

SIMULATION AND ANALYSIS OF LOSS IN IP NETWORKS

by

Velibor Markovski

B.Sc., University “Sts. Cyril and Methodius”, Macedonia, 1996

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in the School
of
Engineering Science

© Velibor Markovski 2000
SIMON FRASER UNIVERSITY
October 2000

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without the permission of the author.

APPROVAL

Name: Velibor Markovski
Degree: Master of Applied Science
Title of thesis: Simulation and analysis of loss in IP networks

Examining Committee: Prof. Mehrdad Saif, Committee-Chair, Chairman

Prof. Ljiljana Trajković, Senior Supervisor

Prof. Stephen Hardy, Supervisor
School of Engineering Science

Prof. Joseph G. Peters, Supervisor
School of Computing Science

Prof. Tsunehiko (Tiko) Kameda, External examiner
School of Computing Science

Date Approved: _____

Abstract

The primary goal of this research is to simulate and analyze the loss process in Internet Protocol (IP) networks. Obtaining insight into the loss patterns in a network is important for measuring the network performance during transfers of multimedia traffic.

In order to analyze the packet loss process, we collected specific information about when loss appears in the network, which packets are lost, whether the loss is clustered in bursts (episodes), how long these loss episodes are, and what the impact of network protocols on loss is. Of particular interest were consecutive packet losses because they contribute to the prolonged loss episodes, which adversely affect multimedia traffic. We expect that having better knowledge about the loss patterns may help design decisions for enabling or improving Quality of Service (QoS) support in packet networks.

We use a simulation-based approach for collection of loss data. We investigate the loss process during UDP and TCP transfers, focusing on loss episodes. We analyze consecutive packet losses for both per-flow and aggregate loss at the router buffers. Our analysis indicates that packet loss exhibits long-range dependence and preserves the self-similar property of the traffic patterns sent by video sources.

Acknowledgments

I would like to extend my special thanks to my senior supervisor, Dr. Ljiljana Trajković, for her encouragement and support throughout my period of research at Simon Fraser University. Thank you for your trust, valuable feedback, and inspiration for many of the ideas presented in this work.

I would also like to thank Dr. Stephen Hardy and Dr. Joseph Peters, for being on my supervisory committee, and Dr. Tsunehiko Kameda, for being my external examiner.

Working in the Communication Networks Laboratory has been rewarding experience for me. I want to acknowledge the fruitful cooperation with Dr. Fei Xue, and the help and encouragement from Nazy Alborz, Maryam Keyvani, Milan Nikolić, Giorgio Bombelli, and Allison Gau. Thank you for being wonderful friends and colleagues.

Dedication

To my parents and my wife Ema, for their endless love and support.

Contents

Abstract	iii
Acknowledgments	iv
Dedication	v
List of Tables	ix
List of Figures	xiii
1 Introduction	1
2 Background and related work	4
2.1 Sources of loss in the Internet	5
2.1.1 Sources of loss in the routers	5
2.1.2 Sources of loss on the links	6
2.2 Packet loss characterization	7
2.2.1 Unconditional and conditional loss probability	7
2.2.2 Two-state Markov model (Gilbert model)	7
2.2.3 Extended Gilbert model	9
2.2.4 General Markov chain model	11
2.2.5 Heavy-tailed distribution of packet loss	12
2.3 Background on self-similar processes	13
2.3.1 Definition of long-range dependence and self-similarity	13
2.3.2 Estimation of the Hurst parameter	15

3	Methodology	17
3.1	Obtaining packet loss data	17
3.1.1	Measurement of packet loss	17
3.1.2	Loss collection using simulation	18
3.2	Network simulator <i>ns-2</i>	20
3.2.1	Simulation input scripts	21
3.2.2	Trace-driven simulations	21
3.2.3	Simulation output traces	22
3.2.4	Extraction of loss data	23
4	Simulation scenarios	24
4.1	Simulation using a simple topology	24
4.1.1	Simulation scenarios	24
4.1.2	Source traffic	26
4.2	Simulation using a complex topology	27
4.2.1	Choice of the background traffic	28
5	Simulation results and loss modeling	33
5.1	Definition of loss episodes	34
5.2	Simple topology scenario - UDP sources	35
5.2.1	Loss rates	35
5.2.2	Textured dot strip plots of loss patterns	36
5.2.3	Loss episodes	41
5.3	Simple topology scenario - TCP sources	42
5.3.1	Loss rates	45
5.3.2	Loss episodes	47
5.4	Simple topology scenario - mixed UDP/TCP sources	49
5.4.1	Loss rates	50
5.4.2	Loss episodes	51
5.5	Complex topology scenario	54
5.6	Comparison with the Gilbert models	55
6	Analysis of loss on multiple time scales	60

7 Conclusion and future work	64
7.1 Conclusion	64
7.2 Future work	66
References	68

List of Tables

4.1	Encoder and trace parameters summary.	27
4.2	Parameters for the Web sessions.	31
4.3	Parameters for the FTP sessions.	32
5.1	Loss rates for the UDP simulations with <i>Star Wars</i> and <i>Talk show</i> traces.	36
5.2	Loss rates for the TCP simulations (“trace-driven” and FTP) and UDP simulations with the <i>Star Wars</i> trace.	46
5.3	Contribution of loss episodes of various lengths to the overall number of loss episodes. The number of sources = n , the buffer size $B = 100$ KB.	47
5.4	Number of loss episodes of length k for a specific UDP and TCP flow. Simulation with 80 UDP and 40 TCP sources. The buffer size is $B = 100$ KB.	51
5.5	Input data for the Gilbert models.	57

List of Figures

2.1	Elements of a communications system for multimedia transfer	5
2.2	Generic schematic of a router	6
2.3	Gilbert model	8
2.4	Multistate Markov models with infinite state space	10
2.5	Multistate Markov models with finite state space	10
2.6	General Markov chain model of order 2 (for illustration, only some of the transition probabilities are shown)	12
2.7	Pareto probability density function.	14
2.8	Pareto and exponential probability density functions plotted on a semilog scale.	14
4.1	Simple network topology: n sources, a router, and a sink.	25
4.2	Traffic pattern for the <i>Star Wars</i> trace.	28
4.3	Complex network topology.	29
4.4	ftp session, ftpdata bursts, and ftpdata connections.	31
5.1	Definition of loss episode and loss distance.	35
5.2	Loss rates at the router buffer for the UDP simulations with <i>Star Wars</i> and <i>Talk show</i> traces. The number of sources $n = 100$, the buffer size $B = 100$ KB, and the packet size is 200 bytes.	37
5.3	Loss rates at the router buffer for the UDP simulations with the <i>Star Wars</i> trace. The observation interval is from 60 to 600 sec and the loss rate calculation interval is 5 sec. The number of sources $n = 100$, the buffer size $B = 100$ KB, and the packet size is 200 bytes.	38

5.4	Loss rates at the router buffer for the UDP simulations with the <i>Star Wars</i> trace. The observation interval is from 200 to 300 sec and the loss rate calculation interval is 1 and 5 sec. The number of sources $n = 100$, the buffer size $B = 100$ KB and the packet size is 200 bytes. .	38
5.5	Loss rates of a particular source (top) and aggregate arrival rate (bottom) for the UDP simulations with the <i>Star Wars</i> trace. The observation interval is from 200 to 300 sec. The loss rate calculation interval is 1 and 5 sec and the arrival rate calculation interval is 5 sec. The number of sources $n = 100$, the buffer size $B = 100$ KB, and the packet size is 200 bytes.	39
5.6	Textured dot strip plot of packet loss instances at the router buffer from a simulation run with $n = 80$ sources and buffer size $B = 100$ KB.	40
5.7	Aggregate number of dropped packets in intervals of duration 5 sec. Simulation run with $n = 80$ sources and buffer size $B = 100$ KB.	40
5.8	Contribution of loss episodes of various lengths to the overall number of loss episodes. Simulation with the <i>Star Wars</i> trace, n sources, and buffer size $B = 50$ KB.	43
5.9	Contribution of loss episodes of various lengths to the overall number of loss episodes. Simulation with the <i>Star Wars</i> trace, n sources, and buffer size $B = 50$ KB. This figure is a zoomed version of Figure 5.8 depicting on a linear scale only the values for loss episodes of length 1, 2 and 3.	43
5.10	Contribution of loss episodes of various lengths to the overall number of loss episodes. Simulation with the <i>Star Wars</i> trace, n sources, and buffer size $B = 100$ KB.	44
5.11	Contribution of loss episodes of various lengths to the overall number of loss episodes. Simulation with the <i>Star Wars</i> trace, n sources, and buffer size $B = 100$ KB. This figure is a zoomed version of Figure 5.10 depicting on a linear scale only the values for loss episodes of length 1, 2 and 3.	44
5.12	Contribution of loss episodes of various lengths to the overall number of loss episodes. Simulation with the <i>Talk show</i> trace, n sources, and buffer size $B = 50$ KB.	45

5.13	Length of loss episodes for a single TCP source vs. time. The number of sources in the simulation is $n = 120$	48
5.14	Congestion window of a single TCP source vs. time. The number of sources in the simulations is $n = 120$	48
5.15	Contribution of loss episodes of various lengths to the overall number of loss episodes. TCP sources, buffer size $B = 50$ KB.	49
5.16	Scenario with m UDP and $n - m$ TCP sources. The shaded circles represent one specific UDP and one specific TCP source (numbered 1 and $m+1$, respectively).	50
5.17	Average loss rates for each source. The first 80 sources (numbered from 1 to 80) are using UDP, and the last 40 sources (numbered 81 to 120) are using TCP. The buffer size is $B = 50$ KB.	52
5.18	Average loss rates for each source. The first 80 sources (numbered from 1 to 80) are using UDP, and the last 40 sources (numbered 81 to 120) are using TCP. The buffer size is $B = 100$ KB.	52
5.19	Average loss rates for each source. The first 40 sources (numbered from 1 to 40) are using UDP, and the last 80 sources (numbered 41 to 120) are using TCP. The buffer size is $B = 50$ KB.	53
5.20	Average loss rates for each source. The first 40 sources (numbered from 1 to 40) are using UDP, and the last 80 sources (numbered 41 to 120) are using TCP. The buffer size is $B = 100$ KB.	53
5.21	Contribution of loss episodes for the mixed UDP/TCP scenario. The total number of sources is fixed ($n = 120$), the number of UDP sources is m and the number of TCP sources is $n - m$. The buffer size is $B = 100$ KB.	56
5.22	Contribution of loss episodes for the router buffer $R_3 \rightarrow R_4$ and a specific UDP video flow $S \rightarrow D$	56
5.23	Comparison of the loss episode length probabilities obtained from the simulation data with the loss episode length probabilities obtained from the Gilbert model. The number of sources $n = 100$, the buffer size $B = 50$ KB, the packet size is 200 bytes, the <i>Star Wars</i> trace is used, and the loss from source number 50 is observed.	58

6.1	The $\log_2(\Gamma)$ vs. j plot of the UDP packet loss process and the aggregate input traffic for buffer size $B = 100$ KB [56].	61
6.2	The $\log_2(\Gamma)$ vs. j plot of the TCP packet loss process and the aggregate input traffic for buffer size $B = 100$ KB [56].	61
6.3	Variance-time plot of the UDP loss process obtained through simulation with $n = 100$ sources and buffer size $B = 100$ KB.	62
6.4	R/S plot of the UDP loss process obtained through simulation with $n = 100$ sources and buffer size $B = 100$ KB.	62

Chapter 1

Introduction

The last decade of the 20th century has been marked with tremendous growth of the Internet. Several key parameters indicating the size of the Internet have experienced exponential increase: the number of users, hosts, registered domain names, and the amount of transferred data. For example, the number of hosts grew from 1,313,000 in January 1993 to 72,398,092 in January 2000, doubling almost every year [27]. The increase of the amount of data in the Internet is even faster [18]. For example, at the UUNet (now part of WorldCom Inc.) the traffic growth for the one-year period between 1998 and 1999 surpassed 1000%.

One of the reasons for this extremely fast growth is the proliferation of many new services in the largest “network of networks”. The Internet becomes a common platform for exchange of various information and media types. The applications such as World Wide Web, File Transfer Protocol and E-mail still have a large share of the overall network traffic [53], but new services such as video and audio streaming, video conferencing, Internet telephony, exchange of multimedia files, and many more, are increasing the total amount of data in the network and reshaping the existing traffic patterns. The changing nature of the traffic patterns requires sufficient network resources to satisfy the requirements of the new emerging applications. The requirements of the interactive applications delivering multimedia content can be very stringent and are usually expressed in terms of packet loss, delay, delay jitter, and throughput. Certain applications either fail to function or perform poorly if their requirements are not met.

The Internet, in its today’s stage, does not have a mechanism that would guarantee

the values for packet loss, delay, or throughput. Although high-capacity fiber-optic links and fast routers are being added to the Internet infrastructure, the problem of congestion remains. Congestion in the network means increased loss rates and delay, and thus degraded application performance.

In the past years, many approaches have been suggested to support Quality of Service (QoS) guarantees for multimedia applications. The focus is on both the end systems and the network as main components for delivering QoS guarantees. Understanding the behavior of the QoS parameters such as loss or delay is crucial for many of the research endeavors to make the Internet offer more than “best-effort” service.

The primary goal of this research is to gain better understanding of the loss process in the Internet Protocol (IP) networks. We expect that having better insight into the loss behavior might help many design decisions for enabling or improving QoS support. We also want to explore the viability of the simulation-based approach for collection of loss data, as opposed to end-to-end measurements.

The key tasks and deliverables provided by this research are:

- Collection of sets of loss processes for two dominant transport protocols using network simulation tools. With a simulation-based approach, we can have access to *all* the data needed for loss analysis: all the lost packets at the routers in our simulated network as well as all the generated data from all the sources.
- Generation of representative traffic is an extremely important issue for network simulations. The performance of the network (either simulated or real) is related to the amount and characteristics of the source input traffic. We employ a combination of trace-driven simulation as well as simulation with generated Web and FTP traffic with parameters according to existing traffic models.
- Collection and characterization of loss episodes. We focused our analysis on the contribution of loss episodes for both per-flow and aggregate loss at a router buffer. We show that lengthier loss episodes emanate from the UDP transfers, while the TCP transfers are characterized by shorter loss episodes due to the adaptive nature of the TCP sources.
- Comparison of the contribution of UDP loss episodes with existing loss models, such as the Gilbert model and Extended Gilbert model. The contribution of

loss episodes for both the aggregate loss at a router buffer and the loss of a single UDP stream showed approximately geometric decrease with increase of the length of the loss episode. This behavior is modeled well with the simple Gilbert model. For the longer loss episodes, the Gilbert model deviates from the simulated data and underestimates the contribution of longer loss episodes. The Extended Gilbert model describes the contribution of loss episodes more accurately. However, both models do not take into account the durations of the consecutive successfully received packets.

- Characterization of loss on multiple time scales. Our analysis suggests that the aggregate loss at a router buffer has similar scale-dependent characteristics to the aggregate input traffic.

Background on loss analysis and characterization is provided in Chapter 2. We present an overview of the related work in this area and introduce the currently used models for loss analysis. Also, the corresponding methods for obtaining the model parameters from a loss trace are described.

We comment on the choice of simulation approach for performing analysis of loss behavior in Chapter 3. The Chapter supports the choice of network simulation as a flexible alternative to measurements for obtaining loss data and introduces the network simulator *ns-2*, which is widely used in the networking research community.

The various simulation scenarios used for obtaining the loss data are presented in Chapter 4. We provide details about the main two topologies used in the simulations (simple and complex). The deployed sources are using either UDP or TCP as a transport protocol. We especially emphasize the choice of background traffic for the complex topology,

The results of the analysis of loss episodes is presented in Chapter 5. We compare the influence of UDP and TCP, two major transport protocols in today's Internet, on the characteristics of loss and the durations of the loss episodes.

The loss data collected with this research have been used for further statistical analysis of the scale-dependent properties of loss. We briefly present this ongoing research in Chapter 6.

Directions for future work are given in Chapter 7. In the same Chapter, we present the conclusions of the thesis.

Chapter 2

Background and related work

This chapter gives an overview of the past and current work in loss analysis and modeling. The significance of loss analysis can be observed only in the context of an end-to-end information transfer. Therefore, we start with the basics of an end-to-end communication for multimedia traffic.

A unicast video or audio communication system can be represented by the block diagram in Figure 2.1 [32]. The analog audio or video signal is first digitized and fed into the encoder. The coder compresses the digital signal in order to reduce the bandwidth needed for transmission of the audio/video information. The compressed signal is passed to the network adapter. The network adapter passes packets containing the audio/video information to the network. After traversing several links and routers, packets reach the destination (for unicast communication) or multiple destinations (for multicast communication). The destination (receiver) performs decoding, error handling, and delay equalization of the received packets and performs digital to analog conversion of the application-level frames. The output device is usually a screen and/or loudspeakers.

Information loss can appear in almost any of the elements of the communication system shown in Figure 2.1. For example, at the encoder, portions of the original information are lost if lossy compression techniques are used. The network adapter, on both the sender and receiver sides, may introduce loss if the data arrives at a higher rate than the rate the network adapter can handle and its buffer is full. The network may also introduce a significant amount of loss.

On the other hand, the coding schemes for both video and audio advance rapidly

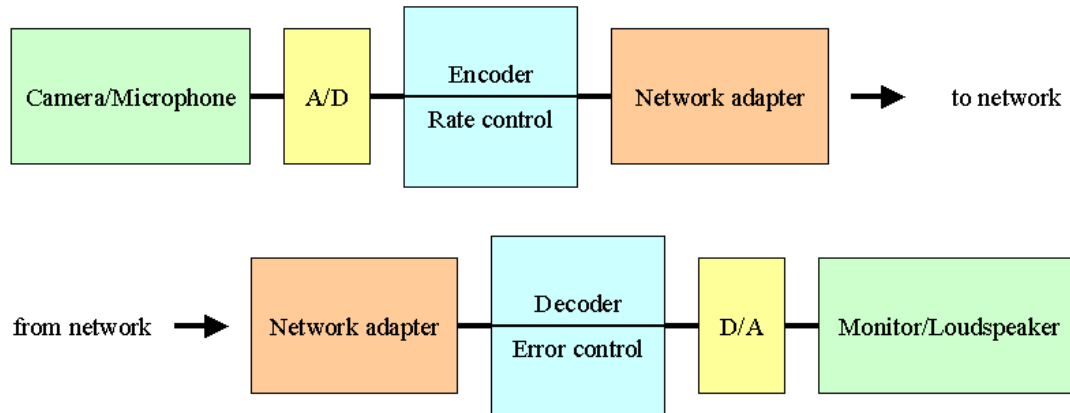


Figure 2.1: Elements of a communications system for multimedia transfer

and can provide high compression ratios while maintaining high perceptual quality. Also, network adapters usually have high data rates and are rarely a bottleneck for data transmission. However, loss of packets in the network can still have a huge impact on the audio/video quality perceived by the end users.

Why does loss appear in a network such as the current Internet? To answer this question, we have to obtain some insight into the structure of the Internet and look back to some of the original principles that led to the Internet creation.

2.1 Sources of loss in the Internet

The sources of loss in the Internet can be associated with the main elements of the network infrastructure: the routers and the links interconnecting the routers.

2.1.1 Sources of loss in the routers

If the load is higher than the forwarding speed of the routers, the packets are temporarily stored in the buffers of the routers, waiting for their turn to be forwarded on the output link. A typical router can be represented as in Figure 2.2 [33]. It consists of input and output ports, a switch fabric and a routing processor. The packets waiting to be switched at the input ports are stored in the input port buffers while switched packets waiting to exit on the output link are stored in the output port

buffers. The size of these buffers is finite. If the buffers are full, any new incoming packet is dropped. In other words, packets are lost due to *buffer overflow*.

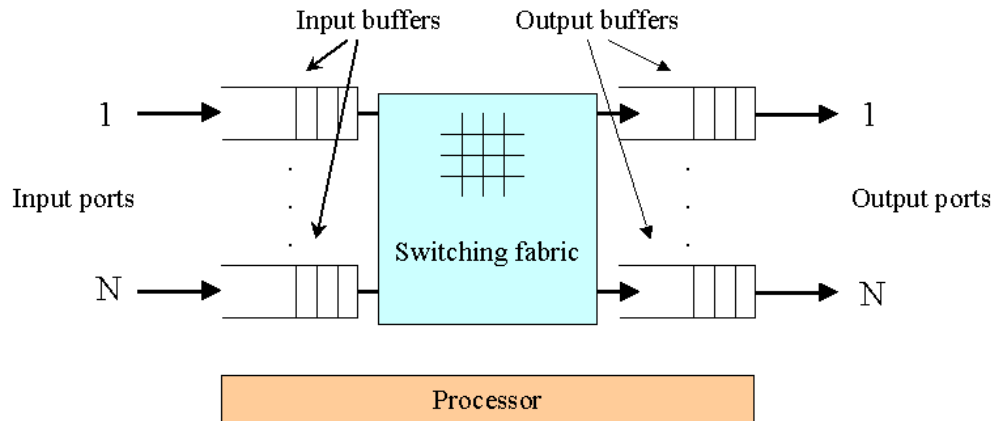


Figure 2.2: Generic schematic of a router

Buffer overflow is the main cause of loss in wireline networks accounting for more than 99 % of all the lost packets [28]. We must note that packet loss due to buffer overflow is a consequence of limited buffer size *and* congestion in the network. The congestion in the network, and thus the necessary condition for having packet loss, is not resolved by only increasing the buffer memory [30, 40]. Nagle shows that even with an infinite buffer size, packet loss is not eliminated [40]. Routers with infinite buffers (assuming that such routers exist) would drop packets not because of lack of space in the buffer, but because of expiration of the time-to-live (TTL) in the header of the IP packets.

Routing pathologies such as route loops [42] may also cause expiration of the TTL field and consequently packet drops.

2.1.2 Sources of loss on the links

Link failures and lossy links are another cause of packet loss. For example, a typical wireless link often has high bit-error rates due to channel fading, noise or interference, and intermittent connectivity due to handoffs. Corrupted packets might be dropped at any of the hops on the end-to-end path or at the receiver. Although sophisticated

coding and checksum schemes exist for detection and correction of bit-errors, unrecoverable packet corruption might occur, which is equivalent to a packet drop [8, 17, 42].

2.2 Packet loss characterization

2.2.1 Unconditional and conditional loss probability

One of the early works on measurement and analysis of packet loss in the Internet was done at the French computer science institute INRIA (Institut National de Recherche en Informatique et en Automatique) [13]. Using UDP probes containing sequence numbers and sent at regular intervals δ , Bolot measured the loss between the INRIA site and several universities in North America and Europe. The focus of the loss analysis was on loss probabilities only, such as unconditional packet loss probability $ulp = P(\text{packet } n \text{ is lost})$ and conditional probability $clp = P(\text{packet } n+1 \text{ is lost} | \text{packet } n \text{ is lost})$. Bolot showed empirically that $clp \geq ulp$. For very closely spaced probes (small values of δ), $clp > ulp$. In other words, if the previous packet was lost, then the probability that the next packet will be lost is higher. This correlation between two successive losses indicates that loss appears in bursts.

Unconditional loss probability ulp is an important parameter that affects the performance of end-to-end interactive applications. However, loss characteristics cannot be captured by specifying only the loss probability, and they require a detailed study of the loss patterns. Conditional loss probability clp refers to the loss of two consecutive packets, but does not provide insight into the behavior of loss episodes with lengths greater than two.

2.2.2 Two-state Markov model (Gilbert model)

Unicast and multicast measurements in the Internet [14, 58] indicated that the probability of loss of k consecutive packets (loss episode of length k) decreases approximately geometrically with k . This property is captured by a two-state homogeneous Markov chain. The state diagram of the model (also known as the Gilbert model) is shown in Figure 2.3.

State $X = 0$ represents the state of successful packet arrival and state $X = 1$

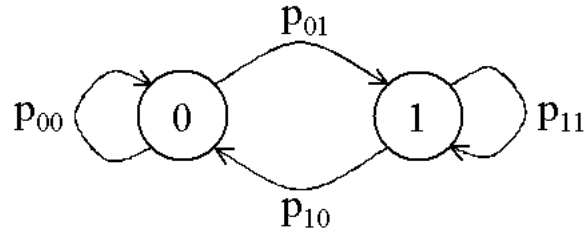


Figure 2.3: Gilbert model

represents lost packet. p_{01} and p_{10} denote the corresponding state transition probabilities:

$$\begin{aligned}
 p_{01} &= P(X = 1|X = 0) = P(\text{packet } n \text{ is lost} \mid \text{packet } n-1 \text{ is received}) \\
 p_{10} &= P(X = 0|X = 1) = P(\text{packet } n \text{ is received} \mid \text{packet } n-1 \text{ is lost}). \quad (2.1)
 \end{aligned}$$

For steady state, transition and state probabilities can be expressed in the following matrix form:

$$\begin{bmatrix} 1 - p_{01} & p_{10} \\ p_{01} & 1 - p_{10} \end{bmatrix} \begin{bmatrix} P(X = 0) \\ P(X = 1) \end{bmatrix} = \begin{bmatrix} P(X = 0) \\ P(X = 1) \end{bmatrix}. \quad (2.2)$$

From Equation 2.2 and the equation $P(0) + P(1) = 1$, we obtain the unconditional loss probability $P(X = 1) = \frac{p_{01}}{p_{01} + p_{10}}$. The conditional loss probability is equal to the probability of having a loss, given the previous packet was lost, which is $P(X = 1|X = 1) = 1 - p_{10}$. Equation 2.3 gives the probability of having a loss episode with length k (k consecutively lost packets) given that the flow had entered the loss state $X = 1$.

$$p_k = (1 - p_{10})^{k-1} \cdot p_{10}. \quad (2.3)$$

Consequently, the lengths of loss episodes (in the Gilbert model) are geometrically distributed.

The Gilbert model has a memory of only one past event. The probability that the next event will be either successfully received or lost packet depends only on the previous state.

Obtaining parameters for the Gilbert model from a packet trace

A set of formulas can be derived from the Markov model to obtain the values for p_{01} and p_{10} based on a packet trace containing the information about the lost and successfully received packets. We use the definitions and basic formulas introduced in [49]:

- o_k is the number of loss episodes of length k .
- a is the total number of packets (successfully received and lost).
- $d = \sum_{k=1}^{\infty} k \cdot o_k$ is the total number of lost packets.

It is shown that:

$$\begin{aligned} p_{01} &= P(X = 1|X = 0) = \sum_{k=1}^{\infty} \frac{o_k}{a} \\ 1 - p_{10} &= P(X = 1|X = 1) = \frac{\sum_{k=1}^{\infty} (k-1) \cdot o_k}{d-1}. \end{aligned} \quad (2.4)$$

2.2.3 Extended Gilbert model

Sanneck and Carle suggest the Extended Gilbert model for modeling the loss behavior [49]. The state space for the model can be either unlimited or limited. The state diagrams for both cases are shown in Figure 2.4 and Figure 2.5. State $X = 0$ refers to the state of successful packet arrival (same as in the Gilbert model), and the states $X \geq k$ ($k > 0$) refer to packet loss. For any k in the multistate model with infinite state space, and for $k < m$ in the model with finite state space (m being the number of states), $P(X = k)$ is the probability of having exactly k losses, and $P(X \geq k)$ is the probability of having at least k losses. For $k = m$ (in the finite state space), $P(X = k) = P(X = m)$ is the probability of having at least m losses.

The state probabilities can be determined from the state transition matrix, which has the following form:

$$\begin{bmatrix} p_{00} & p_{10} & \cdots & p_{(m-1)0} & p_{m0} \\ p_{01} & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & p_{(m-1)m} & p_{mm} \end{bmatrix} \quad (2.5)$$

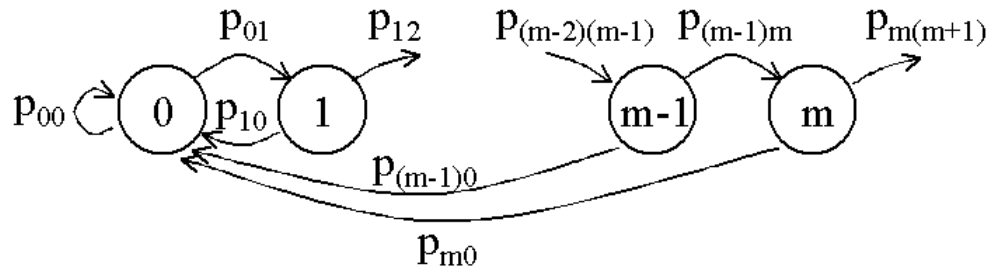


Figure 2.4: Multistate Markov models with infinite state space

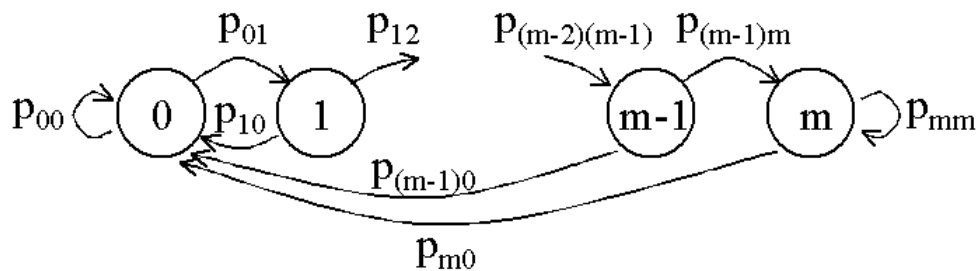


Figure 2.5: Multistate Markov models with finite state space

The Extended Gilbert model with finite state space has a memory of up to m loss events. The probability that the next event will be a successfully received or lost packet depends on the number of previously experienced consecutive losses. This is true if the previous state(s) are loss events. Both the Gilbert and the Extended Gilbert models have only one state describing the non-loss period. This means that the models cannot differentiate between one and more successfully received packets (non-loss). The probability of going from a non-loss state to either a loss or a non-loss state does not depend on the number of successfully received packets.

Obtaining parameters for the Extended Gilbert model from a packet trace

Using the definitions introduced in the previous subsection, the parameters p_{ij} can be expressed with the following equations:

$$\begin{aligned}
 p_{01} &= \frac{\sum_{k=1}^{\infty} o_k}{a} \\
 p_{00} &= 1 - p_{01} \\
 p_{i(i+1)} &= \frac{\sum_{k=i+1}^{\infty} o_k}{\sum_{k=i}^{\infty} o_k}, \quad 1 \leq i < m \\
 p_{i0} &= 1 - p_{i(i+1)}, \quad 1 \leq i < m \\
 p_{mm} &= \sum_{k=m}^{\infty} \frac{(k-m) \cdot o_k}{d-m} \\
 p_{m0} &= 1 - p_{mm}.
 \end{aligned} \tag{2.6}$$

2.2.4 General Markov chain model

The Markov chain model of order n has a memory of all the past n events, both loss and non-loss events [58]. The probability that the next event will be either a successfully received or a lost packet depends on the past n events, regardless of whether these events were losses or non-losses. For example, $P(X_n = 1 | X_{n-1} = 1, X_{n-2} = 1, X_{n-3} = 0, X_{n-4} = 1, X_{n-5} = 0)$ is the probability that a packet is lost, given that the packets $n-3$ and $n-5$ are successfully received, and packets $n-1$, $n-2$, and $n-4$ are lost. This gives more information than the Extended Gilbert model for which the above expression would shrink to $P(X_n = 1 | X_{n-1} = 1, X_{n-2} = 1, X_{n-3} = 0) = p_{23}$. This is because the Extended Gilbert model can only track the state history up to the most recent successfully received packet. The above example is given under the assumption that the last state m in the finite state Extended Gilbert model is greater than or equal to the order n of the general Markov model. The drawback of the general Markov chain model is that *all* the last n states should be remembered. Since these events can have a value of 1 or 0 (loss or non-loss), 2^n states are needed to track the history of n events. Figure 2.6 shows the state diagram for $n = 2$.

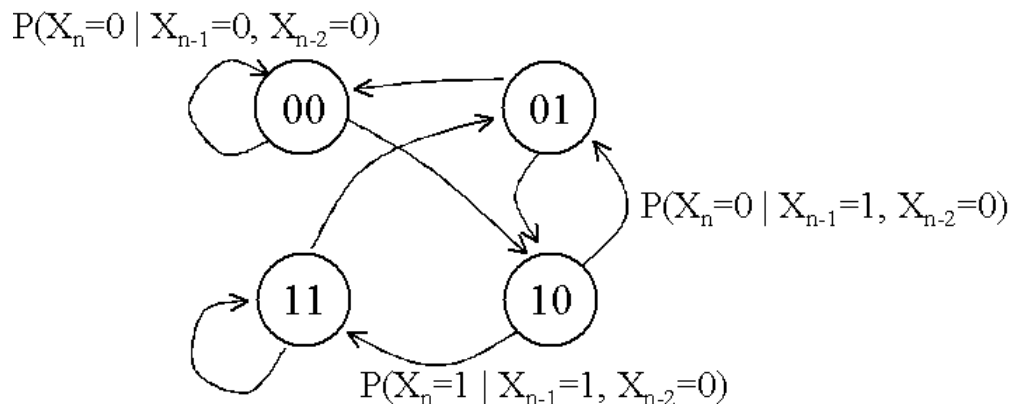


Figure 2.6: General Markov chain model of order 2 (for illustration, only some of the transition probabilities are shown)

Obtaining parameters for the general Markov chain model from a packet trace

The transition probabilities for the general Markov chain model of order n can be obtained by using the following equation [58]:

$$P(X_i = a | X_{i-1} = b_1, X_{i-2} = b_2, \dots, X_{i-n} = b_n) = \frac{n_{ba}}{n_{\underline{b}}} \quad \text{for } n_{\underline{b}} > 0. \quad (2.7)$$

where a and b_i can take values 1 or 0 (loss or non-loss), $n_{\underline{b}}$ is the number of occurrences of state \underline{b} ($X_{i-1} = b_1, X_{i-2} = b_2, \dots, X_{i-n} = b_n$) and n_{ba} is the number of transitions from state \underline{b} to state a .

2.2.5 Heavy-tailed distribution of packet loss

Based on empirical data obtained from end-to-end measurements of audio packet loss, Borella et al. use the Pareto distribution to model the distribution of loss episode lengths [16]. The Pareto distribution is a heavy-tailed distribution and is described with the shape parameter β and location parameter a .

$$F(x) = P(X \leq x) = 1 - \left(\frac{a}{x}\right)^\beta, \quad \text{for } a, \beta > 0, x \geq a. \quad (2.8)$$

The probability density function is:

$$f(x) = \beta a^\beta x^{-\beta-1}. \quad (2.9)$$

The probability density function for $\beta = 1.5$ and $a = 1$ is shown in Figure 2.7. If presented on a linear plot, the probability density function for the Pareto distribution resembles the plot for exponential distribution. However, on a semilog plot shown in Figure 2.8, one can clearly see the heavy tail, i.e., the slowly decreasing probability density function of the Pareto distribution, as opposed to the exponential distribution, where $f(x)$ decreases rapidly with x . For $1 < \beta \leq 2$, a random variable with Pareto distribution has an infinite variance. If $0 < \beta \leq 1$, both the mean and the variance are infinite.

Borella et al. use separate models for the lower and the upper tail of the packet loss distribution calculated from the measured data. Considering $a = 1$, the obtained values of $\beta = 2.84$ for the lower tail and $\beta = 0.53$ for the upper tail indicate a loss process of extreme burstiness.

2.3 Background on self-similar processes

The area of Internet traffic modeling has experienced many changes in the last decade. The major finding in several key papers in the field was that the traffic in local area networks (such as Ethernet) and wide area networks possesses self-similar properties [22, 23, 34, 36, 44]. It has also been found that self-similarity is an inherent characteristic of World Wide Web and Variable Bit Rate (VBR) video traffic, independent of scene and codec algorithms [20, 26].

2.3.1 Definition of long-range dependence and self-similarity

A continuous time process $X(t)$ is called self-similar with scaling factor H if the finite-dimensional distributions of $X(at)$ are equal to those of $a^H X(t)$, for all $a > 0$ [37]. A more appropriate definition for discrete time processes is given in [19, 36, 37].

Let $X = \{X_t\}$ ($t = 0, 1, 2, \dots$) be a covariance stationary stochastic process with variance $var(X)$, and autocorrelation function $r(k)$, $k \geq 0$. Let $\{X_k^{(m)}\}$ be new series obtained by averaging the original series in nonoverlapping blocks of size m ($m = 1, 2, \dots$).

$$X_k^{(m)} = \frac{1}{m}(X_{km-m+1} + \dots + X_{km}), \quad k \geq 1. \quad (2.10)$$

The new series $X^{(m)}$ has variance $var(X^{(m)})$ and autocorrelation function $r^{(m)}(k)$.

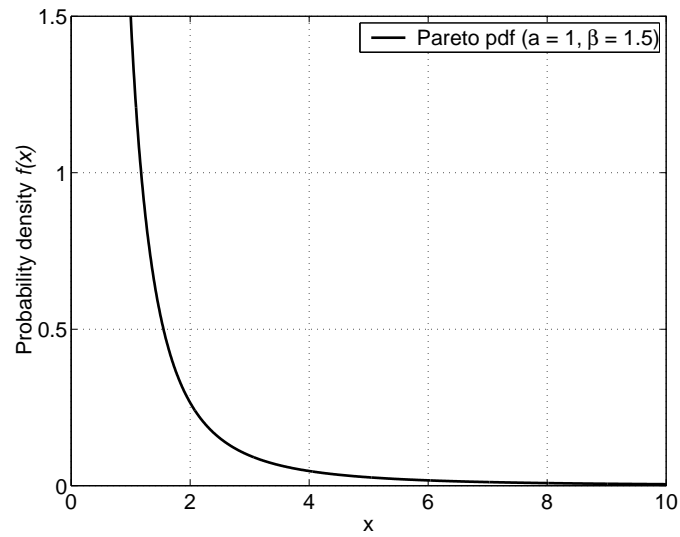


Figure 2.7: Pareto probability density function.

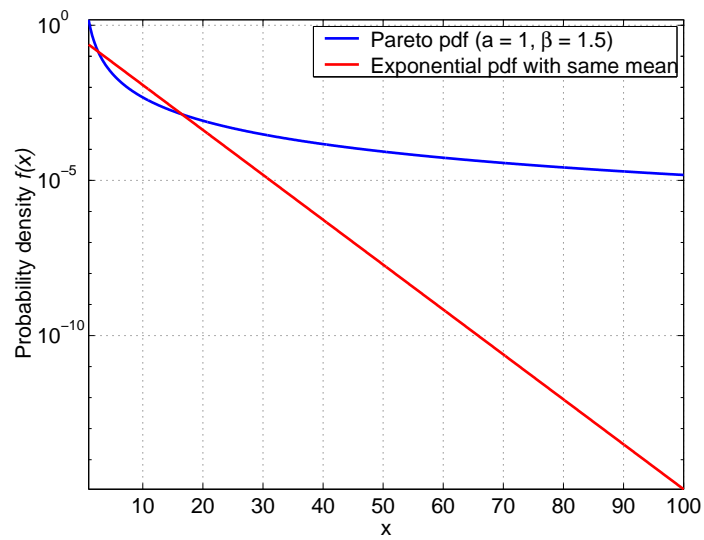


Figure 2.8: Pareto and exponential probability density functions plotted on a semilog scale.

Definition: A discrete time stochastic process X is called *long-range dependent* if it satisfies any of the following properties:

1. $\sum_k r(k) = \infty$
2. the spectral density $f(w) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} r(k)e^{-iwk}$ is singular near $w = 0$
3. the variance $\text{var}(X^{(m)})$ decreases more slowly than the reciprocal of the sample size m .

Let us now assume that the autocorrelation of the series X has the following form:

$$r(k) \sim |k|^{-\beta}, \quad 0 < \beta < 1. \quad (2.11)$$

Definition: A discrete time stochastic process X is called *exactly second-order self-similar* with self-similarity parameter $H = 1 - \beta/2$ ($0 < \beta < 1$) if, for all $m = 1, 2, \dots$:

- $r^{(m)}(k) = r(k), \quad k \geq 0, \quad \text{and}$
- $\text{var}(X^{(m)}) = \text{var}(X) \cdot m^{-\beta}.$

Exactly second-order self-similar means that the averaged processes $X^{(m)}$ have identical correlation structure with X .

Definition: A discrete time stochastic process X is called *asymptotically second-order self-similar* with self-similarity parameter $H = 1 - \beta/2$ ($0 < \beta < 1$) if, for all k large enough, $r^{(m)}(k) \rightarrow r(k)$, as $m \rightarrow \infty$.

Asymptotically second-order self-similar means that the averaged processes $X^{(m)}$ have a nondegenerate correlation structure, indistinguishable from X as $m \rightarrow \infty$.

2.3.2 Estimation of the Hurst parameter

Several methods have been proposed to estimate the self-similarity parameter H , also known as Hurst parameter. Long-range dependent processes have values for the Hurst parameter between 0.5 and 1.0. Statistically, the Hurst parameters can be estimated using several different methods: variance-time plots, R/S analysis, periodogram-based analysis, and wavelet analysis.

1. The above definition of self-similar processes implies the use of variance-time plots for estimation of the Hurst parameter. Variance-time plots are obtained by plotting $\log(\text{var}(X^{(m)}))$ vs. $\log(m)$. A simple least squares line is fitted to the plot and the corresponding slope represents the estimated value for $(-\beta)$. The estimated value for the Hurst parameter is $\hat{H} = 1 - \hat{\beta}/2$.
2. The R/S statistic or rescaled adjusted range statistic has been suggested as a robust method for testing the presence of long-range dependence [37]. The main characteristic of R/S analysis is its robustness with respect to the marginal distribution of the analyzed data. For a series X with length n , the rescaled adjusted range $R(n)/S(n)$ can be calculated by using the following equations:

$$\begin{aligned} W(k) &= \sum_{j=1}^k X_j - k \cdot \left(\frac{1}{n} \sum_{j=1}^n X_j \right), \quad k = 1, \dots, n \\ R(n) &= \max\{0, W(1), \dots, W(n)\} - \min\{0, W(1), \dots, W(n)\} \\ S^2(n) &= \text{sample variance of } X_1, \dots, X_n. \end{aligned} \quad (2.12)$$

The plot $\log(R(n)/S(n))$ vs. $\log(n)$ is called R/S plot, or *pox diagram*. The slope of the fitted least squares line is an estimate for the Hurst parameter.

3. Another approach for estimating the Hurst parameter is based on the periodogram $I(x) = (2\pi n)^{-1} |\sum_{k=1}^n X_k e^{ikx}|^2$ ($0 \leq x \leq \pi$ and $i^2 = -1$) of the series X [11]. The advantage of periodogram-based methods is the possibility to gain information about the confidence intervals for the estimated value.
4. A wavelet-based Hurst parameter estimator, suggested in [3], is based on a spectrum estimator obtained by performing a time average of the wavelet *detail coefficients* $|d_x(j, k)|^2$ at a given scale:

$$\Gamma_x(2^{-j}\nu_0) = \frac{1}{n_j} \sum_k |d_x(j, k)|^2. \quad (2.13)$$

where n_j is the number of wavelet coefficients at scale j . The linear relationship between $\log_2(\Gamma)$ and scale level j over a range $[j_1, j_2]$ indicates the presence of a long-range dependent behavior. The estimator \hat{H} for the Hurst parameter can be determined by performing a linear regression of $\log_2(\Gamma)$ on scale level j in the range $[j_1, j_2]$.

Chapter 3

Methodology

In order to analyze the packet loss process, we need specific information about when loss appears in the network, which packets are lost, whether or not the loss is clustered in bursts (episodes), and if so, how long these loss episodes last, and finally, what is the impact of network protocols on loss. The answers to these questions might reveal the loss pattern in the network. Gathering the data for loss analysis can be performed either by measurement of loss in a live network such as the Internet or by collection of loss in a simulated network. In this chapter, we describe our methods for obtaining the loss data, the choice of the network simulator and the approach for data analysis.

3.1 Obtaining packet loss data

3.1.1 Measurement of packet loss

Measurement is an essential step towards analysis and evaluation of system variables and their properties. The Internet is a huge and complex system, consisting of millions of networks, hosts, routers, and links. Ubiquitous measurement of variables important for the Internet's performance, such as packet loss, delay, throughput, at every network element and every stream is practically impossible. Therefore, many researchers focused either on end-to-end measurements using source-destination pairs [13, 15, 16, 21, 57], or on end-to-end measurements using a mesh of many Internet sites [5, 46].

Internet measurements can be classified into two categories: passive and active.

Passive measurements record the traffic flowing on a particular link or passing through a network element. Some of the early high-precision passive measurements were done between 1989 and 1992 on the local area network at the Bellcore Morristown Research and Engineering Center [36]. A custom-built traffic monitor was used for the purpose of collecting the packet header and recording an accurate timestamp. The increase of processor power, appearance of software tools for network monitoring and data acquisition such as `tcpdump` [29], and kernel modifications such as `bpf` [39] enabled general purpose workstations to act as passive traffic collection points. The main obstacles for ubiquitous deployment of the passive measurement approach are privacy and security issues.

Active measurements inject new traffic in the network. Usually, traffic with a predefined pattern is sent from host A to host B and specific parameters are measured at one of the endpoints. An example traffic generator for active measurements is Vern Paxson's Network Probe Daemon which is still used as a basic tool for the National Internet Measurement Infrastructure [46, 43]. Active measurements have the disadvantage of adding load to the actual Internet traffic, which might skew the results of the measurements. The advantage is the capability of obtaining metrics which are difficult, if not impossible, to obtain using passive measurements.

Packet loss measurements can be obtained using either of the above methodologies. For example, one way to perform passive measurements of packet loss at a router would be to collect the traffic at the router's input and output links. Comparison of the two data sets can reveal the packet loss process that occurs in the router. This approach has the limitation of measuring the loss at only one network element. An extension of this idea would be to measure the loss on all the network elements along the end-to-end path. However, having in mind that the Internet is partitioned into many administrative domains, with their own access, privacy, security, and measurement policies, this idea seems infeasible. Therefore, active measurements are more commonly used for obtaining insight into the packet loss process in a real network.

3.1.2 Loss collection using simulation

Simulation is another alternative for collecting loss data. However, use of simulation for network research has its limitations. The size of the simulated network depends on

the constraints of computer memory and processing power. Simulations are a model of reality with many real-world implementation details often abstracted. Abstractions might improve the simulator performance (depending on the level of abstraction), but might hide some important details at the same time.

Despite these drawbacks, simulation provides numerous advantages over live Internet measurements. Simulation at the packet level gives access to *all* the data pertaining to a particular simulation scenario. The output traces of the simulations provide detailed information about the generated, lost (dropped), and queued packets with the corresponding timestamps, sequence numbers, source and destination addresses, and other parameters of interest that can be set up in the simulation. While we can focus on a per-flow (end-to-end) loss using end-to-end measurements, it is very difficult, if not impossible, to measure detailed statistics about the loss process that appears inside the network. With packet-level simulation, the overall packet behavior can be observed and the data can be collected at *any* node in the network. Another important advantage of simulation is that the results are reproducible, which enhances the analysis and comparison of results [7].

Simulation also enables great flexibility in choosing the parameters that influence the occurrence of loss: size of buffers in routers, link speeds, background traffic, etc. The size of the router buffers have a direct effect on the amount of loss in the network. A small buffer implies a small maximum queueing delay, but also, high loss for bursty traffic. The speed of the links connecting the routers determines the transmission delay of a single packet (equal to the ratio of the packet size and the output link speed), which determines how fast the router buffer is emptied. Although the amount of background traffic directly influences the loss in the network, it should always be considered in relation to the other parameters of the network.

One should be aware of the limitations of simulations and be careful with the interpretation of the results because a simulated network always contains abstractions of certain aspects of real network implementations. For loss analysis, it is important to compare the loss behavior of the simulated scenarios with the loss behavior of already performed measurements in the Internet. End-to-end loss data can be obtained using simulation. Comparison of the results for end-to-end loss from a simulation environment with results for end-to-end loss already obtained by measurements can be a good validation of the insight that the simulation can provide into the Internet loss

behavior. In other words, the comparison between simulation and empirical results can serve as a test for the validity of the simulation results and their connection to the “real-world” network behavior.

In this research, we use simulation to collect the data necessary for loss analysis. The flexibility of exploring various simulation scenarios and unrestricted data access were the primary reasons for our choice.

3.2 Network simulator *ns-2*

In order to obtain the loss data, we use *ns-2*, a network simulator widely adopted in the network research community. *ns-2* evolved as a part of the VINT (Virtual InterNetwork Testbed) project, a collaborative project among University of Southern California, Xerox PARC, Lawrence Berkeley National Laboratory, and the University of California, Berkeley [54]. The main goal of the VINT project is to provide a simulation framework for analysis and development of Internet protocols.

ns-2 is a discrete event simulator which provides support for:

- various network protocols (transport, multicast, routing)
- simple or complex topologies (including topology generation)
- agents (defined as endpoints where network-layer packets are constructed or consumed)
- various traffic generators
- simulated applications (FTP, Telnet, Web)
- several queue management and packet scheduling schemes
- error models
- local area networks
- wireless networks

The code of the simulator is written in C++ and Object Tcl (OTcl). There is one-to-one correspondence between a class in C++ (compiled hierarchy of classes in *ns-2*) and a class in OTcl (interpreted hierarchy). This software architecture (also called *split programming model*) enables high-performance simulation of packet level routines (implemented in C++) and flexible configuration and flexible control of the simulation using an interpreted language such as OTcl [7].

The following subsections contain some of the *ns-2* specific details for our simulations. The description of the simulation scenarios is given in Chapter 4.

3.2.1 Simulation input scripts

All the necessary information to configure and control a simulation run in *ns-2* is given in the form of an input OTcl script. The simulation objects (nodes, links, traffic sources) are instantiated with the script, and immediately mirrored in the compiled hierarchy. The input script defines the topology, builds the agents (sources and destinations), sets the trace files and sets the start times for the initial events in the simulation. The initial events might later generate new events. For example, the user can only specify the start of an FTP session and the number of packets to be transferred. This is an initial event indicating the start of the FTP session. When the FTP transfer starts, new events will be generated such as an arrival at a router queue, a check of the TTL field, a departure from a router queue, an arrival at a receiver, a generation of an ACK packet etc. The simulator always executes events in the order specified in the event list, which is always sorted by time.

3.2.2 Trace-driven simulations

Trace-driven simulation in *ns-2* is enabled with the OTcl class `Application/Traffic/Trace` which is a subclass of the traffic generator class `Application/Traffic`. The objects of the class `Application/Traffic/Trace` are attached to a particular trace file. *ns-2* accepts a trace file as a list of 32-bit pairs <interarrival time, packet size>. The first value indicates the time in μsec until the next packet is generated, while the second value indicates the size of the next packet in bytes. Our simulations, explained in detail in Chapter 4, were driven by genuine video traces (*Star Wars* and *Talk show*). In order to avoid synchronization of the generated traffic by the sources,

each source starts at random point within the traffic trace. If the end of the trace is reached before the end of the simulation, the source continues from the beginning of the trace.

3.2.3 Simulation output traces

Tracing in *ns-2* can be performed by using *trace* or *monitor* objects. Trace objects collect the data for each packet arrival, departure or drop. Monitor objects collect data on an aggregate level and are implemented as counters of specific parameters of interest (total number of packet or byte arrivals, departures or drops). Monitor objects are useful when basic information about the dynamics of the simulation is needed. However, in order to have a comprehensive understanding of the loss patterns, we perform tracing on a per packet basis. The aggregate information that can be obtained by monitor objects is not sufficient to support a detailed study of the observed stochastic process (in our case: packet loss process).

An output trace in *ns-2* has a fixed format, shown in the following example:

```
r 17.922645 201 202 tcp 552 ----- 0 10.47 191.17 28 49925
+ 17.922645 202 203 tcp 552 ----- 0 10.47 191.17 28 49925
d 17.922645 202 203 tcp 552 ----- 0 10.47 191.17 28 49925
r 17.922648 203 199 udp 200 ----- 0 204.0 199.60 3143 49809
r 17.922744 203 202 ack 40 ----- 0 191.12 10.42 394 49934
+ 17.922744 202 201 ack 40 ----- 0 191.12 10.42 394 49934
- 17.922744 202 201 ack 40 ----- 0 191.12 10.42 394 49934
- 17.922913 202 203 ack 40 ----- 0 133.90 180.10 72 49904
r 17.922968 203 199 udp 200 ----- 0 204.0 199.60 3144 49813
```

The first column indicates the type of event: packet is received (r), enqueued (+), dequeued (-) or dropped (d). The second column is the timestamp of the corresponding event. The third and fourth column point to the nodes in the topology between which the tracing occurred. The fifth column contains a description of the the particular packet (TCP data, TCP ack, UDP, CBR, etc.). The sixth column contains the packet size in bytes. The seventh column is rarely used and might contain various flags. The

eighth column gives the IP flow identifier as defined in IPv6. The next two columns represent the source and destination address. Column eleven is the packet sequence number and the last column indicates the unique packet id.

3.2.4 Extraction of loss data

From the output trace shown above, the data of interest can be obtained. We use the Unix command `gawk` to extract the time instances and sequence numbers of lost packets. We focus on extraction and analysis of:

1. Aggregate loss that occurs at a router (all the lost packets due to buffer overflow at a certain router)
2. Per-flow loss (all the lost packets that belong to a specific end-to-end flow)

Chapter 4

Simulation scenarios

This chapter explains the details of the simulation scenarios: topology, traffic sources and protocols, and composition of the background traffic.

4.1 Simulation using a simple topology

In order to gain an initial understanding of the traffic loss patterns at a router buffer, we first consider a simple network topology consisting of n sources generating traffic and feeding a common router buffer connected to a traffic sink (destination), as shown in Fig. 4.1. The topology is generated automatically by selecting the number of video sources n in the input scripts for the *ns-2* simulator. The speeds of the links that connect the sources to the router are chosen to be larger than the peak bit-rate of the employed traffic sources. In this manner, the traffic from the sources arrives unaltered to the router buffer. The link speed between a particular source and the router is 10 Mbps, and the propagation delay is 1 msec. The link speed between the router and the sink is 44.736 Mbps, and the propagation delay is 5 msec. The router employs a First In First Out (FIFO) buffer with a DropTail queue management policy. The simulation runs lasted over 20 min, and the data was collected and analyzed for the time period between 60 and 1200 sec.

4.1.1 Simulation scenarios

We consider three scenarios for the simple network topology:

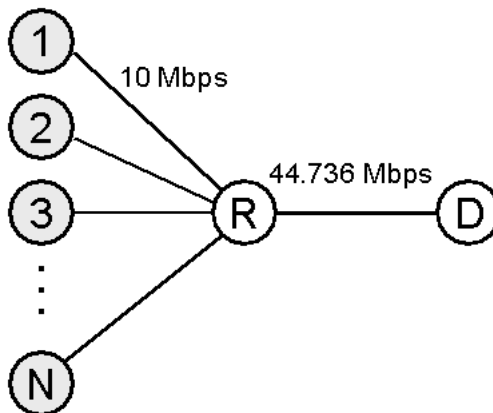


Figure 4.1: Simple network topology: n sources, a router, and a sink.

1. In Scenario 1, all the sources use UDP transport protocol to deliver a video trace to the traffic destination. The sources are driven by genuine traffic traces. The idea behind this scenario deploying only UDP sources is to observe the aggregate loss behavior at a router buffer in the presence of one protocol *only*. This scenario might also reveal how the loss behavior is influenced by the choice of the transport protocol, in this case UDP. One possible example for the above scenario could be a number of multimedia servers connected to a common edge router. We are aware that in reality, we rarely have isolated flows of traffic using one protocol. Usually, the traffic on the Internet links and the traffic that is fed to a certain router uses a mix of transport protocols [53]. However, the opportunity to simulate traffic sources that use only one particular transport protocol might give insight into how the loss behavior is influenced by the choice of transport protocol.

The UDP sources in the simple scenario are generating their traffic according to a traffic trace and sending the data using 200-byte, 552-byte and 1500-byte packets. The first consideration while choosing the packet size was for applications requiring small packetization delay and the second consideration was to have a comparable packet size with the sizes used for TCP transport.

The buffer size B is varied from 25 to 200 KB in steps of 25 KB, which corresponds to a maximum queuing delay of 4.58 to 36.62 msec, respectively.

The chosen values are reasonable for video communications. The values below 10 msec are within the range of latencies specified for high-end interactive video [24]. The number of sources n is varied from 70 to 100, thus spanning a utilization of the output link of between 60% and 90%.

2. Scenario 2 consists of sources sending the same data using TCP. The network topology consists of n TCP sources feeding their traffic to a common router. Again, only one transport protocol is used, in this case TCP. When using TCP transfers, larger packet sizes are desirable in order to maximize the transfer throughput. For the TCP transfers in our simulations, the data is sent using 552-byte and 1500-byte packets rather than 200-byte packets as used in some of the UDP simulations. All the comparisons of the loss patterns in the case of either UDP or TCP transfers are performed using equivalent packet sizes. The buffer size B is varied from 50 to 200 KB in steps of 50 KB.
3. In scenario 3, UDP and TCP transfers are mixed. The total number of sources is still n , but the ratio of the number of UDP to TCP sources varies.

4.1.2 Source traffic

The simulations for the simple topology scenarios are performed with genuine MPEG-1 encoded traces as source traffic [26, 48]. For our analysis, we used the *Star Wars* and *Talk show* video traces [1, 2]. The encoder and trace parameters summary is given in Table 4.1. The traffic pattern for the *Star Wars* movie is shown in Fig. 4.2. We can clearly see from Table 4.1 and Fig. 4.2 that the source traffic is bursty. This is typical for a trace obtained by MPEG video encoding, which produces variable bit-rates depending on the scene activity and scene complexity in the original uncompressed movie. For example, in the case of the *Star Wars* trace, the peak bit-rate (calculated over a period of one frame) is more than 12 times larger than the average bit-rate (calculated over the period of the trace duration - 2 hours).

In order to avoid synchronization of the generated traffic by the sources, each source begins at a random time point within the traffic trace. If the end of the trace is reached before the end of the simulation, the source continues sending the packets from the beginning of the traffic trace.

Encoder parameters	<i>Star Wars</i>	<i>Talk show</i>
encoder input (pel)	480x504	384x288
resolution (bits/pel)	8	8
pattern	IBBPBBPBBPBB	IBBPBBPBBPBB
GOP size	12	12
frame rate (frames/sec)	24	25
number of video frames	174,136	40,000
peak rate (Mbps)	4.446	2.669
mean rate (Kbps)	374.4	363.4
peak/mean ratio	12.23	7.13
coefficient of variation	1.165	1.136

Table 4.1: Encoder and trace parameters summary.

4.2 Simulation using a complex topology

In order to perform more realistic simulations with a variety of traffic sources, we use a more complex topology, suggested in [9], and shown in Fig. 4.3. The routers are denoted with R , and the shaded nodes are sources and/or destinations of Web, FTP, or video traffic. We generate the network topology automatically, and our input scripts allow for variations in the number of routers (transit nodes), their buffer sizes, the interconnecting link speeds, and the number of hosts (end nodes) attached to the routers. Each router is connected to the neighboring routers and to a number of hosts. For our particular simulation set, we employed 200 host nodes and 4 transit nodes although our scripts enable generation of topology with a user specified number of transit and host nodes. The links between the routers are 100 Mbps except for the last two routers (R_3 and R_4 on Fig. 4.3). The link between these last routers simulates the bottleneck in the network and for different simulation runs, its speed is varied between 1.5 Mbps and 10 Mbps. The hosts are connected to the routers with fast links with speeds between 22 and 32 Mbps.

For every router, five of the hosts are chosen to be Web servers. The remaining hosts are Web clients, FTP, or video traffic sources/destinations. Web clients start their sessions by randomly choosing one of the Web servers in the network. FTP and video sessions can be initiated between any two hosts.

We observe the loss process in the buffers of the routers R_3 and R_4 . Besides the

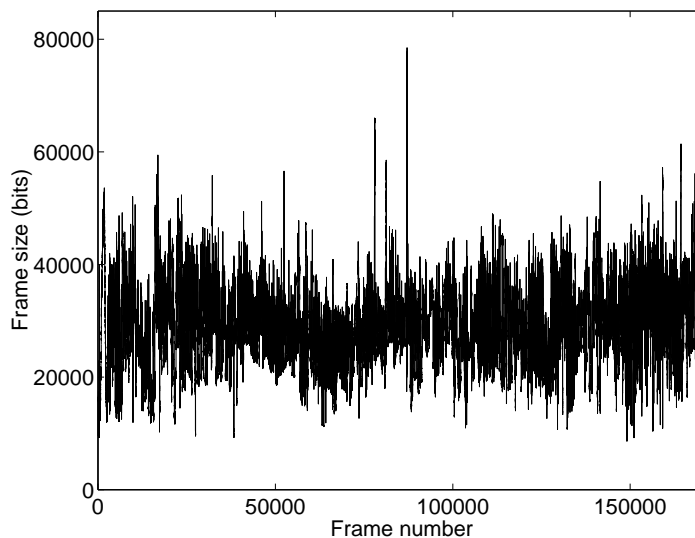


Figure 4.2: Traffic pattern for the *Star Wars* trace.

aggregate loss, we also observe the loss process of a particular UDP video flow from source S to destination D .

4.2.1 Choice of the background traffic

The generation of representative traffic is an extremely important issue for network simulations. The performance of the network (either simulated or real) is related to the amount and characteristics of the traffic given as an input by the sources. An open question in the Internet research community is how to characterize the traffic generated by the sources as well as the aggregate traffic on the links. The diversity of Internet traffic is a key problem associated with its characterization. A number of different applications at the end hosts use various protocols to generate and feed traffic with diverse characteristics into the network. Capturing this behavior in a concise description is not an easy task.

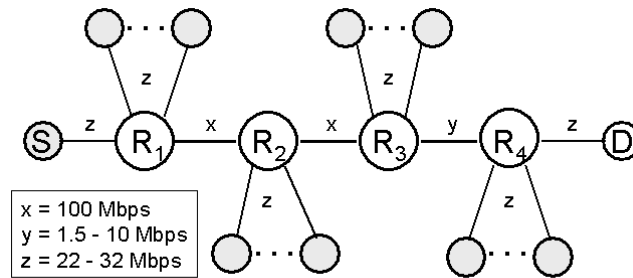


Figure 4.3: Complex network topology.

Aggregate and source traffic

Despite the difficulties in characterizing the “immense moving target”, as the authors of [45] refer to the Internet’s diversity and drastic change over time, advances in the area of traffic modeling and characterization enabled the improvement of traffic synthesis for network simulations. Measurement and modeling has been applied to both aggregate and source traffic.

As mentioned in Chapter 2, one of the major findings of the traffic characterization efforts was that Ethernet, wide-area, and World Wide Web traffic exhibit self-similar properties [20, 36, 44]. Measurements on the MCI backbone links in 1997 revealed that the most dominant protocols in the Internet are TCP and UDP [53]. All these studies were performed for measured *aggregate* traffic traces.

Besides the modeling of aggregate traffic, other studies also focused on the *source* application behavior, by analyzing the traffic for various application level protocols like FTP, Telnet, Simple Mail Transfer Protocol (SMTP), and Network News Transfer Protocol (NNTP) [41], or by analyzing the logs of web clients or web servers [6, 10].

We use these results to set up the traffic generation scenarios in our network.

Web traffic simulation parameters

The Web traffic generation in our complex scenario is according to the SURGE model proposed in [10] and also used in [22]. The parameters for the user Web sessions are presented in Table 4.2. All the parameters refer to the Pareto distribution of the

second kind, which is defined with the following equation:

$$\begin{aligned} F(x) &= 1 - \frac{a^\beta}{(x+a)^\beta} \\ f(x) &= \beta a^\beta (x+a)^{-\beta-1}, \quad \text{for } x, a, \beta > 0. \end{aligned} \quad (4.1)$$

This distribution has a similar form to the Pareto distribution of the first kind defined in the Equation 2.8 and Equation 2.9, but the random variable can take on values in the wider range $(0, \infty)$ instead of (a, ∞) (recall that a is the location parameter of both types of Pareto distributions).

The Web related parameters in Table 4.2 set the generation of the web traffic in our simulations. Currently, *ns-2* supports HTTP 1.0 (Hypertext Transfer Protocol) [12]. Using this protocol, a web server responds to a client's Web-page request by sending several Web objects (the main part of the page, inlined images and other associated data).

The Web parameters of interest for our simulation are:

- Inactive OFF time (inter-page time): Time between two consecutive Web-page requests by the user. This parameter is also called “think time”.
- Number of page components (objects): Number of objects (inlined images and associated data) that are downloaded with the initially requested Web page.
- Active OFF time (inter-object time): Time between the transfer of two objects.
- Page component size (object size): Size of the downloaded Web object.

HTTP 1.0 establishes a separate TCP connections for the delivery of every Web object, even though these objects are part of the same Web page, usually residing on the same Web server. The specification of HTTP 1.1 introduces persistent connections and pipelining [25], but this behavior is not incorporated in the current *ns-2* implementation.

FTP traffic simulation parameters

The choice of the FTP parameters in our simulation was based on the statistical analysis of wide-area traffic on [41, 44] and the Internet backbone measurements reported in [53]. The FTP protocol enables basic file sharing between two hosts [47].

Web page parameter	Pareto parameters	Mean value
Inactive OFF time (Inter-page time)	$a = 5$ $\beta = 2$	10 sec
Number of page components (Number of objects per page)	$a = 1$ $\beta = 1.5$	3
Active OFF time (Inter-object time)	$a = 0.167$ $\beta = 1.5$	0.5 sec
Page component size (Object size)	$a = 2$ $\beta = 1.2$	12 KB

Table 4.2: Parameters for the Web sessions.

Separate TCP connections are created for the control and data information. An *ftp session* consists of one or more *ftpdata connections*, which represent the transfer of particular files or directory listings. The durations of the *ftpdata connections* can have various lengths (depending on the transfer rate and the size of the transferred file). If the interarrival time between two *ftpdata connections* is smaller than 4 sec, then these connections belong to the same *ftpdata burst*. The value of 4 sec (maximum interarrival time for *ftpdata connections* belonging to the same *ftpdata burst*, or minimum interburst time) is according to [44]. Figure 4.4 reflects the definitions of an *ftp session*, *ftpdata bursts*, and *ftpdata connections*.

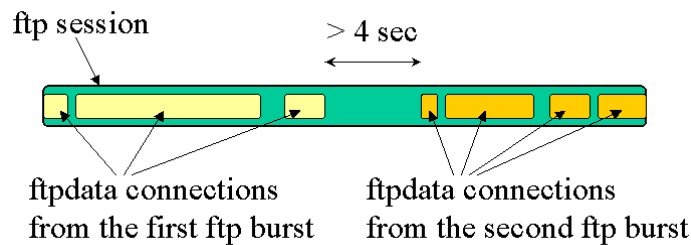


Figure 4.4: ftp session, ftpdata bursts, and ftpdata connections.

Generation of ftp sessions in the simulation is governed by an exponential distribution. The mean interarrival rate of the ftp sessions is equal to the total number of sessions in the simulation divided by the simulation time. The total number of sessions is assigned at the start of the simulation and this value can be varied.

The generation of ftp bursts within one session also follows an exponential distribution, but shifted by the minimum interburst time of 4 sec. In other words, the probability density function of the interburst times is $f(x) = \lambda e^{-\lambda(x-4)}$, $x \geq 4$. The mean interburst time is $(1/\lambda) + 4$ and is chosen to be equal to 5 sec ($1/\lambda = 1$ sec).

One ftp session can consist of several bursts and we chose this number to be uniformly distributed in the range of 1 to 4 with a mean value of 2.5 bursts per session. Recall that the number of ftp bursts does not reflect the number of transferred files during the ftp session. Each ftp burst is an aggregate of one or more *ftpdata connections* corresponding to the transfer of individual files or directory listings. As mentioned above, we model only ftp bursts in the simulation and not the individual *ftpdata connections* within the bursts.

The number of bytes in each burst follows the Pareto distribution of the first kind with $\beta = 1.18$ and mean value of 80,000 bytes. With an average of 2.5 bursts per session and 80,000 bytes per burst, the mean size of each ftp session is 200,000 bytes, which corresponds to the average ftp session size measured in wide-area networks [53].

A summary of the FTP parameters is given in Table 4.3.

FTP parameter	Distribution	Mean value
Session interarrival time	Exponential $f(x) = \lambda_1 e^{-\lambda_1 x}$	$\frac{\text{simulation_time}}{\text{number_of_ftp_sessions}}$
Burst interarrival time	Exponential $f(x) = \lambda_2 e^{-\lambda_2(x-4)}$, $x \geq 4$	5 sec
Burst size	Pareto of first kind $f(x) = \beta a^\beta x^{-\beta-1}$ $\beta = 1.18$, $a = 12,203$	80,000 bytes

Table 4.3: Parameters for the FTP sessions.

Video traffic simulation parameters

The parameters for video traffic are the same as for the simple topology and are described in Section 4.1.2. The video sessions have random starting points, last for a random amount of time and are initiated between two random hosts. The minimum duration of the video sessions is 60 seconds (to account for short video clips or movie trailers) and the maximum duration is limited by the maximum simulation time.

Chapter 5

Simulation results and loss modeling

Probability of loss is a very important parameter that affects the quality of interactive multimedia applications. The probability of loss or loss rate is a summary statistic, and is calculated over a certain period of time by dividing the number of dropped packets by the total number of generated packets. This definition refers to the loss rate experienced by a certain traffic source. We can also calculate the loss rate at a router by dividing the number of dropped packets by the total number of packets that arrived at the router. In both cases, the calculation is made with input values (dropped and received packets) obtained over a specific period of time.

Although the loss rate is an important parameter, it cannot capture the detailed loss patterns in the network. For the same loss rate, different loss patterns can have different effects on multimedia applications.

Consecutive packet losses have a huge impact on the quality of a multimedia transfer. While single losses can be tolerated and/or corrected by the applications, several consecutive losses cannot and result in degraded video or audio quality.

In this Chapter, we present the results for the contribution of lost packets and loss episodes for the simple and complex topology.

5.1 Definition of loss episodes

One or more consecutively lost packets constitute a *loss episode*. This definition can be applied to a loss of a particular source traffic or to a loss that happened at a particular router buffer.

If the packets are from one source, the loss episode contains the lost packets with consecutive sequence numbers. In other words, a loss episode begins with a lost packet if the previous packet was successfully received. For example, if the packets with sequence numbers 1, 4, and 6 are successfully received and packets 2, 3, and 5 are lost, then the first loss episode begins with packet 2 and ends with packet 3, while the second loss episode begins and ends with packet 5.

If we are referring to a router, we define a loss episode as the number of packets that arrived one after another at the router, but were dropped due to buffer overflow.

Lost packets are the constituents of loss episodes. If we say that a loss episode is of length k , this means that the loss episode consists of exactly k consecutively lost packets. For the above example, the length of the first loss episode is two packets, and the length of the second loss episode is one packet.

Loss distance is a measure of the spacing between two consecutively lost packets. It is defined as “the difference in sequence numbers of two successively lost packets which may or may not be separated by successfully received packets” [35]. If these two successively lost packets are separated by one or more successfully received packets, then the loss distance is a measure of the spacing between two consecutive loss episodes. The definition of *loss episodes* and *loss distances* is illustrated in Figure 5.1.

Before proceeding with the results for the distribution of loss episodes and lost packets, we will introduce the following variables:

$o_k(t_{start}, t_{end})$: number of loss episodes of length k packets

$O_{total}(t_{start}, t_{end})$: total number of loss episodes

$npkt_k(t_{start}, t_{end})$: number of lost packets belonging to loss episodes of length k

$Npkt(t_{start}, t_{end})$: total number of lost packets

L_{max} : maximum length of all the loss episodes (length of the longest loss episode)

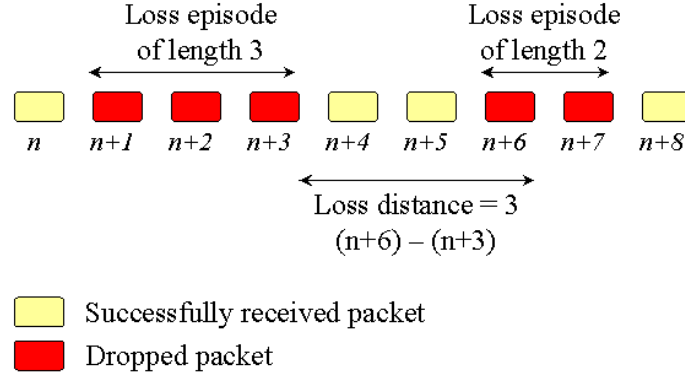


Figure 5.1: Definition of loss episode and loss distance.

The number of lost packets from loss episodes of length k is equal to:

$$npkt_k(t_{start}, t_{end}) = k \cdot o_k(t_{start}, t_{end}). \quad (5.1)$$

t_{start} and t_{end} denote the start and the end of the observation interval.

5.2 Simple topology scenario - UDP sources

5.2.1 Loss rates

We start the analysis of loss with an overview of the loss rates at the router buffer for the simulations with UDP traffic. Table 5.1 provides a summary of the number of arrived and dropped packets at the router buffer for the simulations with the *Star Wars* and *Talk show* traces. The buffer size is varied from 25 to 200 KB, corresponding to maximum queueing delays of 4.58 to 36.6 msec. The number of deployed sources is $n = 100$ and the packet size is 200 bytes. It is clear that the loss rate is higher for the simulations with the *Star Wars* trace, due to the burstier nature of this particular traffic trace (see Table 4.1). For both cases, the increase of the buffer size results in a reduction of the loss rate. However, the tradeoff is an increased value of the maximum queueing delay. The loss rates for the simulations with the two video traces are shown in Figure 5.2.

The loss rates in Table 5.1 are calculated for the observation interval [$t_{start} = 60$ sec, $t_{end} = 1200$ sec]. Loss rates can also be calculated over shorter intervals. We

$n = 100$ pktsize = 200 bytes		<i>Star Wars</i> (Arrived: 27,451,164)		<i>Talk show</i> (Arrived: 25,866,487)	
Buffer size (KB)	Max queuing delay (msec)	Dropped	Loss rate (%)	Dropped	Loss rate (%)
25	4.58	4,512,616	16.4	1,524,967	5.90
50	9.16	3,279,716	11.9	845,210	3.27
75	13.73	2,121,275	7.73	461,867	1.79
100	18.31	1,130,829	4.12	223,571	$8.64 \cdot 10^{-1}$
125	22.89	577,287	2.10	93,431	$3.61 \cdot 10^{-1}$
150	27.47	268,873	$9.79 \cdot 10^{-1}$	24,765	$9.57 \cdot 10^{-2}$
175	32.05	79,563	$2,90 \cdot 10^{-1}$	1,406	$5.44 \cdot 10^{-3}$
200	36.62	13,621	$4.96 \cdot 10^{-2}$	0	0

Table 5.1: Loss rates for the UDP simulations with *Star Wars* and *Talk show* traces.

can calculate the loss rate over the interval $[t_{start}, t_{start} + T]$, and slide t_{start} to cover the observation interval. This is shown in Figure 5.3 for $T = 5$ sec, $t_{start} = 60$ sec and $t_{end} = 600$ sec. The period between 200 and 300 sec is zoomed and shown in Figure 5.4 for $T = 1$ and $T = 5$ sec. The reduction of the loss rate calculation interval shows that there exists a variability in the loss rates, which cannot be captured by specifying only the long-term average loss rate.

The loss rates shown in Figures 5.3 and 5.4 refer to the loss at the router buffer. The loss rates for a particular source traffic, for example source number 50 in the simulation with 100 UDP sources is represented in Figure 5.5. Parallel to this figure, we show the aggregate arrival rate calculated on a time scale of 5 sec.

As mentioned earlier, loss rate represents an aggregate measure about the underlying loss process. It does not provide any information about how the loss is distributed within the observed time interval. Therefore, we further consider the loss patterns in more detail, focusing on the loss episodes.

5.2.2 Textured dot strip plots of loss patterns

In order to investigate the patterns of $o_k(t_{start}, t_{end})$ and $npkt_k(t_{start}, t_{end})$, we examine the overall loss pattern of a particular simulation run with the *Star Wars* trace, 80 UDP sources, and buffer size of 100 KB. According to the output link speed (44.736

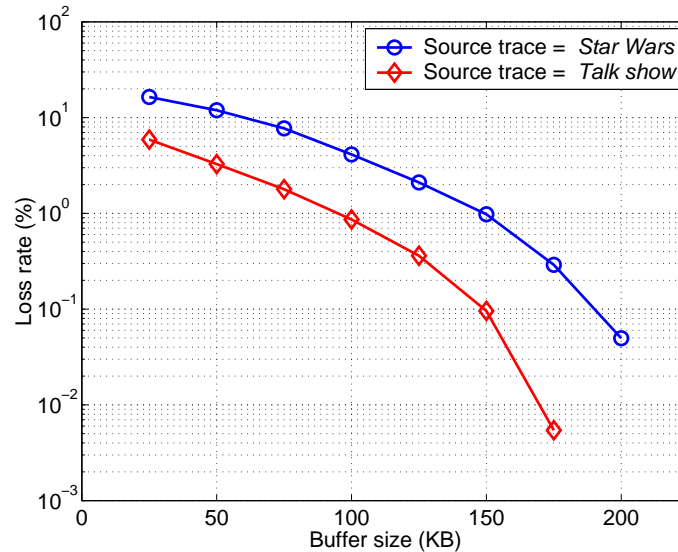


Figure 5.2: Loss rates at the router buffer for the UDP simulations with *Star Wars* and *Talk show* traces. The number of sources $n = 100$, the buffer size $B = 100$ KB, and the packet size is 200 bytes.

Mbit/s), this buffer size corresponds to a maximum buffer delay of 18.31 msec.

We use textured dot strip plots to provide an initial insight into the loss patterns. The idea of the textured dot strip plots is to depict the densities of the observed variable. In our case, the observed variable is the time when the loss occurs. The dots are randomly placed along the vertical axis within the height of the plot for better visibility of the bands with more or less losses. We obtained these plots by using *XGobi*, a tool for interactive graphics and data analysis [51]. Textured dot strip plots have also been used for illustrations of traffic patterns [55]. From the textured dot strip plot in Figure 5.6, we can identify periods of high congestion or low congestion. One of the periods of high congestion is band A between 615 and 625 sec. A period of lower congestion is band B between 836 and 846 sec. Two long periods with no loss are band C between 826 and 836 sec and band D between 846 and 854 sec [38].

An alternative approach to observing the bands of higher or lower congestion is to count the number of lost packets within intervals (bins) covering the observation period. This is shown in Figure 5.7 for fixed-size intervals of 5 sec. Both the textured

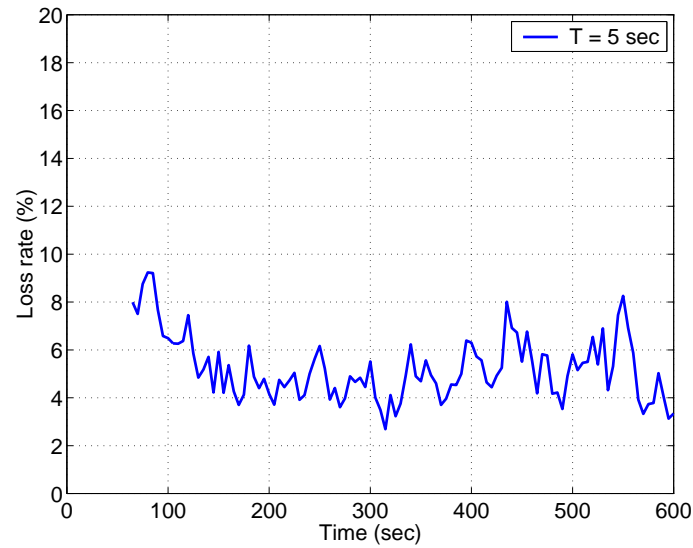


Figure 5.3: Loss rates at the router buffer for the UDP simulations with the *Star Wars* trace. The observation interval is from 60 to 600 sec and the loss rate calculation interval is 5 sec. The number of sources $n = 100$, the buffer size $B = 100$ KB, and the packet size is 200 bytes.

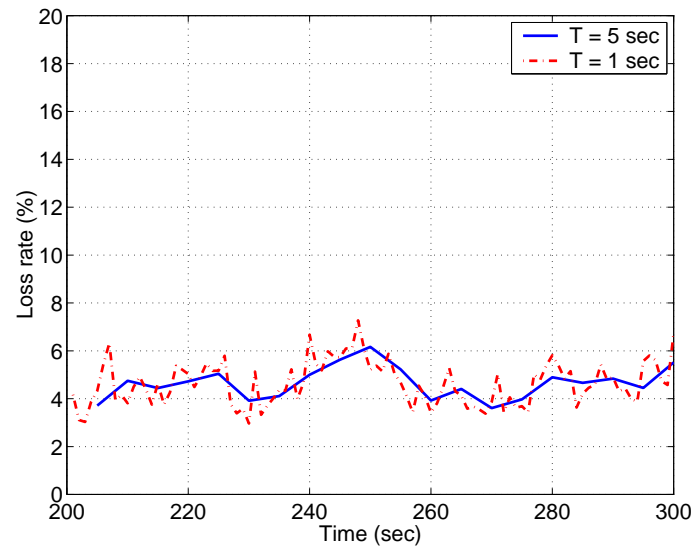


Figure 5.4: Loss rates at the router buffer for the UDP simulations with the *Star Wars* trace. The observation interval is from 200 to 300 sec and the loss rate calculation interval is 1 and 5 sec. The number of sources $n = 100$, the buffer size $B = 100$ KB and the packet size is 200 bytes.

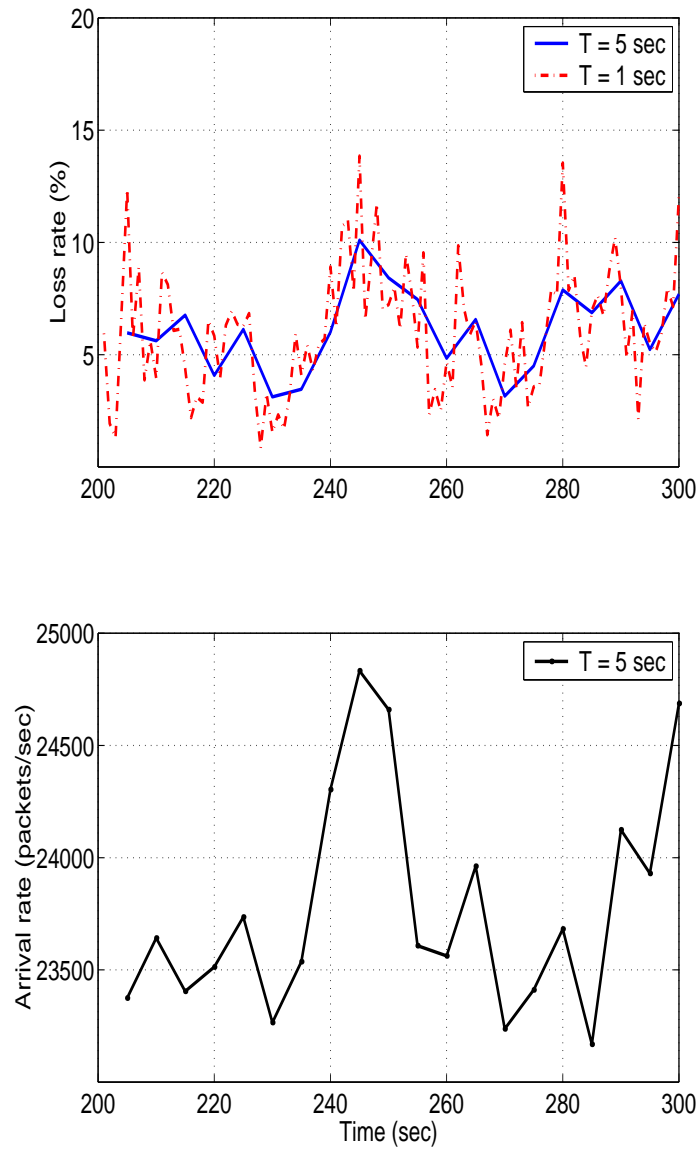


Figure 5.5: Loss rates of a particular source (top) and aggregate arrival rate (bottom) for the UDP simulations with the *Star Wars* trace. The observation interval is from 200 to 300 sec. The loss rate calculation interval is 1 and 5 sec and the arrival rate calculation interval is 5 sec. The number of sources $n = 100$, the buffer size $B = 100$ KB, and the packet size is 200 bytes.

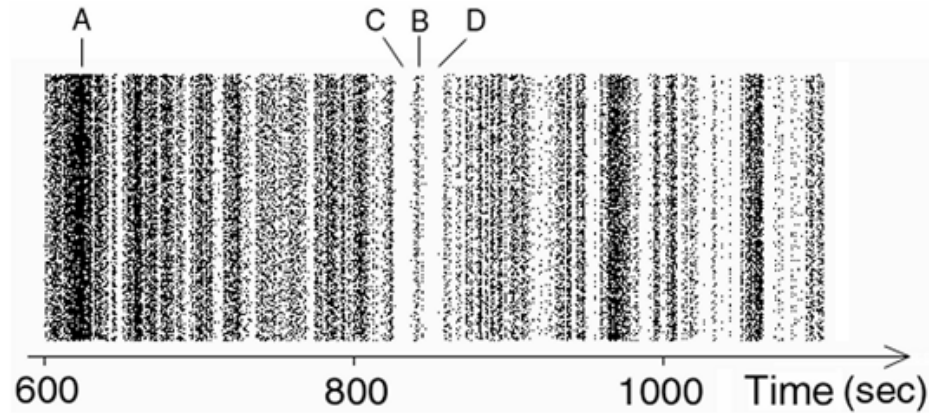


Figure 5.6: Textured dot strip plot of packet loss instances at the router buffer from a simulation run with $n = 80$ sources and buffer size $B = 100$ KB.

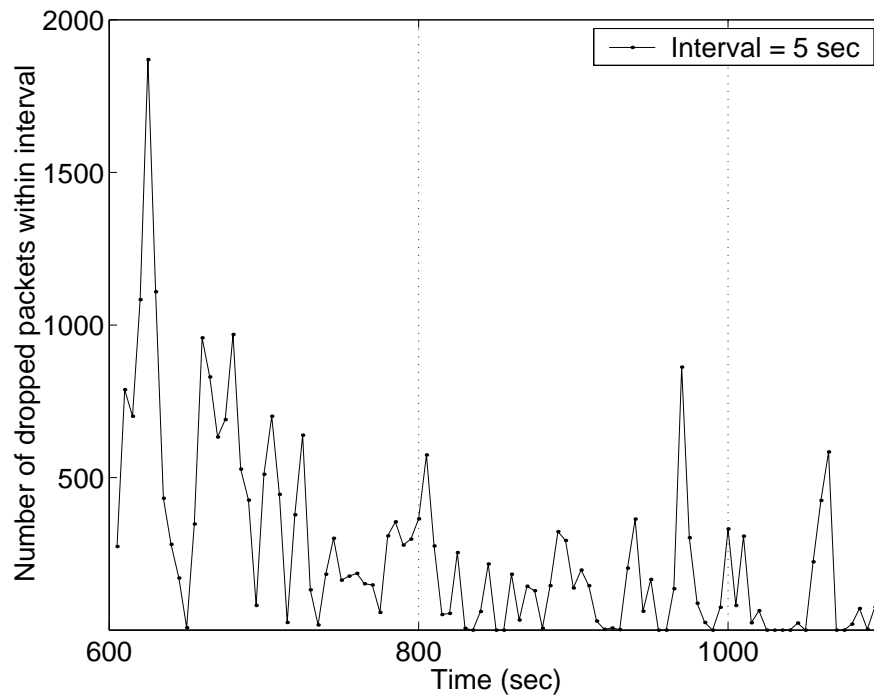


Figure 5.7: Aggregate number of dropped packets in intervals of duration 5 sec. Simulation run with $n = 80$ sources and buffer size $B = 100$ KB.

dot strip plot and the plot showing the absolute frequency of lost packets within fixed-size bins can give us initial insight into the loss pattern. However, this information consists of the time instances of only the lost packets. We did not take the packets that were successfully received into consideration. Having both the times of successfully received and dropped packets, we can identify the loss episodes, which are consisted of consecutively lost packets.

5.2.3 Loss episodes

Using the variables defined in Section 5.1, we can calculate the contribution of loss episodes and lost packets [38]. The contribution expressed as a percentage of loss episodes of length k is equal to $o_k(t_{start}, t_{end}) \cdot 100/O_{total}(t_{start}, t_{end})$, i.e., the ratio between the number of loss episodes of length k and the overall number of loss episodes. The contribution of lost packets from a loss episode of length k is given by $npkt_k(t_{start}, t_{end}) \cdot 100/Npkt(t_{start}, t_{end})$, which is the ratio between the number of lost packets from loss episode of length k and the total number of lost packets in the interval (t_{start}, t_{end}) (also expressed as a percentage). For the UDP simulations with the *Star Wars* trace and buffer sizes of 50 and 100 KB, the results are shown in Figures 5.8–5.11.

We will focus first on the single losses (loss episodes of length one) at the router buffer for a fixed buffer size. For example, if we take a closer look at Figure 5.9 and Figure 5.11, we can see that with the increase of the number of traffic sources (and, accordingly, the input traffic load), the contribution of loss episodes of length one decreases. This is also reflected on the contribution of lost packets to loss episodes of length one. The decrease of the contribution of loss episodes of length one and lost packets from these loss episodes is a direct result of the increased congestion at the router. As we increase the number of sources (while maintaining a fixed buffer size), we expect that the router buffer will be congested more often due to the increased traffic load. In the case of UDP, there is no feedback mechanism that will tell the sources to slow down in response to network congestion. The losses that appear due to buffer overflow are more frequent, and the probability that these losses will be consecutive increases. In other words, with increased congestion at the router, it is more likely that the loss episodes will be longer than one packet (more consecutively lost packets).

Consequently, the probability that we will have single losses (loss episodes of length one) decreases.

The same reasoning explains why we have a smaller contribution of single losses for a fixed number of traffic sources, but smaller buffer size (compare for fixed n , Figures 5.9 to 5.11).

We also calculated the contribution of loss episodes and lost packets at the router buffer for the simulations with the *Talk show* trace. The corresponding graph for the loss episodes for the case with a buffer size of 100 KB is shown in Figure 5.12. The results are qualitatively similar to those obtained using the *Star Wars* trace.

An interesting phenomenon can be observed from Figures 5.9 and 5.11: for a fixed buffer size and variable number of sources, the percentage of loss episodes of length two remains the same. Although we cannot explain the reason for the existence of such an intersecting point, we suspect that the explanation might be the coding scheme used for the video streams.

5.3 Simple topology scenario - TCP sources

In the scenario with UDP sources, the network was unable to provide explicit or implicit feedback about its current state. Even when the router buffer was full or close to its maximum size, the UDP sources still generated traffic whenever they had packets to send. Contrary to UDP, TCP reacts to the implicit feedback about the network conditions in the form of a timeout for a sent packet or reception of several duplicate acknowledgments. After sending a window of data packets, TCP stops the transmission and waits for acknowledgments of the sent packets. Each acknowledgment arrival spurs an increase of the congestion window (denoted by $cwnd$) and thus, further sending of data packets. The increase in the congestion window is more rapid during the *slow start* phase and slower in the *congestion avoidance* phase [28]. When loss is detected, the congestion window is reduced. Lost packets cause the retransmission timer at the sender to expire (since the senders will not receive an acknowledgment from the receiver for the lost packet) or cause several duplicate acknowledgments (if the packets following the lost packet were successfully received and their acknowledgments reached the sender). In effect, lost packets indicate to the sender, by means of retransmission timer expiry or reception of multiple duplicate acknowledgments, that

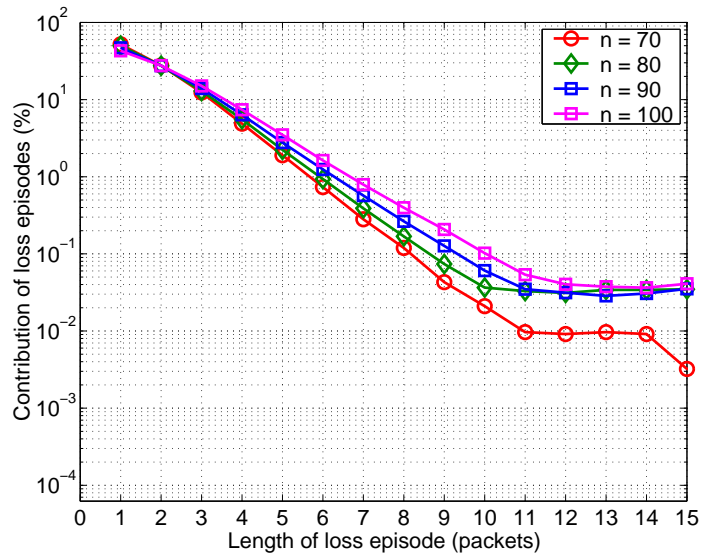


Figure 5.8: Contribution of loss episodes of various lengths to the overall number of loss episodes. Simulation with the *Star Wars* trace, n sources, and buffer size $B = 50$ KB.

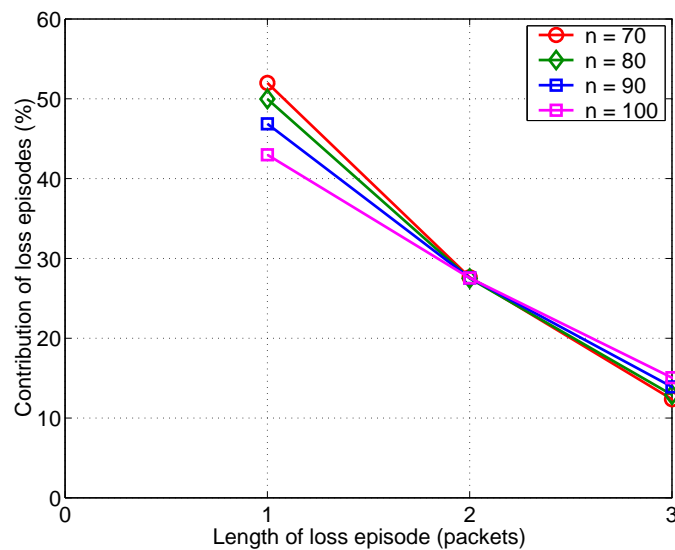


Figure 5.9: Contribution of loss episodes of various lengths to the overall number of loss episodes. Simulation with the *Star Wars* trace, n sources, and buffer size $B = 50$ KB. This figure is a zoomed version of Figure 5.8 depicting on a linear scale only the values for loss episodes of length 1, 2 and 3.

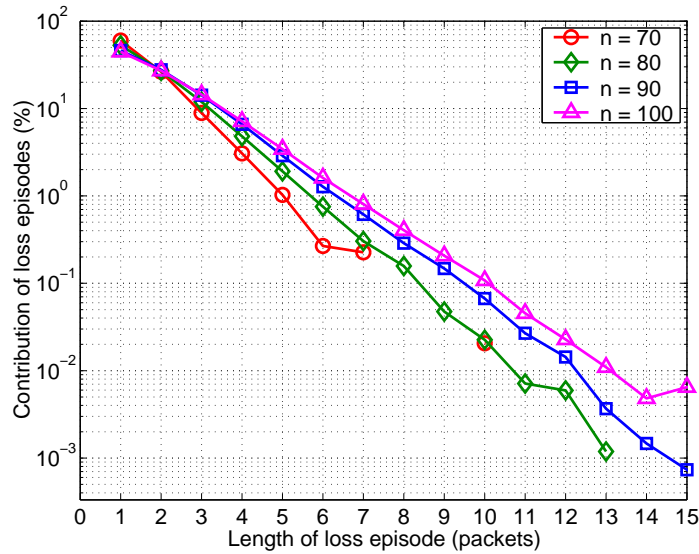


Figure 5.10: Contribution of loss episodes of various lengths to the overall number of loss episodes. Simulation with the *Star Wars* trace, n sources, and buffer size $B = 100$ KB.

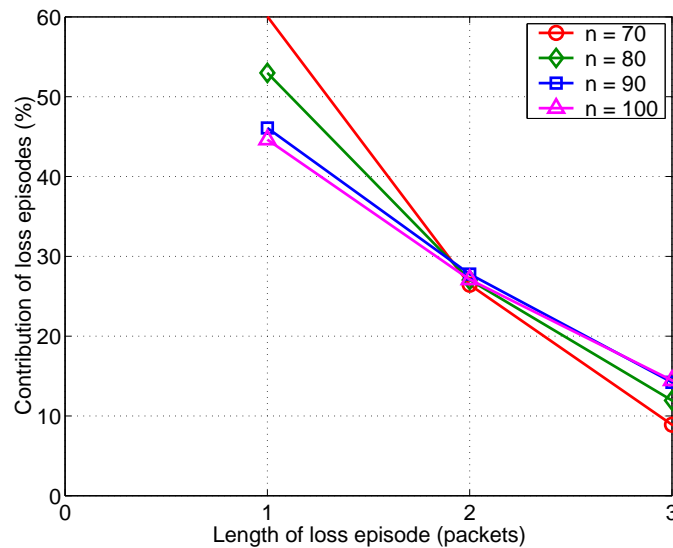


Figure 5.11: Contribution of loss episodes of various lengths to the overall number of loss episodes. Simulation with the *Star Wars* trace, n sources, and buffer size $B = 100$ KB. This figure is a zoomed version of Figure 5.10 depicting on a linear scale only the values for loss episodes of length 1, 2 and 3.

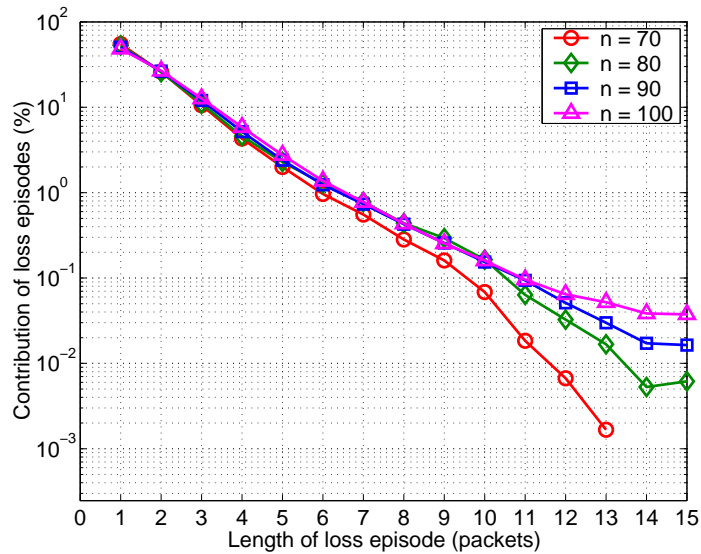


Figure 5.12: Contribution of loss episodes of various lengths to the overall number of loss episodes. Simulation with the *Talk show* trace, n sources, and buffer size $B = 50$ KB.

congestion exists in the network.

5.3.1 Loss rates

Table 5.2 shows the loss rates at the router buffer for the scenario with TCP Reno sources for various values of the buffer size. The packet size during the simulations was fixed to 552 bytes. For comparison, the last column of Table 5.2 show the same statistic for a scenario with only UDP sources using packets of size 552 bytes. We separated the TCP simulations into so called “trace-driven” and FTP.

We use the term “trace-driven” to refer to our modification in the sources’ algorithm for sending packets. We used this approach in order to perform a direct comparison with the UDP results. However, we should note that sending of the packets from each source conforms primarily to the TCP protocol. This means that the source will “try” to send the packets according to the traffic trace, but will never violate the TCP algorithm. For example, if *cwnd* packets are sent and no acknowledgment is received, the source will refrain from sending further packets even though the trace might suggest that the next packet should be sent by the source. On the

$n = 100$ pktsize = 552 bytes	<i>TCP simulations</i>		<i>UDP simulations</i>
	“trace-driven”	FTP	
Buffer size (KB)	Loss rates		
25	$2.96 \cdot 10^{-1}$	4.58	13.2
50	$3.33 \cdot 10^{-3}$	4.14	8.41
75	0	3.47	4.50
100	0	3.68	1.60

Table 5.2: Loss rates for the TCP simulations (“trace-driven” and FTP) and UDP simulations with the *Star Wars* trace.

other hand, if the source is allowed to send further packets according to the TCP protocol, the traffic trace will be checked to see if that particular packet should be sent immediately or should be delayed by some time according to the interarrival times table in the traffic trace.

The FTP assumes that the source always has data waiting to be sent to the destination. The sending of data at the source has nothing to do with a traffic trace, but is governed only by the TCP protocol.

We can see from Table 5.2 that for the “trace-driven” simulations, loss occurs only for extremely small buffer sizes. Compared to the UDP transfers, the loss for the “trace-driven” TCP transfers is at least an order of magnitude smaller. The reason for the extreme difference is twofold: first, TCP reacts to the loss at the router buffer by decreasing the congestion window and second, the source might wait to send a packet in order to meet the specified interarrival time given in the traffic trace. This leads to a decrease in the aggregate input traffic at the router and consequently a decrease in the aggregate loss.

On the other hand, the loss rate for FTP transfers decreases slowly with the increase in buffer size. In the case of FTP transfers, the sources always have data to send. A situation where a source waits to send the data (because the trace might specify so) does not happen in the case of FTP. It should be mentioned that even though the loss rate for FTP is higher, more packets are sent by the sources and more packets reach the destination. The throughput measured at the link between the router and the destination (the throughput of successfully received packets), is higher for FTP than for the “trace-driven” transfers. For a buffer size of 25 KB, the

throughput for FTP is 44.548 Mbps (99% of the link capacity), while the throughput for the “trace-driven” transfers is 38.794 Mbps (87% of the link capacity).

5.3.2 Loss episodes

The TCP transfers are characterized by short loss episodes. Figure 5.13 presents the lengths of the loss episodes from a single source with the corresponding instances when these loss episodes started. For better visibility, only the loss episodes between 250 and 300 sec are shown. Figure 5.13 suggests that most of the loss episodes are consisted of just one or two packets. Indeed, for the particular example with $n = 120$ sources and $B = 100$ KB, loss episodes with only one packet (single losses) constitute over 90% of all the loss episodes. Increasing the number of TCP sources shows similar distribution of the loss episode lengths. Table 5.3 shows the contribution of loss episodes of length 1 to 3.

n	$k = 1$	$k = 2$	$k = 3$
120	91.6%	7.14%	1.10%
140	93.1%	6.09%	0.75%
160	93.8%	5.48%	0.03%

Table 5.3: Contribution of loss episodes of various lengths to the overall number of loss episodes. The number of sources = n , the buffer size $B = 100$ KB.

The reason for the shorter loss episodes is the congestion window $cwnd$, which limits the burst of packets from a single source. Figure 5.14 shows the decrease in the congestion window upon occurrence of loss. The crosses in the Figure represent the length of the loss episodes in packets.

Each source contributes to the behavior of the aggregate traffic entering the router buffer. If we consider the consecutive losses that appear at the router buffer, we notice that the lengths of the loss episodes are also shorter than in the UDP scenario. The contribution of loss episodes for a buffer size $B = 50$ KB is shown in Figure 5.15.

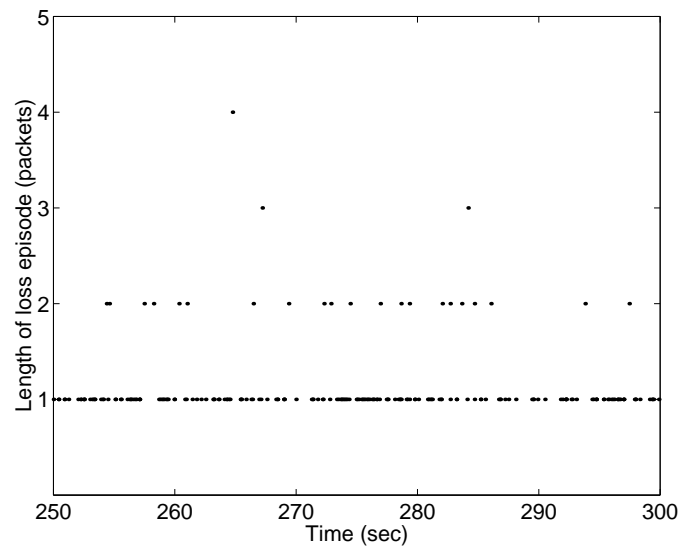


Figure 5.13: Length of loss episodes for a single TCP source vs. time. The number of sources in the simulation is $n = 120$.

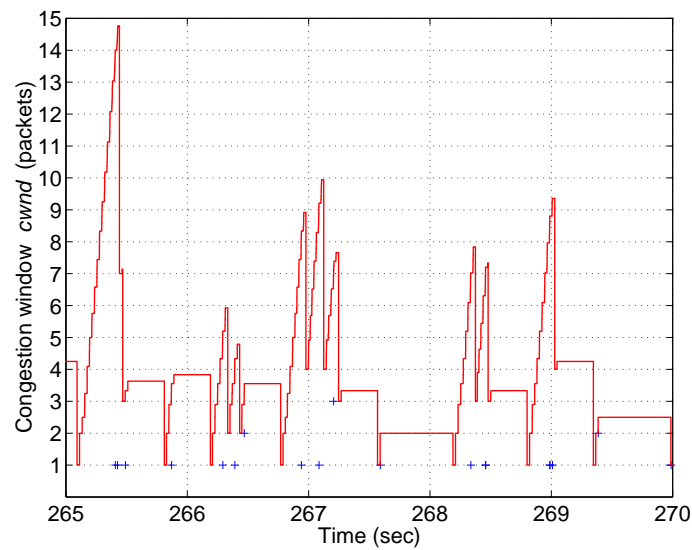


Figure 5.14: Congestion window of a single TCP source vs. time. The number of sources in the simulations is $n = 120$.

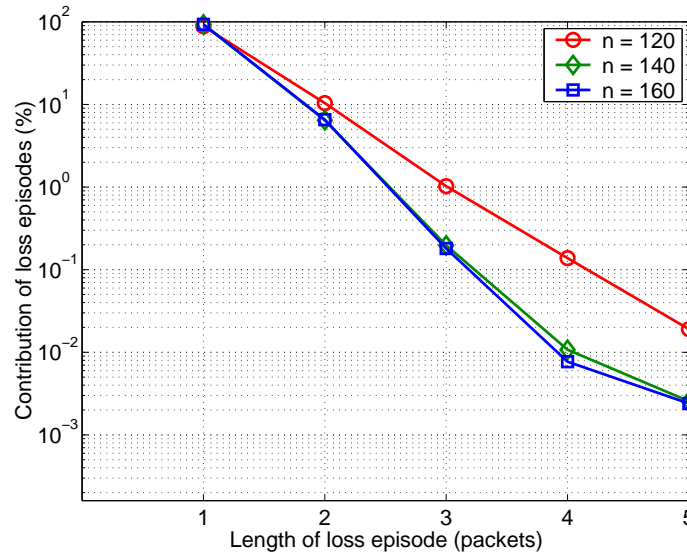


Figure 5.15: Contribution of loss episodes of various lengths to the overall number of loss episodes. TCP sources, buffer size $B = 50$ KB.

5.4 Simple topology scenario - mixed UDP/TCP sources

Comparing Figure 5.15 with Figure 5.8, we can see that the contribution of loss episodes of length k ($k \geq 2$) is much smaller when TCP transfer is used. However, one should be careful with the comparison of the performance parameters for the UDP and TCP scenarios. Two important issues should be taken into account when comparing the influence of the protocols on the contribution of loss episodes of a certain length k . First of all, in the analysis of loss episodes, we did not consider packet delay from the source to the destination. Because of the TCP retransmission scheme, the average end-to-end delay is increased. Second, in the UDP scenario, we observed the loss of a particular UDP flow and the loss at the router buffer in a network where all other sources were using UDP as a transfer protocol. This means that the other sources were sending packets with no respect to the current state in the network. However, in the TCP scenario, all of the sources used the TCP algorithm to regulate their sending of packets depending on the successful or unsuccessful delivery of the previous packets. Therefore, direct comparison of the loss rates and the contribution

of loss episodes for a UDP flow in the UDP scenario and TCP flow in the TCP scenario can be misleading. In the first case, the UDP flow is sharing the network link with other greedy UDP flows, while in the second case, the TCP flow is sharing the link with other TCP flows, which adapt their rate depending on the state in the network.

Therefore, the next step is analysis of per-flow and aggregate loss in a scenario of mixed UDP and TCP sources. We compare the loss rates and the loss episodes for a single UDP and TCP flow that share the link with the other sources in the network.

In the first subscenario we have 80 UDP and 40 TCP sources ($m = 80$ in Figure 5.16) while in the second, we have 40 UDP and 80 TCP sources ($m = 40$).

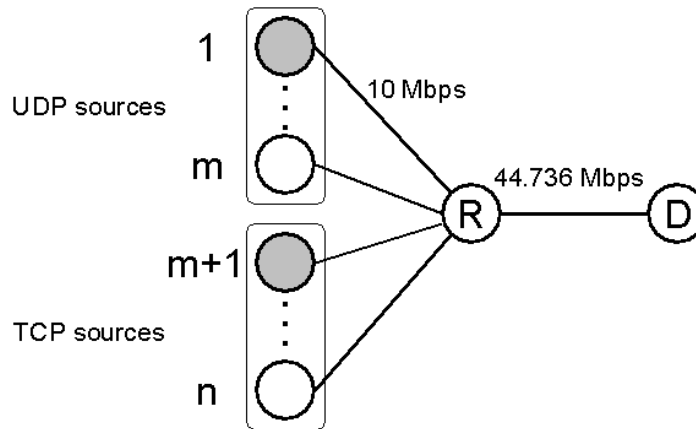


Figure 5.16: Scenario with m UDP and $n - m$ TCP sources. The shaded circles represent one specific UDP and one specific TCP source (numbered 1 and $m+1$, respectively).

5.4.1 Loss rates

We compare the loss rates for particular sources in the mixed scenario. For the case of 80 UDP and 40 TCP sources, the average loss rates for specific sources, are shown in Figure 5.17 (buffer size $B = 50$ KB) and Figure 5.18 (buffer size $B = 100$ KB). For 40 UDP and 80 TCP sources, the average loss rates for specific sources are shown in Figure 5.19 (buffer size $B = 50$ KB) and Figure 5.20 (buffer size $B = 100$ KB). The average loss rate for all the UDP sources, calculated as the ratio of the number of dropped packets over the number of generated packets from all the UDP sources, is

shown with a dashed line. In a same manner, we represent the average loss for all the TCP sources. We can clearly see that the loss rates exhibited by the TCP sources are smaller than the loss rates for the UDP sources. As discussed in the previous section, the difference is a direct result of the flow control mechanism in TCP that is absent from the UDP protocol.

5.4.2 Loss episodes

We first present the contribution of loss episodes of various lengths for a fixed total number of sources $n = 120$. The number of UDP sources m is varied from 20 to 100 (in steps of 20). The number of TCP sources is $n - m$. The general pattern is that the contribution of loss episodes of length k decreases with the increase in the number of UDP sources, and is presented in Figure 5.21. Regardless of the buffer size and the ratio of UDP vs. TCP sources, the average loss rate for the UDP sources is higher than the average loss rate for the TCP sources (see Figures 5.17–5.20).

Now we concentrate on the loss episodes of a particular UDP and TCP flow in a scenario where the flows share the network capacity with the same background traffic. We will choose a UDP flow with a loss rate closest to the median loss rate from all the UDP flows. Similarly, we choose the TCP flow with a loss rate closest to the median loss rate from all the TCP flows. Our example will refer to the scenario of 80 UDP and 40 TCP sources, and the corresponding flows with closest loss rate to the median loss rate is source 23 from the UDP sources and source 106 from the TCP sources. Table 5.4 shows the number of loss episodes of various lengths for the UDP and TCP flow with highest loss rates: It can be seen that both the absolute and the relative

Loss episode length (packets)	1	2	3	4
Number of loss episodes (UDP source 23)	1311	260	93	29
Contribution of loss episodes	77.07%	15.29%	5.47%	1.70%
Number of loss episodes (TCP source 106)	581	66	11	9
Contribution of loss episodes	86.59%	9.84%	1.64%	1.34%

Table 5.4: Number of loss episodes of length k for a specific UDP and TCP flow. Simulation with 80 UDP and 40 TCP sources. The buffer size is $B = 100$ KB.

count of loss episodes longer than 1 packet is greater for the UDP flow. This relates

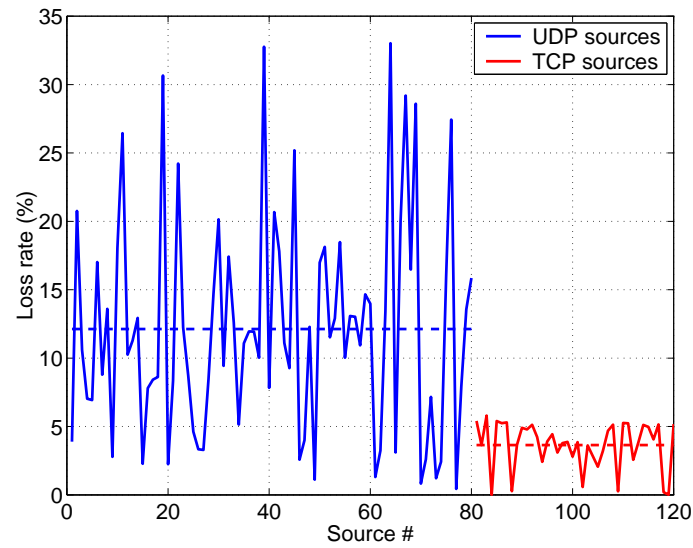


Figure 5.17: Average loss rates for each source. The first 80 sources (numbered from 1 to 80) are using UDP, and the last 40 sources (numbered 81 to 120) are using TCP. The buffer size is $B = 50$ KB.

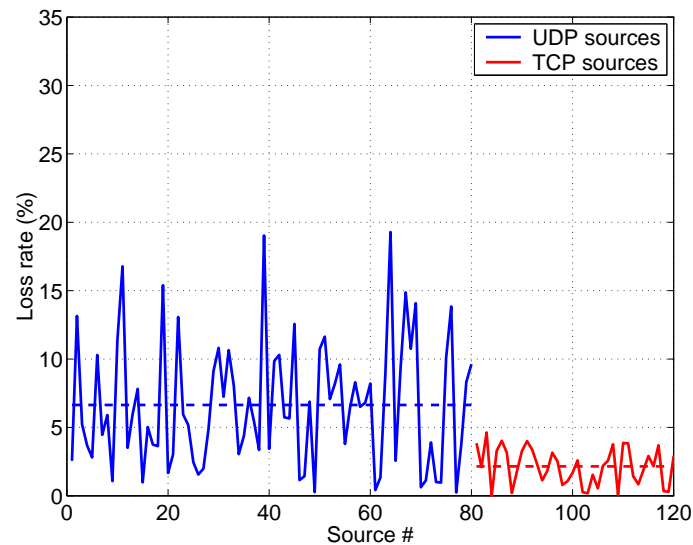


Figure 5.18: Average loss rates for each source. The first 80 sources (numbered from 1 to 80) are using UDP, and the last 40 sources (numbered 81 to 120) are using TCP. The buffer size is $B = 100$ KB.

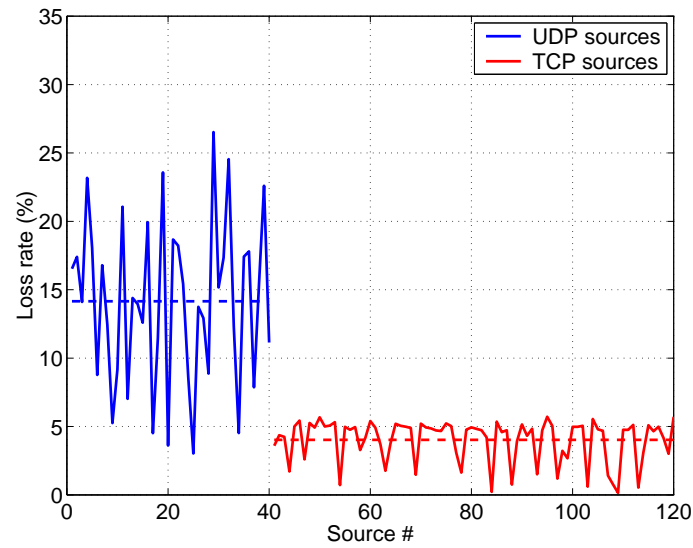


Figure 5.19: Average loss rates for each source. The first 40 sources (numbered from 1 to 40) are using UDP, and the last 80 sources (numbered 41 to 120) are using TCP. The buffer size is $B = 50$ KB.

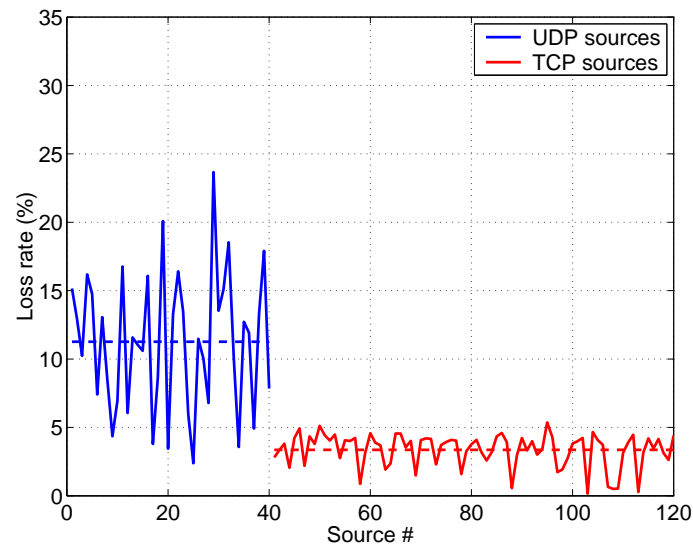


Figure 5.20: Average loss rates for each source. The first 40 sources (numbered from 1 to 40) are using UDP, and the last 80 sources (numbered 41 to 120) are using TCP. The buffer size is $B = 100$ KB.

to the discussion in the previous sections for only UDP and only TCP transfers. In summary, TCP sources adapt to the network conditions, and thus the TCP transfers have lower loss rates and shorter loss episodes than the UDP transfers of the same data.

At this point, one might ask the question why UDP (or better to say Real-Time Transport Protocol [50] which runs over UDP) is predominantly used in the Internet for transfer of video data. The problem is that the TCP retransmission scheme introduces a large delay variance and the decrease of the congestion window upon a loss event¹ has an adverse impact on the quality of the audio/video as perceived by the user. Therefore, a lot of the current networking research is in the area of TCP-friendly protocol development which will combine the requirements for fast data transfer, small delay variation, slowly varying rate, as well as responsiveness to congestion and adaptiveness to changing network conditions.

5.5 Complex topology scenario

Next, we consider the complex topology given in Figure 4.3. The traffic generated by the hosts in this network topology differs from the traffic in the simple topology, where we considered sources generating packets according to a trace. In this scenario, more complex traffic generation occurs, and the sources generate either Web, FTP, or video traffic over UDP.

By applying the specific details and parameters of the traffic generation described in Chapter 4, we observe the loss behavior at the bottleneck link R_3 – R_4 and the loss of a single UDP video flow from node S to node D . The link R_3 – R_4 is assumed to be the only bottleneck in the network. The packets that arrive at the router buffer R_3 and wait to go to the router R_4 might be dropped if the buffer overflows. The contribution of loss episodes for both the per-flow and aggregate loss is depicted in Figure 4.3.

Figure 5.22 presents the contribution of loss episodes of the flow $S \rightarrow D$ and at the router buffer $R_3 \rightarrow R_4$. We should note the similarity between the loss episode pattern for the aggregate loss in Figure 5.22 and, for example, the loss given in Figures 5.8

¹The exact dynamics of the TCP congestion window upon occurrence of loss is specified with IETF Request for Comments 2581 [4].

and 5.10. The same comparison can be made with the per-flow loss results (see for example Figure 5.23). However, there is considerable difference in the simulation environments that produced these figures: both the topology and the variation in types of source traffic. However, even in the more complex scenario, the contribution of loss episodes has the same pattern of an approximately geometric decrease, which can be modeled with the Gilbert models, as shown in Section 5.6.

5.6 Comparison with the Gilbert models

The models presented in Section 2.2 were derived from UDP loss traces obtained from real Internet measurements. Since our approach to gathering loss data is through simulation and not through measurements, a question arises as to whether the loss process generated through our simple simulation scenario has similar characteristics to the loss process derived from a real-world environment.

We compare the probability of having loss episode of various lengths with the probabilities obtained from two models described in Section 2.2: the two-state Markov chain (Gilbert model) and the Extended Gilbert model. These models were applied to loss data obtained by end-to-end measurements, which means loss affecting a single stream of data. Therefore, we perform all the calculations for loss data from a single traffic source, and not for the aggregate loss at the router buffer. We present an example for a UDP source in the scenario described in Section 5.2 (UDP sources only).

For the simulation with the *Star Wars* trace, $n = 100$ sources and buffer size $B = 50$ KB, the input data needed to compute the parameters for the models are given in Table 5.5.

a is the total number of lost and received packets from a particular source (source with number 50 in the example), d is the total number of lost packets for that source, and o_k is the number of loss episodes of length k .

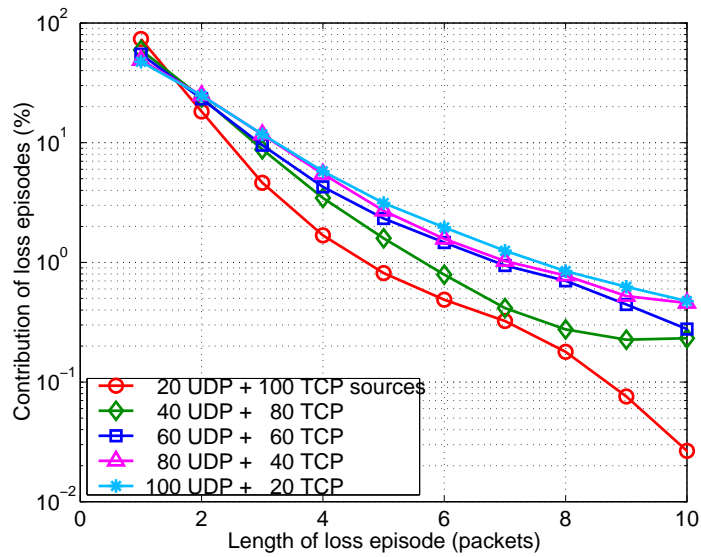


Figure 5.21: Contribution of loss episodes for the mixed UDP/TCP scenario. The total number of sources is fixed ($n = 120$), the number of UDP sources is m and the number of TCP sources is $n - m$. The buffer size is $B = 100$ KB.

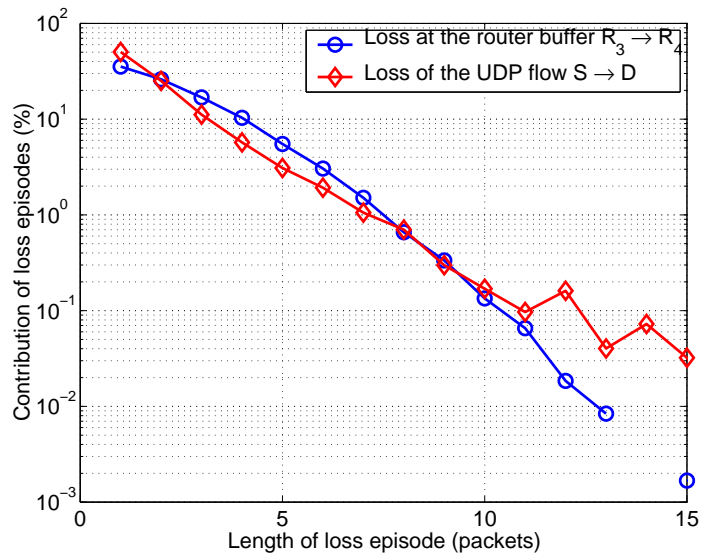


Figure 5.22: Contribution of loss episodes for the router buffer $R_3 \rightarrow R_4$ and a specific UDP video flow $S \rightarrow D$.

Parameter	Value							
a	266,509							
d	47,950							
k	1	2	3	4	5	6	7	8
o_k	15169	6042	2522	1186	608	346	180	99
k	9	10	11	12	13	14	15	16
o_k	49	32	15	7	7	4	2	2

Table 5.5: Input data for the Gilbert models.

Parameters for the Gilbert model

The parameters for the Gilbert model can be obtained by using Equation 2.4. The calculated transition and steady-state matrices for the Gilbert model are given with:

$$\begin{bmatrix} p_{00} & p_{10} \\ p_{01} & p_{11} \end{bmatrix} = \begin{bmatrix} 0.9014 & 0.5479 \\ 0.0986 & 0.4521 \end{bmatrix} \quad (5.2)$$

$$\begin{bmatrix} P(0) \\ P(1) \end{bmatrix} = \begin{bmatrix} 0.8475 \\ 0.1525 \end{bmatrix}. \quad (5.3)$$

The unconditional probability of loss is $ulp = P(1) = 0.1525$. However, the conditional probability of loss is much higher: $clp = p_{11} = 0.4521$. This is in accordance with the result from [13]. This result also confirms the discussion in Section 2.2.1 that loss appears in bursts and that if a packet is lost, there is a high probability that the next packet will be lost as well.

Using the transition probabilities given in Equation 5.2, we can also calculate the probability of having a loss episode of length k (or loss-episode length probability distribution). For $1 \leq k \leq L_{max}$ ($L_{max} = 16$ in the above example) and with the assumption that a loss occurred, this probability is equal to: $P(\text{length} = k) = p_{11}^{k-1} \cdot p_{10}$. Figure 5.23 presents a comparison between the loss-episode length distribution derived from the Gilbert model and the distribution obtained directly from the simulation data. The main result is that the Gilbert model characterizes the loss-episode length accurately for smaller values of k , but underestimates the probabilities of having longer bursts. Due to the fact that our maximum loss-episode length in the

particular simulation run was only 16 packets, we could not explore the behavior of the distribution tail and compare the results with [16] (see Section 2.2.5).

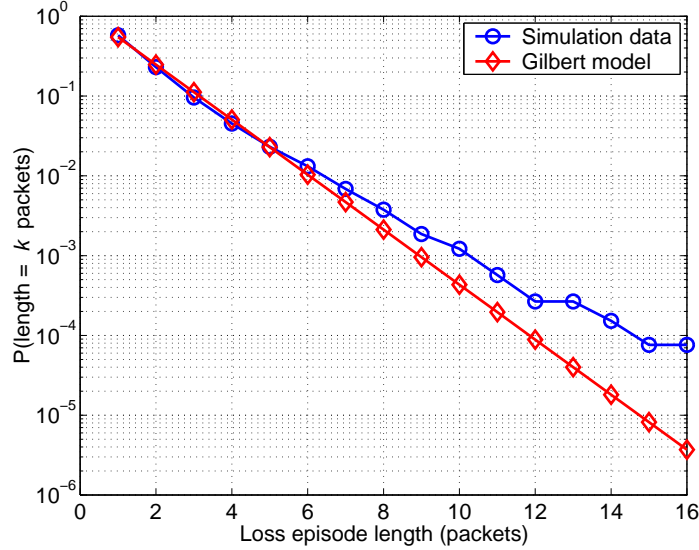


Figure 5.23: Comparison of the loss episode length probabilities obtained from the simulation data with the loss episode length probabilities obtained from the Gilbert model. The number of sources $n = 100$, the buffer size $B = 50$ KB, the packet size is 200 bytes, the *Star Wars* trace is used, and the loss from source number 50 is observed.

Parameters for the Extended Gilbert model

The parameters for the Extended Gilbert model of order m can be obtained by using Equation 2.6. The probability of having a loss episode of length k ($1 \leq k \leq m$) is equal to: $P(\text{length} = k) = p_{12} \cdot p_{23} \cdots p_{(k-1)k} \cdot p_{10}$. Using Equation 2.6, it can be shown that the Extended Gilbert model exactly models the probabilities of having a loss episode of length k for $k < m$:

$$\begin{aligned}
 P(\text{length} = k) &= p_{12} \cdot p_{23} \cdots p_{(k-1)k} \cdot p_{10} = \\
 &= \frac{\sum_{i=2}^{i=L_{max}} o_i}{\sum_{i=1}^{i=L_{max}} o_i} \cdot \frac{\sum_{i=3}^{i=L_{max}} o_i}{\sum_{i=2}^{i=L_{max}} o_i} \cdots \frac{\sum_{i=k}^{i=L_{max}} o_i}{\sum_{i=k-1}^{i=L_{max}} o_i} \cdot \left(1 - \frac{\sum_{i=k+1}^{i=L_{max}} o_i}{\sum_{i=k}^{i=L_{max}} o_i}\right) =
 \end{aligned}$$

$$= \frac{\sum_{i=k}^{i=L_{max}} o_i}{\sum_{i=1}^{i=L_{max}} o_i} \cdot \left(1 - \frac{\sum_{i=k+1}^{i=L_{max}} o_i}{\sum_{i=k}^{i=L_{max}} o_i}\right) = \frac{\sum_{i=k}^{i=L_{max}} o_i - \sum_{i=k+1}^{i=L_{max}} o_i}{\sum_{i=1}^{i=L_{max}} o_i} = \frac{o_k}{\sum_{i=1}^{i=L_{max}} o_i}. \quad (5.4)$$

The last expression is the ratio between the number of loss episodes of length k and the total number of loss episodes, which is exactly the probability of having a loss episode of length k (obtained from the simulation trace). However, for $k = m$, we obtain a different probability than the one calculated directly from the simulated data.

$$P(\text{length} = m) = \frac{\sum_{i=m}^{i=L_{max}} o_i}{\sum_{i=1}^{i=L_{max}} o_i} \cdot \left(1 - \frac{\sum_{i=m}^{L_{max}} (i - m) o_i}{d - m}\right) \neq \frac{o_m}{\sum_{i=1}^{i=L_{max}} o_i}. \quad (5.5)$$

The Extended Gilbert model is easy to implement as a counter of successfully lost packets up to an order of m [31]. When a packet is successfully received, the value of the counter is set to zero. The system should also keep track of the number of loss episodes of length k ($1 \leq k \leq m$) in order to get the values o_k needed for the computation of the transition probabilities.

Chapter 6

Analysis of loss on multiple time scales

The loss data generated through extensive simulations served as a basis for obtaining the results in Chapter 5. The same data were also used for further statistical analysis of the scale-dependent properties of loss. We briefly demonstrate part of this ongoing research for the aggregate UDP and TCP loss.

Using wavelet analysis, our previous study has shown that the aggregate loss during UDP transfers shows long-range dependent characteristics [56]. Long-range dependence exists if there is linear relationship between the \log_2 of the spectrum $\Gamma_x(2^{-j}\nu_0)$ and the scale level j (which corresponds to a time scale of 2^j sec). The graph for the wavelet spectrum of the aggregate UDP loss shown with the \ominus label in Figure 6.1 indicates that the loss process for the UDP video transfers exhibits long-range dependence on the coarser time scales. In our experiments, the breakpoint scale level after which long-range dependence was observed is $j = 10$ which corresponds to a time scale of $2^{10} = 1024$ msec.

To further investigate the long-range behavior beyond scale level $j = 10$, we apply standard graphical techniques for estimating the Hurst parameter, such as the variance-time and R/S plots. The plots are obtained using the statistical package Splus, and the variance-time and R/S tools for long-range dependence analysis [52].

First, a new time process X is formed from the original loss trace. The aggregation (binning) of the new process is performed on a time scale of 1024 msec. Then we plot $\log(\text{var}(X^{(m)}))$ vs. $\log(m)$ and $\log(R(n)/S(n))$ vs. $\log(d)$ (see Section 2.3.2). The

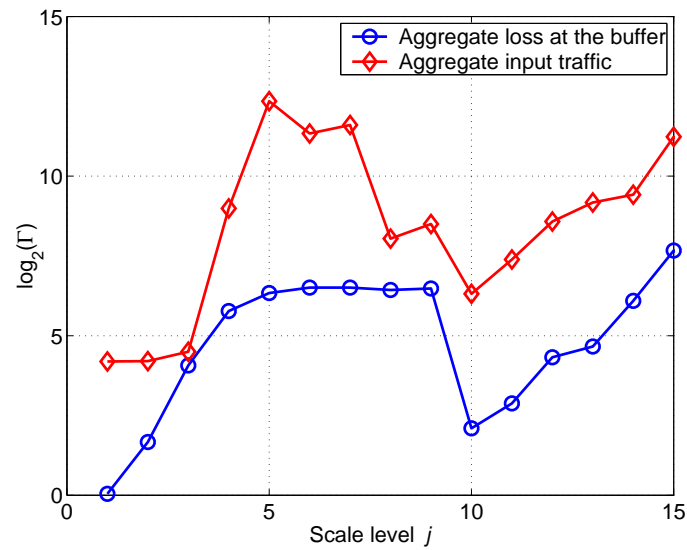


Figure 6.1: The $\log_2(\Gamma)$ vs. j plot of the UDP packet loss process and the aggregate input traffic for buffer size $B = 100$ KB [56].

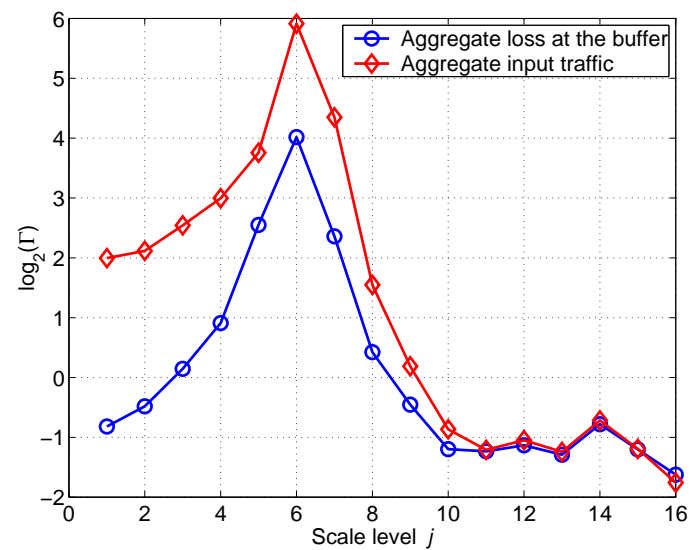


Figure 6.2: The $\log_2(\Gamma)$ vs. j plot of the TCP packet loss process and the aggregate input traffic for buffer size $B = 100$ KB [56].

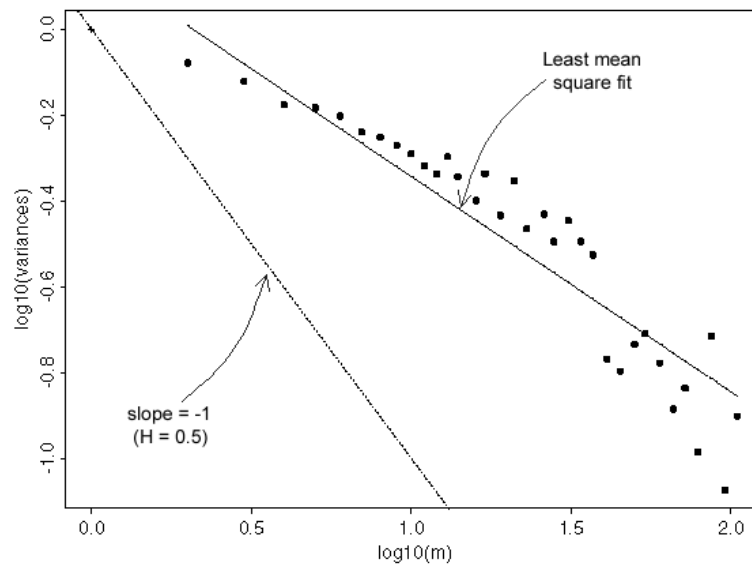


Figure 6.3: Variance-time plot of the UDP loss process obtained through simulation with $n = 100$ sources and buffer size $B = 100$ KB.

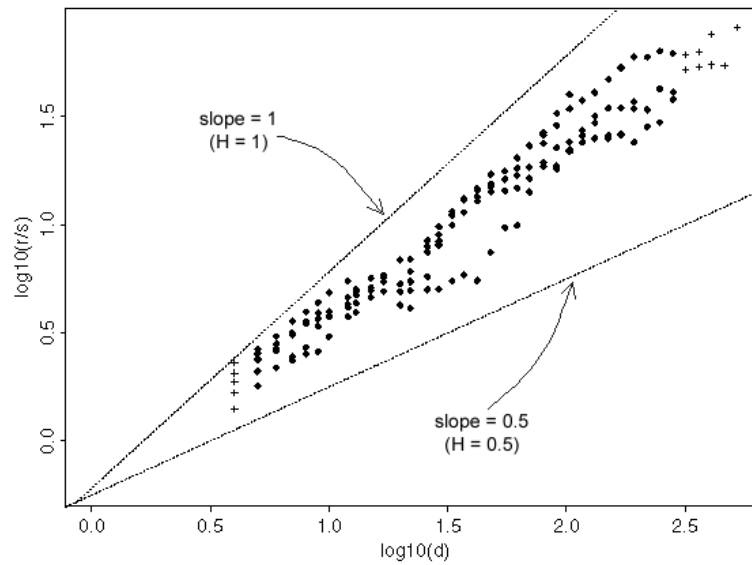


Figure 6.4: R/S plot of the UDP loss process obtained through simulation with $n = 100$ sources and buffer size $B = 100$ KB.

plots are presented in Figures 6.3 and 6.4. The slope in Figure 6.3 is equal to $2H - 2$ and we obtain $\hat{H}_{variance-time} = 0.749$. The estimated Hurst parameter from Figure 6.4 is the value of the fitted least squares line (not shown in the Figure): $\hat{H}_{r/s} = 0.789$. Both estimators of the Hurst parameter have a value between 0.5 and 1, which is an indication of long-range dependence. This is especially visible in Figure 6.4 where the two straight lines correspond to slopes (and accordingly to values of the Hurst parameter) of 0.5 and 1, respectively.

The long-range dependence of the loss process can be related to the characteristics of the traffic feeding the router buffer. The wavelet spectrum $\Gamma_x(2^{-j}\nu_0)$ for the loss process ($-\ominus-$ label in Figure 6.1) has similar pattern to the wavelet spectrum of the aggregate input traffic process ($-\diamond-$ label in Figure 6.1). Analysis of the aggregate input traffic confirms that the input traffic is also long-range dependent and gives the following estimates for the Hurst parameter: $\hat{H}_{variance-time} = 0.862$ and $\hat{H}_{r/s} = 0.871$.

Even when the input traffic is not long-range dependent, which is the case in the TCP scenario, the loss pattern observed on various time scales follows the pattern of the input traffic (see Figure 6.2).

The above results imply that for our simulations with DropTail buffers, the aggregate loss process preserves the scaling properties of the aggregate input traffic process. This brings us back to the field of traffic analysis, because the scaling characteristics of the traffic are reflected on the scaling characteristics of the loss. We must note that the last statement is based on our simulation experiments, and all of them used buffers with DropTail queue management policy. However, we have not investigated the effect of other types of active queue management (for example Random Early Detection gateways) on the relation between the aggregate loss at the buffer and the aggregate input traffic.

Chapter 7

Conclusion and future work

The main purpose of this research is to obtain insight into the loss patterns and the effect that transport protocols have on the loss behavior. In this Chapter, we present a summary of our results and give possible directions for future work in the area of loss analysis.

7.1 Conclusion

We use the term loss behavior to refer not only to the average loss at the router buffers or of a particular end-to-end flow, but also to the time varying pattern of the lost packets. As reported in previous research, and also confirmed in our analysis, lost packets are often grouped in bursts, or loss episodes. Although single losses can be handled, concealed, or corrected by the end user applications, multiple consecutive losses have adverse effect on the transfer of multimedia traffic. We did not report on the user's perception of a multimedia content that suffered from longer loss episodes during the network transfer.

Our analysis of loss was focused on the contribution of the loss episodes at a congested router buffer and of a single end-to-end flow. For the loss of a specific UDP flow, we applied existing loss models such as the Gilbert model and the Extended Gilbert model.

To collect the loss data for our analysis, we used network simulation. The widely adopted network simulator *ns-2* was the main tool used in our work. The unlimited access to all of the simulation data created a flexible environment for loss collection

and analysis.

Two network topologies were simulated: a very simple one, consisting of n sources, one router, and a traffic sink; and a more complex topology, consisting of several transit nodes, each connected to a number of end hosts. In the first topology, video traffic was transferred from the sources to the destination using either UDP or TCP as a transport protocol. In the more complex topology, we used a mix of traffic sources generating Web, FTP, or video traffic, where the scheduling of the generation of Web requests and FTP transfers was based on previous studies on Web and FTP traffic characterizations.

For the UDP transfers, we noticed that the contribution of loss episodes of length one (single losses) decreases with increase of the traffic load. We showed that the increase of the number of sources n (and consequently the traffic load) leads to higher loss rates and lengthier loss episodes.

We further showed that the contribution of the loss episodes, for both the router buffer loss and the per-flow loss, decreases approximately geometrically with increase of the length of the loss episode. It has already been shown through empirical measurements of audio loss that the audio loss episodes have approximately geometrical distribution. By using network simulations, we showed that similar patterns for the loss episodes hold during video transfers. The geometrical distribution of video loss episodes enabled us to use some of the existing loss models such as the Gilbert model and the Extended Gilbert model and to fit the model to the simulation data. Even the simple Gilbert model, consisting of only two states was a good fit to the per-flow loss episode contribution. The matching between the data from the model and the data from the simulation was more accurate for the shorter loss episodes. We did not perform modeling for the upper tail of the loss episode contribution, but we conjecture that it can be successfully modeled with a Pareto distribution, as was previously done for end-to-end loss of audio traffic. However, the maximum length of the observed loss episodes were not long enough for us to explore this assumption in more detail.

The contribution of lengthier loss episodes for TCP sources is in general smaller than in the UDP case. The main reason for the different behavior of the TCP loss is the nature of the TCP sources, which adapt to the changes in the network conditions. The decrease of the congestion window of a particular TCP source upon a detection of lost packets directly reflects on the number of generated packets in one round-trip

time. The maximum length of the loss episodes is short, and for the various buffer sizes and number of sources deployed in the TCP simulations, the loss episodes of length one (single losses) contribute to more than 90% of the total number of loss episodes. Although we have not addressed the effect of the round-trip time on the contribution of loss episodes, we expect that larger round-trip times will imply reduced sending rate of the TCP sources. This, in turn, will imply lower loss rates, and, consequently, loss patterns mainly consisting of single losses.

The difference between the UDP and TCP loss behavior was even more pronounced in the scenario with mixed UDP/TCP sources, where we showed that both the average loss rates and the lengths of loss episodes are smaller for the TCP senders. While the TCP senders are adaptive, the UDP senders generate their packets whenever they have packets to be sent, without considering the current network conditions.

Although the simulations with the more complex topology involved richer traffic composition, the contribution of loss episodes showed a similar pattern to the simulations with simple topology.

7.2 Future work

There are several key areas for extension of our current research.

The first important area is to incorporate the analysis of packet loss with the analysis of packet delay. For interactive streaming applications, if a packet arrives late, it might be useless and from the aspect of the application can be treated as a lost packet even though it was not actually lost in the network. Therefore, the analysis of the delay is necessary for complete analysis of loss and its impact on the interactive multimedia transfer.

The second key area is to see the effect of various queue management policies on the loss patterns, especially the impact on the loss episode lengths and the aggregate loss behavior compared to the aggregate input traffic. Here we especially refer to Random Early Detection routers which might randomly drop packets even before the buffer is full, and where the decision for early drop is based not on the momentary queue size, but on the tracked average queue size.

Another crucial area of future research is to assess the impact of the consecutive losses on the end-user perception. The first step can be performed by mapping the

packet level loss pattern to a loss pattern on a video frame level. Then, by using either existing objective tests or, preferably, subjective tests with a real video stream, the impact of consecutive losses (i.e., loss episodes) can be determined. This is of utmost importance since it will enable relating the end-user perception to the packet level loss, which might provide a reference basis for effective error correction or error concealment techniques at the end hosts.

References

- [1] Star Wars trace. In ASCII format <<ftp://ftp.telcordia.com/pub/vbr.video.trace>>; and in *ns-2* format <<http://www.research.att.com/~breslau/vint/trace.html>> [Accessed 25 July 2000].
- [2] University of Würzburg archive of video traces. In ASCII format: <<http://nero.informatik.uni-wuerzburg.de/MPEG/>> [Accessed 25 July 2000].
- [3] Patrice Abry and Darryl Veitch. Wavelet analysis of long-range-dependent traffic. *IEEE Transactions on Information Theory*, 44(1):2–15, January 1998.
- [4] Mark Allman, Vern Paxson, and W. Richard Stevens. TCP congestion control. Request for Comments 2581, Internet Engineering Task Force, April 1999.
- [5] Guy Almes. IP performance metrics: Metrics, tools, and infrastructure, July 1997. <<http://http://www.advanced.org/surveyor/>> [Accessed 25 July 2000].
- [6] Martin F. Arlitt and Carey L. Williamson. Web server workload characterization: The search for invariants. In *Proceedings of ACM SIGMETRICS '96: Conference on Measurement and Modeling of Computer Systems*, pages 126–137, Philadelphia, PA, USA, May 1996.
- [7] Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, and Sally Floyd. Improving simulation for network research. Technical Report 99-702, University of Southern California, March 1999.
- [8] Hari Balakrishnan, Venkat Padmanabhan, Srinivasan Seshan, and Randy H. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, December 1997.

- [9] Hari Balakrishnan, Hariharan S. Rahul, and Srinivasan Seshan. An integrated congestion management architecture for internet hosts. In *Proceedings of ACM SIGCOMM '99 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 175–187, Cambridge, MA, USA, August 1999.
- [10] Paul Barford and Mark Crovella. Generating representative Web workloads for network and server performance evaluation. In *Proceedings of SIGMETRICS '98/PERFORMANCE '98, Joint International Conference on Measurement and Modeling of Computer Systems*, pages 151–160, Madison, Wisconsin, USA, June 1998.
- [11] Jan Beran. *Statistics for long-memory processes*, chapter 12. Monographs on statistics and applied probability, No. 61. Chapman & Hall, New York, NY, USA, 1994.
- [12] Tim Berners-Lee, Roy T. Fielding, and Henrik Frystyk Nielsen. Hypertext Transfer Protocol – HTTP/1.0. Request for Comments 1945, Internet Engineering Task Force, May 1996.
- [13] Jean-Chrysostome Bolot. End-to-end packet delay and loss behavior in the Internet. In *Proceedings of ACM SIGCOMM '93 Conference on Communications Architectures, Protocols and Applications*, pages 289–298, San Francisco, CA, USA, September 1993.
- [14] Jean-Chrysostome Bolot, Sacha Fosse-Parisis, and Don Towsley. Adaptive FEC-based error control for Internet telephony. In *Proceedings of IEEE INFOCOM*, pages 1453–1460, New York, NY, USA, March 1999.
- [15] Jean-Chrysostome Bolot, Thierry Turletti, and Ian Wakeman. Scalable feedback control for multicast video distribution in the Internet. In *Proceedings of ACM SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, pages 58–67, London, UK, August 1994.
- [16] Michael S. Borella and Debbie Swider. Internet packet loss: Measurement and implications for end-to-end QoS. In *Proceedings of the 1998 ICPP workshops on*

- architectural and OS support for multimedia applications/flexible communication systems/wireless networks and mobile computing*, pages 3–12, Minneapolis, MN, USA, August 1998.
- [17] Ramón Cáceres and Liviu Iftode. Improving the performance of reliable transport protocols in mobile computing environments. *IEEE Journal on Selected Areas in Communications*, 13(5):850–857, June 1995.
- [18] Vinton G. Cerf. The Net on Mars in 2003. *NewsLink - Alcatel's International Magazine*, 7(2), 1999. <<http://www.alcatel.com/telecom/mbd/publi/newslink/9902/interview.htm>> [Accessed 25 July 2000].
- [19] David R. Cox. Long-range dependence: A review. In H. A. David and H. T. David, editors, *Statistics: An appraisal*, pages 55–74. The Iowa State University Press, Ames, IA, USA, 1984.
- [20] Mark E. Crovella and Azer Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, December 1997.
- [21] Christophe Diot, Christian Huitema, and Thierry Turetli. Multimedia applications should be adaptive. In *Proceedings of third IEEE workshop on the architecture and implementation of high performance communication subsystems(HPCS)*, pages 117–125, Mystic, CT, USA, August 1995.
- [22] Anja Feldmann, Anna C. Gilbert, Polly Huang, and Walter Willinger. Dynamics of IP traffic: A study of the role of variability and the impact of control. In *Proceedings of ACM SIGCOMM '99 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 301–313, Cambridge, MA, USA, August 1999.
- [23] Anja Feldmann, Anna C. Gilbert, and Walter Willinger. Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic. In *Proceedings of ACM SIGCOMM '98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 42–55, Vancouver, BC, CANADA, August 1998.

- [24] Mordechai Fester. Performance issues for high-end video over ATM, August 1995. <http://www.cisco.com/warp/public/cc/sol/mkt/ent/atm/vidat_wp.htm> [Accessed 25 July 2000].
- [25] Roy T. Fielding, Jim Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, and Tim Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. Request for Comments 2068, Internet Engineering Task Force, January 1997.
- [26] Mark Garrett and Walter Willinger. Analysis, modeling and generation of self-similar VBR video traffic. In *Proceedings of ACM SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications*, pages 269–280, London, UK, August 1994.
- [27] Internet Software Consortium. Web resources, 2000. <<http://www.isc.org/>> [Accessed 25 July 2000].
- [28] Van Jacobson. Congestion avoidance and control. In *SIGCOMM '88, Proceedings of the ACM Symposium on Communications Architectures and Protocols*, pages 314–329, Stanford, CA, USA, August 1988.
- [29] Van Jacobson, Craig Leres, and Steven McCanne. `tcpdump`, 1989. <<ftp://ftp.ee.lbl.gov/tcpdump.tar.Z>> [Accessed 25 July 2000].
- [30] Raj Jain. Congestion control in computer networks: Issues and trends. *IEEE Network*, 4(3):24–30, May 1990.
- [31] Wenyu Jiang and Henning Schulzrinne. QoS measurement of Internet real-time multimedia services. Technical Report CUCS-015-99, Department of Computer Science, Columbia University, December 1999.
- [32] Gunnar Karlsson. Asynchronous transfer of video. *IEEE Communications Magazine*, 34(8):118–126, August 1996.
- [33] Srinivasan Keshav and Rosen Sharma. Issues and trends in router design. *IEEE Communications Magazine*, 36(5):144–151, May 1998.

- [34] Steven M. Klivansky, Amarnath Mukherjee, and Cheng Song. On long-range dependence in NSFNET traffic. In *Proceedings of 7th IEEE LAN/MAN Workshop*, Marathon, FL, USA, March 1995.
- [35] Rajeev Koodli and Rayadurgam Ravikanth. One-way loss pattern sample metrics. Internet-draft, Internet Engineering Task Force, July 2000. Expiration Date: December, 2000.
- [36] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, February 1994.
- [37] Benoit B. Mandelbrot and Murad S. Taqqu. Robust R/S analysis of long run serial correlation. In *Proceedings of 42nd session of the International Statistical Institute*, pages 1–37, Manila, Philippines, December 1979.
- [38] Velibor Markovski and Ljiljana Trajković. Analysis of loss episodes for video transfer over UDP. In *Proceedings of the 2000 SCS symposium on performance evaluation of computer and telecommunication Systems (SPECTS'2K)*, pages 278–285, Vancouver, BC, Canada, July 2000. Society for Computer Simulation International (SCS).
- [39] Steven McCanne and Van Jacobson. The BSD packet filter: A new architecture for user-level packet capture. In *Proceedings of Usenix Winter Conference*, pages 259–269, San Diego, CA, USA, January 1993. Usenix.
- [40] John B. Nagle. On packet switches with infinite storage. *IEEE Transactions on Communications*, COM-35(4):435–438, April 1987.
- [41] Vern Paxson. Empirically-derived analytic models of wide-area TCP connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336, August 1994.
- [42] Vern Paxson. *Measurements and analysis of end-to-end Internet dynamics*. PhD thesis, University of California, Berkeley, Berkeley, CA, USA, April 1997.
- [43] Vern Paxson, Andrew Adams, and Matt Mathis. Experiences with NIMI. In *Proceedings of Passive & Active Measurement: PAM-2000*, Hamilton, New Zealand, April 2000.

- [44] Vern Paxson and Sally Floyd. Wide-area traffic: The failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.
- [45] Vern Paxson and Sally Floyd. Why we don't know how to simulate the Internet. In *Proceedings of the 1997 Winter Simulation Conference*, pages 1037–1044, Atlanta, GA, USA, December 1997.
- [46] Vern Paxson, Jamshid Mahdavi, Andrew Adams, and Matt Mathis. An architecture for large-scale Internet measurement. *IEEE Communications Magazine*, 36(8):48–54, August 1998.
- [47] Johnathan Postel and Joyce Reynolds. File Transfer Protocol (FTP). Request for Comments 959, Internet Engineering Task Force, 1985.
- [48] Oliver Rose. Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems. In *Proceedings of the 20th Annual Conference on Local Computer Networks*, pages 397–406, Minneapolis, MN, USA, October 1995.
- [49] Henning Sanneck and Georg Carle. A framework model for packet loss metrics based on loss runlengths. In *Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference 2000 (MMCN 2000)*, pages 177–187, San Jose, CA, USA, January 2000.
- [50] Henning Schulzrinne, Stephen L. Casner, Ron Frederick, and Van Jacobson. RTP: A Transport Protocol for Real-Time Applications. Request for Comments 1889, Internet Engineering Task Force, 1996.
- [51] Deborah S. Swayne, Di Cook, and Andreas Buja. XGobi: Interactive dynamic data visualization in the X window system. *Journal of Computational and Graphical Statistics*, 7(1):113–130, March 1998.
- [52] Murad Taqqu, Bob Sherman, Walter Willinger, and Vadim Teverovsky. Statistical methods for long-range dependence - Splus source code: <<http://math.bu.edu/INDIVIDUAL/murad/methods/index.html>> [Accessed 25 July 2000].

- [53] Kevin Thompson, Gregory Miller, and Rick Wilder. Wide-area Internet traffic patterns and characteristics. *IEEE Network*, 11(6):10–23, November 1997.
- [54] VINT (Virtual InterNetwork Testbed), 1996. <<http://www.isi.edu/nsnam/vint/>> [Accessed 25 July 2000].
- [55] Walter Willinger, Murad S. Taqqu, Robert Sherman, and Daniel V. Wilson. Self-similarity through high-variability: Statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, February 1997.
- [56] Fei Xue, Velibor Markovski, and Ljiljana Trajković. Wavelet analysis of packet loss in video transfers over UDP. In *Proceedings of the First International Conference on Internet Computing - IC'2000*, pages 427–433, Las Vegas, NV, USA, June 2000.
- [57] Maya Yajnik, Jim Kurose, and Don Towsley. Packet loss correlation in the MBone multicast network. In *Proceedings of the IEEE Conference on Global Communications (GLOBECOM)*, pages 94–99, London, UK, November 1996.
- [58] Maya Yajnik, Sue Moon Jim Kurose, and Don Towsley. Measurement and modelling of the temporal dependence in packet loss. In *IEEE INFOCOM*, pages 345–352, New York, NY, USA, March 1999.