 5.4: Results of the Packet Lengths at the source

link, while Table 5.4 summarizes the effects on the size of the packets captured on the Ethernets close to the sources.

The changes that occur on the mean and variance of the packet lengths are not as visible as those declared for the interarrival times. When a change is reported in the tables it mostly implies that there is only a slight change and this change usually occurs at high error rates, see Appendix C. In the case of the packets captured on the link, the increase and decrease of the packets sizes can be caused by the retransmissions recorded at the layer two level. For instance, the GCCS-M transmitter traces and the HTTP server traces of the link are mostly composed of large packets carrying the data from the applications at the layer four and also of small packets consisting of the packet exchange of the layer two. When more of the larger packets are being corrupted by the errors than this requires retransmissions of large packets and in turn the mean of the packet lengths increase. On the other hand, in the case of the HTTP Browser, the packets captured are a combination of medium size packets of the layer four's request for web pages from the servers as well as smaller packets to

acknowledge the receipt of the packets from the server and also the layer two packets. In this case, if most of the packets being corrupted are the acknowledgments, then this causes the mean of the packet lengths to decrease.

As for the sizes of the packets being captured on the Ethernets, they should not be affected by an increase of the error rate, since the retransmissions are not visible to the sources. However in some cases a small change at the highest rate of error can be observed in the mean and variance. As explained in the previous section, the high error rates cause large delays at the source level. Because of these large delays it is not possible to send the same amount of data in a fixed amount of time as in a link with next to zero errors. For example, over 90 web pages can be downloaded in a web session of 30 minutes with a link containing no errors while only around 20 pages can be downloaded in the same amount of time using a link with a 5×10^{-5} error rate. Since the amount of packets being transmitted is not the same then this can explain the small variations in the measurements.

The comparison graphs of some of the autocorrelation functions for the sizes of packets are shown in Figures 5.5 to 5.8. As in the case of the interarrival times, these graphs imply that there can be some variations in the autocorrelation function of the packet lengths when errors are introduced into the link. First consider the graphs for the packets captured on the link, see Figures 5.5 and 5.6. In Table 5.3, there is a change reported for the aggregate traces, the GCCS-M traces as well as the HTML traces. In the case of the aggregate trace, the changes were more apparent in the traffic going from the Green LAN to the Yellow LAN which is not shown in this work. Again there was some periodicity that appeared to disappear as the error rate increase. This trend is also found for the traces of the GCCS-M receiver which are also not shown in this work. However, the graph for the autocorrelation function of the GCCS-M transmitter is shown here. These autocorrelation function are different than the others presented since, it has the shape of a wave. There is a higher

Figure 5.5: Autocorrelation for Packet Lengths of Aggregate and Audio Link Traces

Figure 5.6: Autocorrelation for Packet Lengths of GCCS-M and HTML Link Traces

Figure 5.7: Autocorrelation for Packet Lengths of Aggregate and Audio Source Traces

Figure 5.8: Autocorrelation for Packet Lengths of GCCS-M and HTML Source Traces

correlation found at lags surrounding the lags that are a multiple of 60. This can be related to the fact that the REPEAT software was sending data to be transmitted to the GCCS-M transmitter at approximately 60 second intervals. As the error rate increases the data takes longer to be transmitted and thus the wave like shape of the autocorrelation tends to be more stretched out to almost non existent.

On the other hand, when viewing the autocorrelation graphs for the packets captured on the Ethernets, see Figures 5.7 and 5.8, there does not seem to be any significant changes. However, there are a few variations found in some of the other graphs which are not shown here. In those cases the autocorrelation function appeared to go towards a zero autocorrelation slightly faster as the error rates increase. This effect is observed at autocorrelation lags that are less than 10.

5.3 Data Rate

The last component of the traffic that is analyzed in this thesis is the rate at which the data is being transmitted. The number of bytes transmitted per seconds defines the data rate. As in the previous cases, the detailed measurements for the mean and variance of the data rates are specified in Tables C.9 to C.12 of Appendix C. Table 5.5 shows the observations on the data rates of the narrowband link which are due to an increase in the error rate. As for the transmission rates of the sources, the effects caused by increasing the error rate is reported in Table 5.6.

Since the rates are expressed as bytes per seconds they consist of a combination of arrival time and packet lengths. This implies that because time is involved, the delays experienced in Section 5.1 when the errors were augmented are still influencing the measurements. The larger the delay, the less packets are received in a one second interval. This implies that as the errors increase, the data rates decrease. This is observed in Table 5.6 for the rates observed at the source. On the other hand, the measurements for the rates observed on the link are affected by not only the delays

Data Rates on Link				
	Error Rate	Mean	Variance	Autocorrelation
Aggregate	↑	↓	I	≠
Audio	↑	↓	↑	≠
GCCS-M	↑	↑	↑	≠
HTTP	↑	↓	↓	≠

↑ - Increase, ↓ - Decrease, I - Inconclusive

≡ - No Significant Change, ≠ - Changes Observed

Table 5.5: Results of the Data Rates on the Link

Data Rates at Source				
	Error Rate	Mean	Variance	Autocorrelation
Aggregate	↑	↓	↓	≠
Audio	↑	↓	I	≡
GCCS-M	↑	≡	≡	≠
HTTP	↑	↓	↓	≠

↑ - Increase, ↓ - Decrease, I - Inconclusive

≡ - No Significant Change, ≠ - Changes Observed

Table 5.6: Results of the Data Rates at the source

but also by the retransmission occurring at the layer two, as in Section 5.2. This combination of bytes and seconds entails that the effects caused by an increase in the error rates on the data rates are variable. In some cases when the errors are augmented the data rates increase while in other cases the data rates decreases, see Table 5.5.

The comparison graphs obtained from the autocorrelation functions of the data flows for the different rates of error are shown in Figures 5.9 to 5.12. The variations observed for the autocorrelations of the data rates are not as evident as those for the autocorrelations of the interarrival times and the packet lengths. In the case of the data rates observed on the narrowband link, the autocorrelation for audio decreases to zero slightly faster as the error rate increases, see Figure 5.9. Also in this figure, there is the comparison for the autocorrelations of the aggregate data rates. There are no significant changes present in this graph however there are some variations present in the aggregate data rates going from the Green LAN to the Yellow LAN, not shown here. There is negative autocorrelation found at the lags that are less than 10. These negative autocorrelations increase as the errors increase. The same behavior is also observed in the comparison graph for the HTML data rates, see Figure 5.10. In the case of GCCS-M, there is again high autocorrelations appearing at lags which are a multiple of approximately 60. The high autocorrelations tend to decrease as the rate of error increases. Eventually these high autocorrelations decrease to a zero autocorrelation.

Similar behaviors are occurring for the autocorrelation functions of the transmission rates observed at the sources, see Figure 5.12. The high autocorrelations of the GCCS-M data rates also tend to decrease as the error rates increase. In the case of the GCCS-M receiver, which is not shown here, the high autocorrelations decrease to a zero value. Furthermore, as in the case of the data rates of the link, the autocorrelations of the html data rates also show negative values that increase as the errors are augmented.

Legend: Yellow $\rightarrow$ 0, Green $\rightarrow 5 \times 10^{-6}$, Magenta $\rightarrow 1 \times 10^{-5}$,
Blue $\rightarrow 5 \times 10^{-5}$, Red $\rightarrow 1 \times 10^{-4}$

Figure 5.9: Autocorrelation for Data Rates of Aggregate and Audio Link Traces

Legend: Yellow $\rightarrow 0$, Green $\rightarrow 5 \times 10^{-6}$, Magenta $\rightarrow 1 \times 10^{-5}$,
 Blue $\rightarrow 5 \times 10^{-5}$, Red $\rightarrow 1 \times 10^{-4}$

Figure 5.10: Autocorrelation for Data Rates of GCCS-M and HTML Link Traces

Legend: Yellow $\rightarrow 0$, Green $\rightarrow 5 \times 10^{-6}$, Magenta $\rightarrow 1 \times 10^{-5}$,
 Blue $\rightarrow 5 \times 10^{-5}$, Red $\rightarrow 1 \times 10^{-4}$

Figure 5.11: Autocorrelation for Data Rates of Aggregate and Audio Source Traces

Figure 5.12: Autocorrelation for Data Rates of GCCS-M and HTML Source Traces

Figure 5.11, reveals that there is a slight increase in the autocorrelation at small lags for the aggregate data rates but that no significant changes can be observed in the case of audio.

5.4 Distributions of the Data Rates

The aim of Trial 4 is to study the statistical distribution of the data rates for the four different types of data. The original goal was to study the distributions for all the measurements including the interarrival times and the packet lengths. However the histograms obtained from analyzing their data were too irregular. In this case, only the data rates are fitted for their distributions which is reasonable since they take into account both time and data size. This can be viewed as if the data rate somehow incorporate the interarrival times and the packet lengths.

As described in Section 3.2, the BestFit software from the Palisade Corporation is used to fit different distributions to the collected data. BestFit uses the method of maximum likelihood to estimate the parameters of the distribution from the data. The reference manual states that these parameters, along with other values such as the number of class intervals in the histograms and the lower and upper limits of the plots, are estimated in a manner that provides the best fit for the specified distribution [22]. The selection of the number and width of the class intervals is an important step in the distribution fitting process. This selection can impact the results on whether a particular distribution is going to be rejected or accepted as a fit for the data. It also affects the appearance of the histogram, and thus the visual evaluation of the fit. Under these circumstances, it is decided to rely on BestFit and not to try to second-guess its selection of parameters.

Table 5.7 shows a summary of the best distributions for the measurements. The distributions are selected among the many satisfied fits based on the following criteria:

Distributions of the Data Rates for Aggregate Data		
On Link		
from Yellow to Green	Beta General	
from Green to Yellow	Beta General	
At Source		
on Yellow LAN	Beta General	
on Green LAN	Extreme Value	
Distributions of the Data Rates for Audio Data		
On Link		
from Yellow to Green	Beta General	
from Green to Yellow	Beta General	
At Source		
on Yellow LAN	Beta General	
on Green LAN	Beta General	
Distributions of the Data Rates for GCCS-M Data		
On Link		
from Yellow to Green	Extreme Value	
from Green to Yellow	Extreme Value	
At Source		
on Yellow LAN	Extreme Value	
on Green LAN	Extreme Value	
Distributions of the Data Rates for HTML Data		
On Link		
from Yellow to Green	Inverse Gaussian	
from Green to Yellow	Beta General	
At Source		
on Yellow LAN	Inverse Gaussian	
on Green LAN	Extreme Value	

Table 5.7: Distributions of the Data Rates

- The distribution is ranked among the highest using the chi-squared statistics of the data.
- The distribution is ranked among the highest using the Kolmogorov-Smirnov statistics of the data.
- The distributions visual fit to the data histogram is satisfactory.

It is sometimes difficult to select the best distribution for the data since the selection is based on two different ranking schemes. In some cases, a distribution that is ranked first using the chi-squared statistics can be ranked last when ranked using the Kolmogorov-Smirnov statistics. Consequently, the distribution is chosen using the mean of the two ranking values. By selecting the smallest mean rank, the distribution ends up being well ranked using both the chi-squared and the Kolmogorov-Smirnov statistics. When more than one distribution have the same mean rank, then the selection is made via a visual inspection. Figures 5.13 to 5.16 show some of the comparison graph for the data histograms and their selected distributions. The number of categories in the histograms of these figures is selected automatically by BestFit, and the parameters of the distributions are calculated from the data using the maximum likelihood methods. The sampled data which is presented in all these histograms is the data which was captured when there was no error rate added to the link.

The distributions that are reported in Table 5.7 are well ranked for all the data captured regardless of the error rates. This implies that when the errors on a narrow-band link are augmented the data rates distribution type is not affected. However, in Section 5.3, it is reported that the statistical measurements, the mean and variance, of the data rates show some changes due to the increase in error rate. Consequently, the type of distribution does not vary as the errors increase but the value of its parameters do. The distributions from Table 5.7 are defined as follows:

(a) On link from the Yellow LAN to the Green LAN

(b) Close to the source on Green LAN

Figure 5.13: Distribution Fit of the Data Rates for Aggregate Traces

(a) On link from the Yellow LAN to the Green LAN

(b) Close to the source on Green LAN

Figure 5.14: Distribution Fit of the Data Rates for Audio Traces

(a) On link from the Yellow LAN to the Green LAN

(b) Close to the source on Green LAN

Figure 5.15: Distribution Fit of the Data Rates for GCCS-M Traces

(a) On link from the Yellow LAN to the Green LAN

(b) Close to the source on Green LAN

Figure 5.16: Distribution Fit of the Data Rates for HTML Traces

Beta General Distribution

This distribution does not have a closed form but its probability density function is defined as

$$f(x) = \frac{(x - \min)^{\alpha_1 - 1} (\max - x)^{\alpha_2 - 1}}{(\max - \min)^{\alpha_1 + \alpha_2 - 1} B(\alpha_1, \alpha_2)} \quad (5.1)$$

$$B(\alpha_1, \alpha_2) = \int_0^1 t^{\alpha_1 - 1} (1 - t)^{\alpha_2 - 1} dt \quad (5.2)$$

where $\alpha_1, \alpha_2 > 0$ and $\max > \min$. The mean is $\min + (\max - \min) * \frac{\alpha_1}{\alpha_1 + \alpha_2}$ and the variance is $(\max - \min)^2 * \frac{\alpha_1 \alpha_2}{(\alpha_1 + \alpha_2)^2 (\alpha_1 + \alpha_2 + 1)}$.

Extreme Value Distribution

The extreme value distribution is defined as

$$F(x) = e^{-e^{-\frac{x-a}{b}}} \quad (5.3)$$

where $b > 0$ and a is the mode. The mean is $a - b\Gamma'(1) \approx a + 0.57721b$ and the variance is $\frac{b^2 \pi^2}{6}$.

Inverse Gaussian Distribution

This distribution again has no closed form. The probability density function is defined as

$$f(x) = \sqrt{\frac{\lambda}{2\pi x^3}} e^{-\frac{\lambda(x-\mu)^2}{2\mu^2 x}} \quad (5.4)$$

where $\mu, \lambda > 0$. The mean is μ and the variance is $\frac{\mu^3}{\lambda}$.

The above distributions describe the data rates obtained from the test network, which represents the traffic of a narrowband network. Note that these distributions are not the same as those described in the traffic models of Chapter 2. This leads us to believe that the traffic models for broadband and high speed networks might not be appropriate to describe the traffic of narrowband networks.

Chapter 6

Conclusion

Traffic modelling holds a key role in both past and current research on data networks. This is quite evident from the quantity of literature and study dedicated to it, especially in the area of fractals. The vast majority of this research is however aimed at broadband and high speed networks. In some fields, the Armed Forces for example, there exists a need for research to be more developed in the area of modelling narrowband networks. This thesis is a first step in that direction.

Narrowband links of a network can vary from satellite links, radio frequency links or terrestrial links (e.g. ISDN line). These narrowband links involve error rates which are different from one link to another. The error rate can vary from next to no error at all to as much as a 10^{-4} rate of error or worse. This thesis consists of studying the effects that increasing the error rate of a narrowband link has on the traffic of the network. In order to conduct this study a test network is set-up consisting of two Ethernets connected through a narrowband link in which errors can be injected. On this test network, many trials are conducted using different error rates and for different types of data. The collected data is used to evaluate the changes to the traffic measurements due to the increase in errors.

Some significant findings are realized from the measurements of both the Ethernet

traffic (i.e. the traffic at the source) and the link traffic of the test network. Key findings include the changes that occurred to the first- and second-order statistics (mean, variance, and autocorrelation function) of the measurements. They are summarized as follows:

- For the interarrival time measurements, the statistics vary the most in the case of the Ethernet traffic. These changes are due to the delays caused by the errors injected into the link. In the case of the link traffic, the variations are most apparent for the measurements obtained when the error rates are high. These changes are also related to the delays observed at the source.
- For the packet length measurements, the statistics vary the most in the case of the link traffic. These changes are due to the retransmissions of the corrupted packets. The retransmissions did not have much of an effect on the Ethernet traffic because they only occur at layer two and therefore are transparent to the sources.
- For the data rate measurements, the statistics vary equally for both the Ethernet traffic as well as the link traffic. Because the measurements are composed of time and packet sizes, the changes in the statistics resulted from the combination of the delays and the retransmissions.

The most interesting findings are associated with the statistical distribution of the data rate measurements. The distribution fit performed on the collected data shows that the distribution of the measurements is not affected by the change in error rate. However, because of the change occurring to the mean and variance of the measurements, the values of the distributions parameters are variable. It is also shown that the distributions are different from one type of data to the next. The three reported distributions are Beta, Extreme Value, and Inverse Gaussian. These distributions are unlike those reported for the broadband network models of Chapter 2 which leads us to believe that different network models are required for the narrowband networks.

Future Work

This thesis shows by using a test network that different traffic models are needed for narrowband networks than the ones currently used for broadband and high speed networks. In this event, the next logical step to pursue is the verification of the results obtained by collecting data from a real narrowband network. However, before being able to model narrowband traffic, other aspects of the traffic should be evaluated. In this thesis, the effects of independent random errors on the networks traffic was investigated. The next element that should be studied is how the traffic behaves to bursts of errors as opposed to independent errors. Furthermore, how are network delays affecting the traffic? These should be examined before attempting to build a narrowband traffic model.

Bibliography

- [1] A. Adas, "Traffic Models in Broadband Networks," *IEEE Communication Magazine*, pp. 82-89, July 1997
- [2] Adtech, "Operating Manual, SX/12 Data Link Channel Simulator," *Adtech Inc.*, Honolulu, HI, 1999
- [3] J. S. Bendat and A. G. Piersol, "Random Data: Analysis and Measurement Procedures," *Wiley*, New York, 1986
- [4] R. Burk, M. Bligh, T. Lee, et al., "TCP/IP Blueprints," *Sams Publishing*, Indianapolis, 1997
- [5] N. Desaulniers-Soucy and A. Iuoras, "Traffic Modeling with Universal Multifractals" *Intended for the General Conference, session on "Modeling and Simulation of Communication Systems"*
- [6] A. Feldmann, A. C. Gilbert, W. Willinger and T. G. Kurtz, "The Changing Nature of Network Traffic: Scaling Phenomena," *Computer Communication Review*, **28**(2), April 1998
- [7] A. Feldmann, A. C. Gilbert and W. Willinger, "Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic." *Proceedings of the ACM/Sigcomm98, Vancouver, BC*, pp. 25- 38, 1998
- [8] A. Feldmann, "Characteristics of TCP Connection Arrivals," *Preprint* 1998

- [9] L. J. Forys, A. Erramilli and J. L. Wang, "New Traffic Analysis and Engineering Methods for Emerging Technologies," *In Proc. IEEE Globecom*, pp. 848-854, 1995
- [10] V. S. Frost and B. Melamed, "Traffic Modeling for Telecommunications Networks," *IEEE Communication Magazine*, pp. 70-81, March 1994
- [11] J. Gao and I. Rubin, "Multifractal Analysis and Modeling of Long-Range-Dependent Traffic," *IEEE International Conference on Communication*, Vancouver, BC, June 1999
- [12] A. C. Gilbert, W. Willinger and A. Feldmann, "Scaling analysis of conservative cascades, with applications to network traffic," *IEEE Transactions on Information Theory*, **45**(3), April 1999
- [13] A. C. Gilbert, W. Willinger and A. Feldmann, "Visualizing multifractal scaling behavior: A simple coloring heuristic," *Proceedings of the 32nd Asilomar conference on signals, systems and computers*, Pacific Cove, CA, Nov 1998
- [14] GN Nettest, "WinPharaoh LAN/WAN Users Guide, Revision 3.X," *GN Nettest (Boston) Inc.*, Hopkinton, MA, October 1999
- [15] V. Jacobson, C. Leres and S. McCanne, "The TCPdump Manual Page," *Lawrence Berkeley Laboratory*, Berkeley, CA, June 1998
- [16] L. A. Kulkarni and S. Li, "Traffic Modeling: Matching the Power Spectrum and Distribution," *In Proc. IEEE Globecom*, pp. 1701-1706, 1995
- [17] W. E. Leland, M. S. Taqqu, W. Willinger and D.V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Transactions on Networking*, **2**, pp. 1-15, 1994
- [18] A. Leon-Garcia, "Probability and Random Processes for Electrical Engineering," *Addison Wesley*, Don Mills, Ontario, 1994

- [19] N. Likhanov, B. Tsybakov and N. Georganas, "Analysis of an ATM Buffer with Self-similar (Fractal) Input Traffic," *In Proc. IEEE INFOCOM '95*, pp. 746-752, 1995
- [20] B. Melamed and J.R. Hill, "A Survey of TES Modeling Applications," *Technical Report, Rutgers Center for Operational Research*, New Jersey, 1995
- [21] R. Nelson, "Probability, Stochastic Processes and Queuing Theory," *Springer-Verlag*, New York, 1995
- [22] Palisade Corporation, "Guide to Using BestFit, Distribution Fitting for Windows (Version 4)," *Palisade Corporation*, Newfield, NY, April 2000
- [23] K. Park and W. Willinger, "Self-Similar Network Traffic: An Overview," *In Self-similar Network Traffic and Performance Evaluation*, Wiley, April 2000
- [24] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transaction on Networking*, **3**(3), pp. 226-244, June 1995
- [25] R. H. Riedi, M. S. Crouse, V. J. Ribeiro and R. G. Baraniuk, "A Multifractal Wavelet Model with Application to Network Traffic," *IEEE Transactions on Information Theory*, **45**(3), pp. 992-1019, April 1999
- [26] R. H. Riedi and J. L. Véhel, "Multifractal Properties of TCP Traffic: a numerical study (Extended Version)," *Technical Report No 3129, INRIA Rocquencourt, France*, March 1997, Available at www.dsp.rice.edu
- [27] R. H. Riedi and W. Willinger, "Toward an Improved Understanding of Network Traffic Dynamics," *In Self-similar Network Traffic and Performance Evaluation*, Wiley, April 2000
- [28] M. S. Taqqu, V. Teverovsky and W. Willinger, "Is network traffic self-similar or multifractal?" *Fractals*, **5**(1), pp. 63-73, 1997

- [29] B. Tsybakov and N. D. Georganas, "Self-similar Processes in Communications Networks," *IEEE Transactions on Information Theory*, **44**(5), pp. 1713-1725, September 1998
- [30] W. Willinger, V. Paxson and M. S. Taqqu, "Self-similarity and Heavy Tails: Structural Modeling of Network Traffic," *In a Practical Guide to Heavy Tails: Statistical Techniques and Applications*, Birkhäuser, 1998
- [31] W. Willinger, M. S. Taqqu, R. Sherman and D. V. Wilson, "Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level," *IEEE/ACM Transactions on Networking*, **5**, pp. 71-86, 1997

Appendix A

Network Architecture

The network is responsible for providing communication between nodes (i.e. computers) in a communication system. This system allows any network node to transmit information to another node connected to the communication network. Because communication systems are complex, network designers have developed a general network architecture that guide the design and implementation of networks. The network architecture consists of dividing the system into layer structures. The general idea of layering is to start with the services offered by the underlying hardware, and then add a sequence of layers, each providing a higher level of services. The services provided at the high layers are implemented in terms of the services provided by the low layers. Communication between layers is governed by protocols. Protocols include, but are not limited to, formats and order of the information exchange, and any actions to be taken on the information transmission and receipt. Following is a brief description of the layers in the more traditional seven layer Open System Interconnection (OSI) architecture, as well as the more popular four layer Internet (IP) architecture. Figure A.1 illustrates the two different architectures.

- **OSI architecture:** The OSI architecture, illustrated in Figure A.1(a), defines a partitioning of network functionality into seven layers, where one or more protocols implement the functionality assigned to a given layer. In this sense, the diagram given in Figure A.1(a) is not a protocol graph, but rather a *ref-*

Figure A.1: Protocol Architecture

erence model for a protocol graph. The functions of the seven layers in the OSI model are described in Table A.1. This seven layer model provides a good basis for understanding the functions and features that need to be provided in the protocols that implement the model. However, it would be more efficient to have fewer layers. This was the approach taken by the designers of the IP architecture.

- **IP architecture:** The protocol model for the Internet architecture, also known as TCP/IP architecture, is conceptually simpler than that of the OSI model. It consists of only four layers, as shown in Figure A.1(b). A key feature of the TCP/IP architecture, as illustrated by its figure, is that it does not imply strict layering. The application is free to bypass the defined transport layers (TCP/UDP) and to directly use IP or one of the underlying networks. The functions of the four layers in the TCP/IP architecture are shown in Table A.2.

Layer	Function
Physical	To define a communication channel in order to transmit binary numbers to other systems connected to this channel.
Link	To define how to transmit reliably a packet, or frame, of information between two stations on the same physical network.
Network	To define how a single packet is transmitted between two stations on different physical networks.
Transport	To provide reliable end-to-end communications between two hosts using the functions provided by the network layer.
Session	To provide a name space that is used to tie together the potentially different transport streams that are part of a single application.
Presentation	To provide translation between different data representation.
Application	This layer comprises all the functions which the user applications directly access, e.g. File Transfer Protocol (FTP).

Table A.1: The functions of each OSI layers

Layer	Function
Network	This corresponds to the physical and data link layer of the OSI model. It consists of a wide variety of network protocols (e.g. Ethernet or Fiber Distributed Data Interface (FDDI) protocols), which are implemented by a combination of network adaptors and network device drivers.
Internet(IP)	This corresponds to the network layer of the OSI model. It supports the interconnection of multiple networking technologies into a single, logical internetwork.
Transport (TCP/UDP)	This layer corresponds to the transport layer of the OSI model. It provides alternative logical channels to application programs. TCP provides a reliable byte stream channel while UDP provides an unreliable datagram delivery channel.
Application	This corresponds to the presentation and application layers of the OSI model. There are a large number of protocols at this layer which enable the interoperation of popular applications. Some of these protocols are the File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), and so on.

Table A.2: The functions of each TCP/IP layers

In summary, TCP/IP is a popular and widely adopted implementation of the OSI model. The functions described in the OSI model have largely been implemented in TCP/IP and are now widely utilized both in private corporate intranetworks as well as on the Internet itself. For more information on the layer type models and the details of the TCP/IP protocols the reader is referred to TCP/IP Blueprints by R. Burk et al. [4].

Appendix B

TCPdump

In this thesis, TCPdump is used with the following options. For more details and more options on TCPdump the reader is referred to its manual page [15].

`tcpdump [-e n tt][-r file][-w file] expression`

<code>-e</code>	Print the link-level header on each dump line.
<code>-n</code>	Don't convert addresses (i.e. host addresses, port numbers, etc.) to names.
<code>-tt</code>	Print an unformatted timestamp on each dump line.
<code>-r <i>file</i></code>	Read packets from <i>file</i> (which was created with the -w option). Standard input is used if <i>file</i> is "-".
<code>-w <i>file</i></code>	Write the raw packets to <i>file</i> rather than parsing and printing them out. They can later be printed with the -r option. Standard output is used if <i>file</i> is "-".

The *expression* consists of one or more primitives. Primitives usually consists of an id (name or number) preceded by one or more qualifiers. The following are the primitives that are of interest in this work:

<code>dst host <i>host</i></code>	True if IP destination field of the packet is <i>host</i> , which may be either an address or a name.
<code>src host <i>host</i></code>	True if IP source field of the packet is <i>host</i> .
<code>host <i>host</i></code>	True if either the IP source or destination of the packet is <i>host</i>
<code>net <i>net</i></code>	True if either the IP source or destination address of the packet has a network number of <i>net</i> , which may be either an address or a name. This primitive can also be preceded by <code>src</code> or <code>dst</code> to specify an IP source or destination address as was the case with the <code>host</code> qualifier.

Primitives may be combined using parenthesized group of primitives and operators (parentheses are special to the shell and must be escaped). Note that negation has highest precedence. Alternation and concatenation have equal precedence and associate left to right.

Negation	'!' or 'not'
Concatenation	'&&' or 'and'
Alternation	' ' or 'or'

Examples

To print all packets arriving at or departing from 'jimbo' to a file 'filename':

```
tcpdump -w filename host jimbo, or
tcpdump -w filename host 192.168.3.50
```

To print all traffic of local hosts arriving from LAN 192.168.3 to a file 'filename':

```
tcpdump -w filename src net 192.168.3
```

To print all traffic between 'jimbo' and 'schipol' as well as between 'jimbo' and 'gccs2' to a file 'filename':

```
tcpdump -w filename host jimbo and \( schipol or gccs2 \), or
tcpdump -w filename host 192.168.3.50 && \( 192.168.2.50 || 192.168.2.5 \)
```

Output Format

The files created by TCPdump as in the above examples can only be viewed by using TCPdump with the `-r` option. The format of the output is dependent on the options included with the command (e.g. `-tt`, `-e`, `-n`). In this thesis the TCPdump command is utilized as follows:

```
tcpdump -tt -e -r 'filename'
```

This provided a format where the required information needed for the analysis of traffic traces can be easily filtered out. By default, all the output lines are preceded by a timestamp which is the current clock time in the form `hh:mm:ss:μs`. The timestamp reflects the time the kernel first saw the packet. Using the `-tt` option, tcpdump converts the timestamp to an unformatted timestamp of the form `ss.μs`. This format is easier to manipulate for the data analysis, since it is one number as opposed to multiple numbers separated by colons. The `-e` option is then provided to print out the link level header. On Ethernets, which is the case in this thesis, the source and destination addresses, protocol, and packet length are printed following the timestamp on the output line. This is an example of such an output:

```
955727001.306333 8:0:2:2d:6b:d5 8:0:9:f2:9:e9 0800 60: 192.168.2.5.2067 > 192.168.3.5.2020: . ack 1 win 32768
955727001.497232 8:0:2:2d:6b:d5 8:0:9:f2:9:e9 0800 1514: 192.168.2.5.2067 > 192.168.3.5.2020: . 1:1461(1460) ack 1
win 32768
955727001.511739 8:0:9:f2:9:e9 8:0:2:2d:6b:d5 0800 60: 192.168.3.5.2020 > 192.168.2.5.2067: . ack 1461 win 32768
955727001.603641 8:0:2:2d:6b:d5 8:0:9:f2:9:e9 0800 603: 192.168.2.5.2067 > 192.168.3.5.2020: FP 1461:2010(549) ack
1 win 32768
955727001.603846 8:0:9:f2:9:e9 8:0:2:2d:6b:d5 0800 60: 192.168.3.5.2020 > 192.168.2.5.2067: . ack 2011 win 32219
```

Where the first five elements are the timestamp and link level header as described above. The rest of the output line is the general format of a TCP protocol line, which is of the form:

```
src > dest flags data-seqno ack window
```

Src and dst are the source and destination IP addresses and ports. Flags are some combination of S(SYN), F(FIN), P(PUSH) or a single '.' (no flags). Data-seqno describes the portion of sequence space covered by the data in the packet. The notation is 'first:last(nbytes)' which means 'sequence numbers first up to but not including last which is nbytes bytes of user data'. Ack is the sequence number of the next data expected and window is the number of bytes of receive buffer space available.

Appendix C

Detailed Measurements

Interarrival Times for Aggregate Data			
	Error Rate	Mean	Variance
On Link from Yellow to Green	0	0.0145	0.0018
	$5 \times e^{-6}$	0.0146	0.0027
	$1 \times e^{-5}$	0.0155	0.0051
	$5 \times e^{-5}$	0.0401	0.0471
On Link from Green to Yellow	0	0.0155	0.0017
	$5 \times e^{-6}$	0.0177	0.0031
	$1 \times e^{-5}$	0.0209	0.0051
	$5 \times e^{-5}$	0.0448	0.0480
On Yellow LAN	0	0.0542	0.0224
	$5 \times e^{-6}$	0.0581	0.0236
	$1 \times e^{-5}$	0.0635	0.0268
	$5 \times e^{-5}$	0.1167	0.1692
On Green LAN	0	0.0626	0.0370
	$5 \times e^{-6}$	0.0652	0.0399
	$1 \times e^{-5}$	0.0758	0.0557
	$5 \times e^{-5}$	0.1533	0.2485

Table C.1: Statistics of Interarrival times for Aggregate Data

Interarrival Times for Audio Data					
	Error Rate	Version 1		Version 2	
		Mean	Variance	Mean	Variance
On Link from Yellow to Green	0	0.0132	0.00095	0.0130	0.0012
	$5 \times e^{-6}$	0.0153	0.0013	0.0143	0.0017
	$1 \times e^{-5}$	0.0157	0.0018	0.0142	0.0017
	$5 \times e^{-5}$	0.0187	0.0111	0.0190	0.0120
	$1 \times e^{-4}$	0.0374	0.0480	N/A	N/A
On Link from Green to Yellow	0	0.0162	0.0012	0.0169	0.0013
	$5 \times e^{-6}$	0.0150	0.0015	0.0165	0.0015
	$1 \times e^{-5}$	0.0148	0.0014	0.0168	0.0017
	$5 \times e^{-5}$	0.0191	0.0116	0.0190	0.0102
	$1 \times e^{-4}$	0.0342	0.0468	N/A	N/A
On Yellow LAN	0	0.0695	0.1626	0.0751	0.1849
	$5 \times e^{-6}$	0.0623	0.1381	0.0713	0.1622
	$1 \times e^{-5}$	0.0620	0.1202	0.0733	0.1668
	$5 \times e^{-5}$	0.0762	0.1968	0.0820	0.2095
	$1 \times e^{-4}$	0.1202	0.4159	N/A	N/A
On Green LAN	0	0.0541	0.0822	0.0527	0.0767
	$5 \times e^{-6}$	0.0632	0.1267	0.0578	0.0990
	$1 \times e^{-5}$	0.0657	0.1433	0.0583	0.1012
	$5 \times e^{-5}$	0.0773	0.2020	0.0727	0.1725
	$1 \times e^{-4}$	0.1200	0.4052	N/A	N/A

Table C.2: Statistics of Interarrival times for Audio Data

Interarrival Times for GCCS-M Data					
	Error Rate	Version 1		Version 2	
		Mean	Variance	Mean	Variance
On Link from Yellow to Green	0	0.3205	0.5135	0.1547	0.1185
	$5 \times e^{-6}$	0.4425	0.9702	0.1775	0.1524
	$1 \times e^{-5}$	0.3700	0.8103	0.4397	1.0665
	$5 \times e^{-5}$	0.2653	0.3752	0.4027	0.9789
	$1 \times e^{-4}$	0.2394	0.3846	0.1981	0.2996
On Link from Green to Yellow	0	0.2582	0.3154	0.4094	1.0588
	$5 \times e^{-6}$	0.1318	0.1136	0.3805	0.9558
	$1 \times e^{-5}$	0.2003	0.3408	0.1158	0.1428
	$5 \times e^{-5}$	0.2540	0.3819	0.1026	0.1353
	$1 \times e^{-4}$	0.1938	0.2490	0.2094	0.3251
On Yellow LAN	0	1.1814	60	1.0794	53
	$5 \times e^{-6}$	1.1787	59	1.0676	50
	$1 \times e^{-5}$	1.1752	55	1.0655	49
	$5 \times e^{-5}$	1.0951	48	1.0102	42
	$1 \times e^{-4}$	1.9796	25	0.8905	18
On Green LAN	0	1.3613	72	1.2365	62
	$5 \times e^{-6}$	1.3591	71	1.2294	61
	$1 \times e^{-5}$	1.3521	68	1.2272	60
	$5 \times e^{-5}$	1.2472	59	1.1468	51
	$1 \times e^{-4}$	1.1210	39	1.0339	33

Table C.3: Statistics of Interarrival times for GCCS-M Data

Interarrival Times for HTML Data					
	Error Rate	Version 1		Version 2	
		Mean	Variance	Mean	Variance
On Link from Yellow to Green	0	0.0525	0.0215	0.0514	0.0209
	$5 \times e^{-6}$	0.0510	0.0211	0.0509	0.0214
	$1 \times e^{-5}$	0.0524	0.0266	0.0525	0.0260
	$5 \times e^{-5}$	0.1395	0.1872	0.1402	0.1936
On Link from Green to Yellow	0	0.0441	0.0183	0.0450	0.0257
	$5 \times e^{-6}$	0.0516	0.0224	0.0508	0.0237
	$1 \times e^{-5}$	0.0616	0.0274	0.0573	0.0232
	$5 \times e^{-5}$	0.1483	0.1724	0.1502	0.1746
On Yellow LAN	0	0.1674	0.0143	0.1620	0.0116
	$5 \times e^{-6}$	0.1888	0.0304	0.1831	0.0281
	$1 \times e^{-5}$	0.2296	0.0971	0.2135	0.0709
	$5 \times e^{-5}$	0.9554	3.6844	0.9886	3.5849
On Green LAN	0	0.1578	0.0577	0.1151	0.0513
	$5 \times e^{-6}$	0.1792	0.0797	0.1753	0.0709
	$1 \times e^{-5}$	0.2194	0.1428	0.2063	0.1164
	$5 \times e^{-5}$	0.9698	3.4802	0.9558	3.0497

Table C.4: Statistics of Interarrival times for HTML Data

Packet Lengths for Aggregate Data			
	Error Rate	Mean	Variance
On Link from Yellow to Green	0	53	6 439
	$5 \times e^{-6}$	52	6 753
	$1 \times e^{-5}$	54	6 730
	$5 \times e^{-5}$	60	13 947
On Link from Green to Yellow	0	181	184 330
	$5 \times e^{-6}$	188	185 220
	$1 \times e^{-5}$	198	201 440
	$5 \times e^{-5}$	194	195 740
On Yellow LAN	0	102	8 540
	$5 \times e^{-6}$	101	8 730
	$1 \times e^{-5}$	100	8 677
	$5 \times e^{-5}$	100	11 549
On Green LAN	0	389	313 580
	$5 \times e^{-6}$	358	290 340
	$1 \times e^{-5}$	354	286 850
	$5 \times e^{-5}$	217	157 310

Table C.5: Statistics of Packet Lengths for Aggregate Data

Packet Lengths for Audio Data					
	Error Rate	Version 1		Version 2	
		Mean	Variance	Mean	Variance
On Link from Yellow to Green	0	33	1 142	32	1 124
	$5 \times e^{-6}$	38	1 163	34	1 148
	$1 \times e^{-5}$	38	1 157	34	1 146
	$5 \times e^{-5}$	39	1 156	37	1 164
	$1 \times e^{-4}$	41	1 152	N/A	N/A
On Link from Green to Yellow	0	41	1 148	43	1 135
	$5 \times e^{-6}$	37	1 159	41	1 152
	$1 \times e^{-5}$	36	1 157	41	1 150
	$5 \times e^{-5}$	38	1 156	40	1 154
	$1 \times e^{-4}$	38	1 155	N/A	N/A
On Yellow LAN	0	78	17	78	19
	$5 \times e^{-6}$	78	15	78	18
	$1 \times e^{-5}$	78	15	78	19
	$5 \times e^{-5}$	78	16	78	19
	$1 \times e^{-4}$	79	43	N/A	N/A
On Green LAN	0	78	11	78	16
	$5 \times e^{-6}$	78	12	78	11
	$1 \times e^{-5}$	78	13	78	12
	$5 \times e^{-5}$	78	13	78	14
	$1 \times e^{-4}$	78	31	N/A	N/A

Table C.6: Statistics of Packet Lengths for Audio Data

Packet Lengths for GCCS-M Data					
	Error Rate	Version 1		Version 2	
		Mean	Variance	Mean	Variance
On Link from Yellow to Green	0	58	41 575	61	44 302
	$5 \times e^{-6}$	62	47 378	66	50 719
	$1 \times e^{-5}$	65	50 869	69	54 233
	$5 \times e^{-5}$	112	105 950	112	103 470
	$1 \times e^{-4}$	191	187 600	199	193 600
On Link from Green to Yellow	0	18	459	18	463
	$5 \times e^{-6}$	18	460	18	466
	$1 \times e^{-5}$	18	462	18	465
	$5 \times e^{-5}$	19	484	19	492
	$1 \times e^{-4}$	21	581	22	611
On Yellow LAN	0	278	67 191	279	67 216
	$5 \times e^{-6}$	280	70 019	281	69 959
	$1 \times e^{-5}$	281	70 975	283	71 568
	$5 \times e^{-5}$	298	95 263	294	89 724
	$1 \times e^{-4}$	312	121 490	317	125 210
On Green LAN	0	224	10 806	225	10 723
	$5 \times e^{-6}$	224	10 842	224	10 826
	$1 \times e^{-5}$	223	10 903	224	10 837
	$5 \times e^{-5}$	212	11 836	213	11 776
	$1 \times e^{-4}$	193	12 886	195	12 833

Table C.7: Statistics of Packet Lengths for GCCS-M Data

Packet Lengths for HTML Data					
	Error Rate	Version 1		Version 2	
		Mean	Variance	Mean	Variance
On Link from Yellow to Green	0	69	13 574	72	14 395
	$5 \times e^{-6}$	68	13 360	71	14 126
	$1 \times e^{-5}$	67	13 165	70	14 102
	$5 \times e^{-5}$	51	9 372	51	9 629
On Link from Green to Yellow	0	547	459 840	533	448 420
	$5 \times e^{-6}$	579	470 640	563	459 830
	$1 \times e^{-5}$	619	482 960	591	466 060
	$5 \times e^{-5}$	600	481 620	586	473 920
On Yellow LAN	0	154	19 191	161	19 872
	$5 \times e^{-6}$	153	19 093	160	19 740
	$1 \times e^{-5}$	152	19 037	160	19 797
	$5 \times e^{-5}$	143	17 635	145	18 449
On Green LAN	0	1 236	204 450	1 207	219 530
	$5 \times e^{-6}$	1 233	208 870	1 206	222 330
	$1 \times e^{-5}$	1 225	219 439	1 197	228 510
	$5 \times e^{-5}$	989	400 430	1 016	384 030

Table C.8: Statistics of Packet Lengths for HTML Data

Data Rates for Aggregate Data			
	Error Rate	Mean	Variance
On Link from Yellow to Green	0	1 743	1 635 483
	$5 \times e^{-6}$	1 606	1 862 835
	$1 \times e^{-5}$	1 455	1 819 846
	$5 \times e^{-5}$	704	1 302 161
On Link from Green to Yellow	0	6 119	7 803 062
	$5 \times e^{-6}$	5 868	8 071 589
	$1 \times e^{-5}$	5 453	8 913 001
	$5 \times e^{-5}$	2 281	7 393 907
On Yellow LAN	0	1 880	2 147 390
	$5 \times e^{-6}$	1 739	2 244 721
	$1 \times e^{-5}$	1 581	1 908 735
	$5 \times e^{-5}$	859	1 591 820
On Green LAN	0	6 208	23 734 562
	$5 \times e^{-6}$	5 489	20 034 168
	$1 \times e^{-5}$	4 671	18 061 903
	$5 \times e^{-5}$	1 412	5 545 563

Table C.9: Statistics of Data Rates for Aggregate Data

Data Rates for Audio Data					
	Error Rate	Version 1		Version 2	
		Mean	Variance	Mean	Variance
On Link from Yellow to Green	0	1 132	1 106 217	1 058	1 084 197
	$5 \times e^{-6}$	1 246	1 175 016	1 107	1 077 412
	$1 \times e^{-5}$	1 258	1 138 387	1 086	1 090 463
	$5 \times e^{-5}$	1 025	1 524 222	987	1 414 989
	$1 \times e^{-4}$	577	959 545	N/A	N/A
On Link from Green to Yellow	0	1 408	1 074 520	1 439	1 041 169
	$5 \times e^{-6}$	1 229	1 117 489	1 325	1 112 309
	$1 \times e^{-5}$	1 192	1 159 531	1 326	1 141 969
	$5 \times e^{-5}$	995	1 377 657	1 068	1 528 779
	$1 \times e^{-4}$	525	836 860	N/A	N/A
On Yellow LAN	0	1 128	1 421 779	1 044	1 392 056
	$5 \times e^{-6}$	1 257	1 469 473	1 100	1 368 283
	$1 \times e^{-5}$	1 263	1 418 556	1 070	1 343 140
	$5 \times e^{-5}$	1 028	1 386 268	956	1 277 235
	$1 \times e^{-4}$	624	1 077 192	N/A	N/A
On Green LAN	0	1 446	1 371 561	1 485	1 345 946
	$5 \times e^{-6}$	1 238	1 420 247	1 355	1 399 108
	$1 \times e^{-5}$	1 192	1 446 178	1 343	1 416 549
	$5 \times e^{-5}$	1 013	1 388 116	1 076	1 391 177
	$1 \times e^{-4}$	653	1 108 810	N/A	N/A

Table C.10: Statistics of Data Rates for Audio Data

Data Rates for GCCS-M Data					
	Error Rate	Version 1		Version 2	
		Mean	Variance	Mean	Variance
On Link from Yellow to Green	0	100	145 050	108	165 160
	$5 \times e^{-6}$	108	193 929	118	216 338
	$1 \times e^{-5}$	114	216 294	124	231 502
	$5 \times e^{-5}$	217	638 326	223	649 852
	$1 \times e^{-4}$	447	1 194 878	494	1 221 296
On Link from Green to Yellow	0	31	2 922	32	3 249
	$5 \times e^{-6}$	31	3 268	32	3 831
	$1 \times e^{-5}$	31	3 359	32	3 547
	$5 \times e^{-5}$	35	5 506	37	6 165
	$1 \times e^{-4}$	48	12 775	52	13 875
On Yellow LAN	0	236	912 153	258	1 004 778
	$5 \times e^{-6}$	238	922 876	263	1 022 703
	$1 \times e^{-5}$	239	904 983	265	1 035 630
	$5 \times e^{-5}$	272	1 033 464	291	1 064 674
	$1 \times e^{-4}$	319	922 684	355	1 014 709
On Green LAN	0	165	609 615	182	695 964
	$5 \times e^{-6}$	165	607 555	182	689 309
	$1 \times e^{-5}$	165	598 286	182	667 132
	$5 \times e^{-5}$	170	661 889	186	697 303
	$1 \times e^{-4}$	173	611 343	188	654 667

Table C.11: Statistics of Data Rates for GCCS-M Data

Data Rates for HTML Data					
	Error Rate	Version 1		Version 2	
		Mean	Variance	Mean	Variance
On Link from Yellow to Green	0	716	550 983	751	663 764
	$5 \times e^{-6}$	658	510 624	695	613 779
	$1 \times e^{-5}$	583	473 353	644	553 373
	$5 \times e^{-5}$	173	131 425	173	121 587
On Link from Green to Yellow	0	5 663	10 570 905	5 531	10 896 467
	$5 \times e^{-6}$	5 667	10 316 392	5 554	10 680 578
	$1 \times e^{-5}$	5 438	10 531 113	5 459	10 680 692
	$5 \times e^{-5}$	2 108	7 335 920	2 051	7 255 323
On Yellow LAN	0	711	560 168	760	667 142
	$5 \times e^{-6}$	652	501 325	684	597 866
	$1 \times e^{-5}$	555	414 142	616	513 771
	$5 \times e^{-5}$	133	93 625	133	87 371
On Green LAN	0	5 721	26 573 315	5 594	25 264 989
	$5 \times e^{-6}$	5 165	23 266 077	5 116	22 840 670
	$1 \times e^{-5}$	4 422	19 944 018	4 558	20 050 885
	$5 \times e^{-5}$	956	4 265 606	946	4 169 084

Table C.12: Statistics of Data Rates for HTML Data