

SIMULATION AND ANALYSIS OF QUALITY OF SERVICE PARAMETERS IN IP NETWORKS WITH VIDEO TRAFFIC

by

Bruce Chen

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

© Bruce Chen 2002
SIMON FRASER UNIVERSITY
May 06, 2002

Approval

Name: Bruce Chen

Degree: Bachelor of Applied Science

Title of Thesis: Simulation and Analysis of Quality of Service Parameters in IP Networks with Video Traffic

Dr. John Jones

Director
School of Engineering Science, SFU

Examining Committee:

**Technical and
Academic Supervisor:**

Dr. Ljiljana Trajkovic
Associate Professor
School of Engineering Science, SFU

Committee Member:

Dr. Stephen Hardy

Professor
School of Engineering Science, SFU

Committee Member:

Dr. Jacques Vaisey

Associate Professor
School of Engineering Science, SFU

Date Approved: _____

Abstract

The main objective of this research is to simulate and analyze the quality of service (QoS) in Internet Protocol (IP) networks with video traffic. We use network simulator ns-2 to simulate networks and their behaviors in the presence of video traffic. The video traffic is generated by genuine MPEG-1 video traces transmitted over the User Datagram Protocol (UDP). The selected MPEG-1 video traces exhibit medium to high degrees of self-similarity, and we are interested in how the video traffic affects the characteristics of QoS in the network. The main QoS parameters of interest are packet loss due to buffer overflow and packet delay due to queuing in the network router. Our analysis focuses on the simulation scenario where the router employs a FIFO buffer with a DropTail queue management policy. We analyze the simulation results using statistical approaches. We characterize the packet loss pattern using loss episodes, which define consecutively lost packets. We analyze the packet delay patterns using packet delay distribution and the autocorrelation function. In addition to the FIFO/DropTail simulation scenario, we also perform preliminary investigations on how various queuing mechanisms affect the characteristics of QoS parameters. We simulate buffers employing Random Early Drop (RED), Fair Queuing (FQ), Stochastic Fair Queuing (SFQ), and Deficit Round Robin (DRR). Our preliminary studies compare the QoS characteristics influenced by these queuing mechanisms as well as the IP service fairness of these queuing mechanisms.

Table of Contents

Approval	ii
Abstract.....	iii
List of Figures.....	vi
List of Tables.....	xi
1. Introduction.....	1
2 Background and related work	3
2.1. QoS parameters in the Internet and video traffic	3
2.2. Overview of MPEG	4
2.2.1. MPEG.....	4
2.2.2. MPEG-1	6
2.2.3. MPEG-2	6
2.2.4. MPEG-4	6
2.3. Delivery of MPEG over IP.....	7
2.4. MPEG-1 and MPEG-2 over RTP/UDP/IP.....	8
2.5. Background on self-similar processes	11
2.5.1. Definition of self-similar processes	12
2.5.2. Traits and impacts of self-similar processes	14
3 Simulation methodology	17
3.1. Network simulator ns-2.....	17
3.2. Simulation traces.....	18
3.3. Simulation configuration and parameters	22
4 QoS Parameters	26
4.1. Loss	26
4.2. Delay.....	30
4.3. Other QoS and network performance parameters.....	31
5 Scheduling and queue management schemes.....	33
5.1. FIFO/DropTail	33
5.2. Random early drop (RED)	33
5.3. Fair Queuing (FQ).....	34

5.4. Stochastic fair queuing (SFQ).....	35
5.5. Deficit round robin (DRR).....	35
6 FIFO/DropTail simulation results and analysis.....	37
6.1. Comparison of packetization methods and the addition of RTP header.....	37
6.2. Packet loss.....	40
6.2.1. Aggregate packet loss	40
6.2.2. Effects of traffic load on aggregate packet loss	45
6.2.3. Per-flow packet loss.....	48
6.3. Packet delay	50
6.3.1. Packet delay distribution.....	50
6.3.2. Packet delay autocorrelation function.....	52
6.3.3. Packet delay jitter.....	56
6.3.4. Per-flow average packet delay and standard deviation.....	58
6.4. QoS and network performance parameters	59
6.4.1. Buffer occupancy probability.....	59
6.4.2. Per-flow traffic load, throughput, and loss rate	61
7 RED, FQ, SFQ, and DRR simulation.....	62
7.1. RED.....	62
7.2. FQ, SFQ, and DRR	69
8 Conclusions and future work	81
Appendix A. Aggregate packet loss process	83
Appendix B. Effect of traffic load on aggregate packet loss	88
Appendix C. Contribution of loss episodes for each individual flow.....	91
Appendix D. Additional simulation results	102
References.....	108

List of Figures

Figure 2.1.	Example of MPEG-1 GoP pattern and dependency	5
Figure 2.2.	RTP packet format and its encapsulation into UDP/IP packet	11
Figure 2.3.	Self-similar and Poisson traffic in different time scales	16
Figure 3.1.	Packetization of MPEG traces	22
Figure 3.2.	Network topology for the simulation.....	23
Figure 4.1.	Illustration of loss episodes and loss distance.....	27
Figure 4.2.	Illustration of per-flow and aggregate packet loss.....	28
Figure 5.1.	Common structure of FQ, SFQ, and DRR.....	36
Figure 6.1.	Contribution of loss episodes of various lengths to the overall number of loss episodes. Simulation with different packetization methods with or without RTP headers, using the Terminator trace.	39
Figure 6.2.	Contribution of loss episodes of various lengths to the overall number of loss episodes, linear and log scale.	42
Figure 6.3.	Contribution of lost packet from loss episodes of i to the overall number of lost packets, linear and log scale.	43
Figure 6.4.	Distribution of packet arrival process measured in one-millisecond intervals, linear (top) and log (bottom) scale.	44
Figure 6.5.	Aggregate packet loss process.	45
Figure 6.6.	Contribution of loss episodes of various lengths to the overall number of loss episodes, entire episode length range and episode length up to 22 packets.....	47
Figure 6.7.	Contribution of loss episodes of various lengths to the overall number of loss episodes, episode length from 1 and to 3.	48

Figure 6.8.	The contribution of loss episodes of various lengths to overall number of loss episodes, averaged over all individual flows (per-flow loss), linear scale and log scale.....	49
Figure 6.9.	Probability distribution of packet delay for all delivered packets, linear scale and log scale.....	51
Figure 6.10.	Autocorrelation function for packet delays. Top: packet no. 1 to 200,000. Middle: packet no. 1,000,000 to 1,200,000. Bottom: packet no. 2,000,000 to 2,200,000. Left: 2000 lag scale. Right: 100 lag scale.	53
Figure 6.11.	Autocorrelation function for packet delays. Top: packet no. 4,000,000 to 4,200,000. Middle: packet no. 6,000,000 to 6,200,000. Bottom: packet no. 8,000,000 to 8,200,000. Left: 2000 lag scale. Right: 100 lag scale.....	54
Figure 6.12.	Autocorrelation function for packet delays. Top: packet no. 10,000,000 to 10,200,000. Middle: packet no. 12,000,000 to 12,200,000. Bottom: packet no. 14,000,000 to 14,200,000. Left: 2000 lag scale. Right: 100 lag scale.....	55
Figure 6.13.	Distribution of the magnitude of packet delay jitter, linear scale and log scale.....	57
Figure 6.14.	Average packet delay for packets from the same flow.	58
Figure 6.15.	Standard deviation of packet delay for packets from the same flow.	59
Figure 6.16.	Router buffer occupancy probability distribution, linear scale and log scale.	60
Figure 6.17.	Per-flow load, throughput, and loss, calculated with respect to total traffic load, throughput, and packet loss.	61
Figure 7.1.	The contribution of loss episodes of various lengths to overall number of loss episodes, averaged over all individual flows (per-flow loss), linear scale and log scale. Simulation with FIFO/DropTail and RED.	64
Figure 7.2.	The length of the longest loss episode for each flow. Simulation with FIFO/DropTail and RED.	65

Figure 7.3.	Probability distribution of packet delay for all delivered packets, linear scale and log scale. Simulation with FIFO/DropTail and RED.	66
Figure 7.4.	Average packet delay for packets from the same flow. Simulation with FIFO/DropTail and RED.	68
Figure 7.5.	Standard deviation of packet delay for packets from the same flow. Simulation with FIFO/DropTail and RED.	68
Figure 7.6.	Per-flow load, throughput, and loss, calculated with respect to total traffic load, throughput, and packet loss. Simulation with RED.	69
Figure 7.7.	The contribution of loss episodes of various lengths to overall number of loss episodes, averaged over all individual flows (per-flow loss), linear scale and log scale. Simulation with FQ, SFQ, and DRR.	72
Figure 7.8.	The length of the longest loss episode for each flow. Simulation with FQ, SFQ, and DRR.	74
Figure 7.9.	Probability distribution of packet delay for all delivered packets, linear scale and log scale. Simulation with FQ, SFQ, and DRR.	75
Figure 7.10.	The length of the longest packet delay for each flow. Simulation with FQ, SFQ, and DRR.	77
Figure 7.11.	Average packet delay for packets from the same flow. Simulation with FQ, SFQ, and RED.	77
Figure 7.12.	Standard deviation of packet delay for packets from the same flow. Simulation with FQ, SFQ, and DRR.	79
Figure 7.13.	Per-flow load, throughput, and loss, calculated with respect to total traffic load, throughput, and packet loss. Simulation with FQ.	79
Figure 7.14.	Per-flow load, throughput, and loss, calculated with respect to total traffic load, throughput, and packet loss. Simulation with SFQ.	80
Figure 7.15.	Per-flow load, throughput, and loss, calculated with respect to total traffic load, throughput, and packet loss. Simulation with DRR.	80
Figure A1.	Aggregate packet loss process. Simulation with 100 traffic sources, using one MPEG trace (Terminator 2).....	84

Figure A2.	Aggregate packet loss process. Simulation with 100 traffic sources, using one MPEG trace (Simpsons).	85
Figure A3.	Aggregate packet loss process. Simulation with 100 traffic sources, using one MPEG trace (Jurassic Park 1).....	86
Figure A4.	Aggregate packet loss process. Simulation with 100 traffic sources, using one MPEG trace (Star Wars).....	87
Figure B1.	Contribution of loss episodes of various lengths to the overall number of loss episodes, linear and log scale. Simulation with 40 to 100 traffic sources, using Garrett’s Star Wars MPEG traces.	89
Figure B2.	Contribution of loss episodes of various lengths to the overall number of loss episodes, episode length up to 22 packets. Simulation with 40 to 100 traffic sources, using Garrett’s Star Wars MPEG traces.	90
Figure B3.	Contribution of loss episodes of various lengths to the overall number of loss episodes, episode length from 1 and to 3. Simulation with 40 to 100 traffic sources, using Garrett’s Star Wars MPEG traces.	90
Figure C1.	The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 1 to 10.	92
Figure C2.	The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 11 to 20.	93
Figure C3.	The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 21 to 30.	94
Figure C4.	The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 31 to 40.	95
Figure C5.	The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 41 to 50.	96
Figure C6.	The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 51 to 60.	97
Figure C7.	The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 61 to 70.	98

Figure C8.	The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 71 to 80.	99
Figure C9.	The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 81 to 90.	100
Figure C10.	The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 91 to 100.	101
Figure D1.	Distribution of the magnitude of packet delay jitter, linear scale and log scale. Simulation with FIFO/DropTail.	103
Figure D2.	Average packet delay for packets from the same flow. Simulation with FIFO/DropTail.	104
Figure D3.	Standard deviation of packet delay for packets from the same flow. Simulation with FIFO/DropTail.	104
Figure D4.	Per-flow load, throughput, and loss. Simulation with FIFO/DropTail.	105
Figure D5.	The length of the longest loss episode for each flow. Simulation with FIFO/DropTail.	105
Figure D6.	Average packet delay for packets from the same flow. Simulation with FIFO/DropTail and RED.	106
Figure D7.	Standard deviation of packet delay for packets from the same flow. Simulation with FIFO/DropTail and RED.	106
Figure D8.	Per-flow load, throughput, and loss, calculated with respect to the total traffic load, throughput, and packet loss. Simulation with RED.	107

List of Tables

Table 3.1.	MPEG-1 traces from University of Wurzburg.	19
Table 3.2.	MPEG trace for each traffic source.	24
Table 4.1.	Example of per-flow packet loss contribution calculation.	30
Table 6.1.	Summary of simulation results for the comparison of the effect of different packetization methods and RTP header addition.	38
Table 6.2.	Summary of simulation results for the FIFO/DropTail simulation with 100 traffic sources.	40
Table 6.3.	Summary of simulation results for the FIFO/DropTail simulation with various numbers of traffic sources.	46
Table 7.1.	Summary of simulation results for the FIFO/DropTail and RED simulation with 100 traffic sources.	63
Table 7.2.	Summary of simulation results for the FQ, SFQ, and DRR simulation with 100 traffic sources.	71
Table A1.	Summary of simulation results for the FIFO/DropTail simulation with various single MPEG traces.	83
Table B1.	Summary of statistics of Garrett’s MPEG-1 Star Wars trace.	88

Chapter 1

Introduction

The rapid expansion of the Internet in recent years has significantly changed the characteristics of its data traffic. As user demand increases and the deployment of the broadband network expands, the amount of data traffic has reached an unprecedented level. Among various types of data traffic, video data plays an important role in today's broadband networks. Today's high-speed broadband networks enable video applications over the Internet; video streaming, and video conferencing are common examples of applications that deliver real-time video content over the Internet.

One of the most important concepts related to the service offered by the data or voice network is the quality of service (QoS). QoS refers to the capability of a network to provide better service to data traffic over various network technologies. Some of the primary goals of QoS are guaranteed bandwidth, controlled delay variation and latency, and improved loss characteristics [32]. Unlike the traditional circuit-switching network where the QoS of telephone calls is predetermined, most of the Internet is still a best-effort network based on packet-switching. A best-effort network such as the Internet does not guarantee any particular performance bound and, therefore, QoS must be measured and monitored in order to maintain the performance of the network and the service to the applications [21]. In order to ensure the quality of the delivered video and its consistency, real-time video applications have particularly stringent QoS requirements, such as loss, delay, and delay jitter [20], [27]. Thus, the understanding of the characteristics of the video data traffic and its impact on QoS parameters are critical to improving network congestion management for video data traffic [33].

The main objective of this research is to simulate and analyze QoS of video traffic in Internet Protocol (IP) networks based on the work in [5], [26], and [52]. We use network simulator ns-2 to simulate networks and their behaviors in the presence of video traffic. We characterize the video traffic and its QoS parameters through statistical analysis of

the simulation data. The majority of this research focuses on the network router with a first-in-first-out (FIFO) scheduling and a DropTail queue management scheme. In addition, we also present the simulation of the effect of different scheduling schemes and queue management schemes on the video traffic and its QoS parameters. At the end of this thesis project, we hope that having a better insight into the QoS parameter characteristics of video traffic could help to improve the design of network management tools for better QoS support.

Chapter 2 provides an overview of the research and work related to QoS parameters in the Internet and video traffic. An overview of the MPEG video format and the delivery of MPEG over IP networks is also presented, followed by background on the statistical properties of self-similarity of video data traffic. Chapter 3 explains our simulation approach, introducing network simulator ns-2 and the MPEG video traces used for the simulation. A detailed description of the simulation parameters is also given. Chapter 4 explains our approach to the analysis of various QoS parameters, while Chapter 5 describes the functionalities and different scheduling algorithm and queue management schemes employed in this research. Chapter 6 presents the simulation and analysis results for the FIFO/DropTail simulation scenario, and Chapter 7 presents the simulation and analysis results for simulation scenarios with different scheduling and queue management schemes. Chapter 8 gives the conclusion and provides directions for future work.

Chapter 2

Background and related work

2.1. QoS parameters in the Internet and video traffic

Because of the recent increase in video data traffic and its sensitivity to loss, delay, and delay jitter in networks, network designers are beginning to understand the importance of QoS and network congestion management [33]. The main task network designers are facing is to design buffer management tools that minimize packet loss, delay, and delay variation (jitter). Good design of network management tools requires good understanding of traffic. Thus, accurate modeling of the traffic is the first step in optimizing resource allocation algorithms, so that the provision of network service complies with the QoS requirements while maintaining the maximum network capacity. The model of the network traffic and its influence on the network are critical to providing high QoS [38].

For video as well as other Internet traffic, the characteristics of the traffic are significantly different from the traditional traffic model used for the telephone networks. Many studies have shown that video and Internet traffic possess a complex correlation and exhibit long-range dependence (LRD) that are absent in the Poisson traffic model traditionally used in the telephone networks [12], [25], [51]. Qualitatively, the traditional Poisson model has no memory of the past and, thus, it is inadequate to accurately model LRD in video traffic. The failure of the Poisson model may result in the underestimation of the traffic burstiness, which may have a detrimental impact on network performance, including larger queuing delay and packet loss rate [17], [38]. Because of the LRD in video and Internet traffic [17], [25], [38], it is important to determine the network resources necessary to transport the LRD traffic reliably and with appropriate QoS support. One way to analyze and characterize video traffic and its QoS parameters is through computer simulations.

Several past studies presented in [5], [26], and [52] used computer simulation to analyze and characterize the packet loss behavior in IP networks. They employed the Star Wars MPEG trace [17], [42], [43] to generate video traffic transported using the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) in congested networks. They observed the packet loss pattern and its connection to the LRD of the video traffic. They also considered the Random Early Drop (RED) queue management scheme in addition to the FIFO/DropTail queue. We use similar methodology and we aim to extend the results in [5] and [26].

2.2. Overview of MPEG

This section gives an overview of MPEG (Moving Pictures Experts Group) multimedia system, which is the application layer in the simulation of IP networks in our research. MPEG is one of the most widely used compressed video formats. The video traffic in our simulation is generated from the transmission of MPEG-1 video.

2.2.1. MPEG

MPEG (Moving Pictures Experts Group) is a group of researchers who meet under the ISO (International Standards Organization) to generate standards for digital video (sequences of images in time) and audio compression. In particular, they defined a compressed bit stream, which implicitly defines a de-compressor [4]. MPEG achieves high video compression by using two main compression techniques [11]:

- Intra-frame compression: Compression within individual frames (also known as spatial compression because the compression is applied along the image dimensions).
- Inter-frame compression: Compression between frames (also known as temporal compression because the compression is applied along the time dimension).

The intra-frame compression is performed by transforms and entropy coding. The inter-frame compression is performed by prediction of future frames based on the motion vector. This is achieved using three types of frames:

- I-frames are Intra-frame coded frames that need no additional information for decoding.
- P-frames are forward predicted from an earlier frame with the addition of motion compensation. The earlier frame could be an I or a P-frame.
- B-frames are bi-directionally predicted from earlier or later I or P-frames.

Typically I-frames are the largest in size, P-frames are roughly one-half of the size of I-frames, and B-frames are roughly one quarter of the size of I-frames. I, B, and P-frames are arranged in a deterministic periodic sequence. This sequence is called the Group of Picture (GoP) whose length is flexible, but 12 and 15 frames are common values [1]. The overall sequence of frames and GoPs is called the elementary stream, which is the core of the MPEG video. Figure 2.1 illustrates an example of MPEG GoP and the relationships between different frame types.

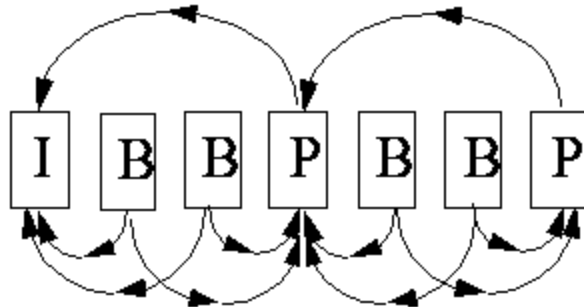


Figure 2.1. Example of MPEG-1 GoP pattern and dependency [4].

Because of the inter-frame compression, MPEG data can exhibit high correlation and burstiness. In addition to GoP and the frame structure, each frame is composed of one or more slices. Each slice is an independently decodable unit. The slice structure is intended to allow decoding in the presence of errors (due to corrupted or lost slices or frames) [4].

2.2.2. MPEG-1

MPEG-1 is an ISO/IEC (International Standard Organization/International Electrotechnical Commission) standard for medium quality and medium bit rate video and audio compression. It allows video to be compressed by the ratios in the range of 50:1 to 100:1, depending on image sequence type and desired quality. The MPEG-1 encoded data rate is optimized for a bandwidth of 1.5 Mbps, which is the audio and video transfer rate of a double-speed CD-ROM player. VHS-quality playback is available from this level of compression. MPEG-1 is one of the most often used video formats on the Web and in video CDs [4], [35].

2.2.3. MPEG-2

MPEG also established the MPEG-2 standard for high-quality video playback at higher data rates between 1.5 and 6 Mbps. MPEG-2 is a superset of MPEG-1 intended for services such as video-on-demand, DVD (digital video disc), digital TV, and HDTV (high definition television) broadcasts. MPEG-2 achieves a higher compression with 20% coding efficiency over MPEG-1. Different from MPEG-1, MPEG-2 allows layered coding. MPEG-2 video sequence is composed of a base layer, which contains the most important video data, and of one or more enhancement layers used to improve video quality [4], [35], [36].

2.2.4. MPEG-4

MPEG-4 is a more powerful compression algorithm, with multimedia access tools to facilitate indexing, downloading, and querying. Its efficient video coding allows MPEG-4 to scale data rates from as low as 64 Kbps to a data rate with quality beyond HDTV. MPEG-4 uses object-based coding which is different from the frame-based coding used in MPEG-1 and MPEG-2. Each video scene is composed of video objects rather than image frames. Each video object may have several scalable layers called video object layers (one base layer and one or more enhance layers). Each video object

layer is composed of an ordered sequence of snapshots in time called video object planes. Video object planes are analogous to I/P/B-frames in MPEG-1 and MPEG-2 standards [15], [50].

MPEG-4 is designed for a wide variety of networks with widely varying performance characteristics. A three-layer system standard for MPEG-4 was developed to help MPEG-4 interface and adapt to the characteristics of different networks. The synchronization layer adds the timing and synchronization information for the coded media. The flexible multiplex layer multiplexes the content the coded media. And the transport multiplex layer interfaces the coded media to the network environment. This three-layer system makes MPEG-4 more versatile and robust than the MPEG-1 and MPEG-2 system [3].

2.3. Delivery of MPEG over IP

For video streaming and real-time applications, most commercial systems use the User Datagram Protocol (UDP) as the transport layer protocol [24], [47]. UDP is suitable for video stream and real-time applications because it has lower delay and overhead compared to the Transmission Control Protocol (TCP). Because UDP is a connectionless protocol, it does not need to establish connection before sending packets as compared to TCP's three-way handshaking for connection setup. Furthermore, because UDP has no flow control mechanism, it can send packets it receives without any delay. The absence of connection and flow control allow UDP to achieve lower delay than TCP [6], [39]. However, UDP is not a reliable packet transport service and, thus, the UDP receiver is not guaranteed to receive all packets. Nevertheless, as long as the packet loss is not too severe, UDP is still the ideal protocol for real-time applications, because not all data is critical as the new data overrides the old data in real-time applications [47]. Furthermore, because UDP does not transfer packets along a fixed path, its pure datagram service nature uses multiple paths to relay data from the source to the destination, helping to

reduce the effect of a single bottleneck in the network and to improve the overall delay [37].

To overcome UDP's shortcoming of not providing any QoS information or guarantees when transferring real-time data, a new transport layer protocol called the Real-Time Transport Protocol (RTP) was developed and specified in RFC 1889 in 1996 [40]. Applications run RTP on top of UDP and use UDP's multiplexing and checksum services. Although RTP does not provide reliable data delivery, flow/congestion control, QoS guarantees, or resource reservation, it provides the basic functionality needed for carrying real-time data over packet network. RTP services include payload type identification, sequence numbering, time stamping, and delivery monitoring. The sequence numbers allow receivers to reconstruct the packet sequence and determine the packet location without actual decoding of packets in the sequence. Time-stamps can be utilized by the sender and the receiver for synchronization and the delay variation calculation. RTP also includes a supplementary protocol, Real-Time Transport Control Protocol (RTCP), to periodically transmit control packets to all participants in the RTP session. RTCP monitors the QoS and conveys information about the participants in an on-going session. The feedback on QoS of the data distribution, such as the fault diagnosis from the receivers, may be useful for adaptive encoding (for example, rate control, slice management, and packetization).

2.4. MPEG-1 and MPEG-2 over RTP/UDP/IP

One of the most important applications of RTP is the transmission of MPEG data over RTP/UDP/IP. RFC 2250, developed in 1998, specifies the RTP payload format and the transmission guidelines for the MPEG-1 and the MPEG-2 video [18].

RTP transfers data in sessions. Each RTP session, which is a pair of destination transport addresses (IP address plus port number), transfers a single RTP payload type (pure video or pure audio). Each RTP header is 16 bytes long, containing information such as sequence numbers and time-stamps. RTP has no restriction on the payload size.

The payload size is limited by the underlying protocols. For the MPEG data, RTP introduces an additional MPEG video-specific header of 4 bytes long.

MPEG-1 multimedia data has three parts: system, video, and audio. Video and audio are specified in the elementary stream format. The system stream is the encapsulation of the elementary stream with presentation time, decoding time, clock reference, and the multiplexing of multiple streams.

MPEG-2 multimedia data has three stream types: elementary, program stream, and transport stream. The elementary stream is similar to that of MPEG-1. The program stream is used in storage media such as DVDs. The transport stream is used for transmission of MPEG-2 such as digital cable TV.

RTP does not specify any encapsulation and packetization guidelines for MPEG-1 system stream and the MPEG-2 program stream. The MPEG-2 transport stream can be encapsulated into RTP packets by packing the 188 byte MPEG-2 transport packets into RTP packets. Multiple MPEG-2 transport packets can be encapsulated into one RTP packet for overhead reduction. The time-stamp in the RTP header records the transmission time for the first byte of the RTP packet. The MPEG-2 transport stream uses the RTP time-stamp for synchronization between the sender and receiver, and the delay variation calculation (time-stamps not used by the MPEG decoder).

Encapsulation of the MPEG-1 and the MPEG-2 elementary streams requires the separation of video and audio. Video and audio streams are encapsulated separately and transferred by separate RTP sessions because an RTP session carries only a single medium type. The payload type field in the RTP header identifies the medium type (video or audio). Because the MPEG-1 and the MPEG-2 type identification information is embedded in the elementary header, RTP does not need to supply additional information. Different from the MPEG-2 transport stream, the time-stamp in the RTP header for the MPEG-1 and the MPEG-2 elementary streams records the presentation time for the video or audio frames. RTP packets corresponding to the same audio or video frame have the same time-stamp. But the time-stamp may not increase

monotonically because when pictures are presented in the order IBBP they will be transmitted in the order IPBB [1].

The encapsulation of the MPEG-1 and the MPEG-2 elementary streams for RTP requires packetization. The packetization method for the video data needs to abide by the following guidelines.

- When the video sequence, GoP, and picture headers are present, they are always placed at the beginning of the RTP payload.
- The beginning of each slice must be placed at the beginning of the payload (after the video sequence, GoP, and picture header if present) or after an integer multiple of slices.
- Each elementary stream header must be completely contained in the RTP packet. The video frame type (I/P/B-frames) is specified in the picture type field of MPEG specific header.

This encapsulation scheme ensures that the beginning of a slice can be found if previous packets are lost (the beginning of a slice is required to start decoding). Slices can be fragmented as long as these rules are satisfied. The beginning and the end of slice bits in the RTP header are used to indicate when a slice is fragmented into multiple RTP packets. When an RTP packet is lost (as indicated by a gap in the RTP sequence number), the receiver may discard all packets until the beginning of the slice bit is set so that the decoder can start to successfully decode the next slice. In our simulation, we follow an MPEG video packetization method similar to the RTP MPEG video packetization method. However, RTP is not used for the MPEG transmission. UDP is the protocol we use, although some of our simulations include the RTP header. More details about our simulation configuration are discussed in Chapter 3. Figure 2.2 illustrates the packet format and the encapsulation of an RTP packet transmitted over UDP/IP.

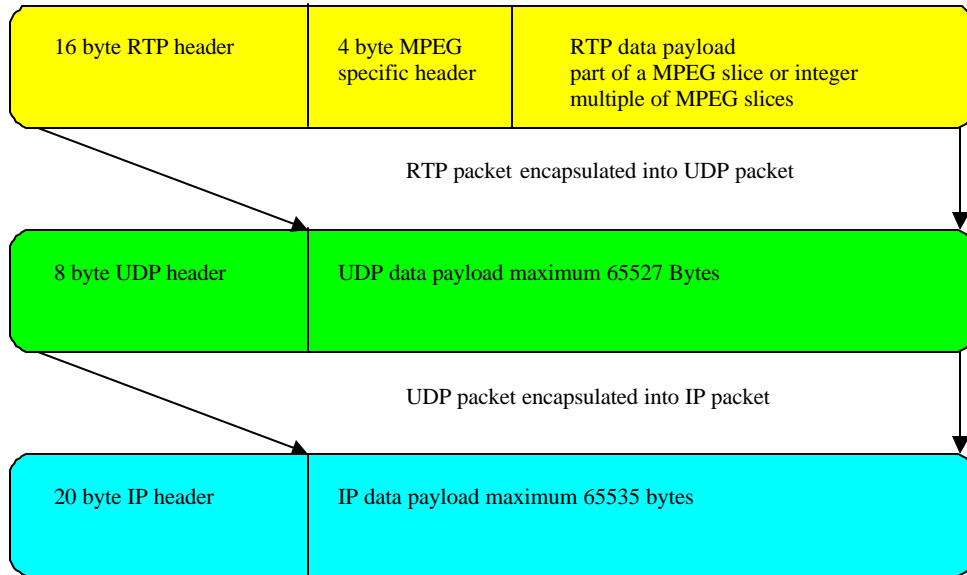


Figure 2.2. RTP packet format and its encapsulation into UDP/IP packet.

2.5. Background on self-similar processes

The rapid change and expansion in the data network in recent years has created a significant impact on network traffic modeling. The Poisson model traditionally used for voice traffic in telephone networks can no longer sufficiently model today's complex and diverse data traffic. One of the most important findings in data traffic engineering is that traffic in local area networks (such as Ethernet) and wide area networks exhibits long-range dependence (LRD) and self-similar properties [9], [12], [13], [22], [25]. LRD and self-similarity have also been found in variable bit rate video traffic [17]. One of the main focuses of our study is to examine how LRD and self-similarity in video traffic affect the characteristics of QoS parameters. In this section, we use the definitions from [34] to provide some basic theoretical background about LRD and self-similarity, and describe their distinctive characteristics and impacts on the data network.

2.5.1. Definition of self-similar processes

The aggregate process $X^m(i)$ of a stationary stochastic process $X(t)$ is defined as

$$X^m(i) = \frac{1}{m} \sum_{t=m(i-1)+1}^{mi} X(t) \quad i = 0, 1, \dots \quad (2.1)$$

$X(t)$ is an exact second-order self-similar stochastic process with Hurst parameter H ($0.5 < H < 1$) if

$$\mathbf{g}(k) = \frac{\mathbf{S}^2}{2} ((k+1)^{2H} - 2k^{2H} + (k-1)^{2H}) \quad (2.2)$$

for $k \geq 1$, and for all m , and where $\gamma(k)$ is the autocovariance function of $X^m(t)$.

$X(t)$ is asymptotically second-order self-similar with Hurst parameter H ($0.5 < H < 1$) if

$$\lim_{m \rightarrow \infty} \mathbf{g}^m(k) = \frac{\mathbf{S}^2}{2} ((k+1)^{2H} - 2k^{2H} + (k-1)^{2H}). \quad (2.3)$$

It can be shown that Eq. (2.2) implies that $\gamma(k) = \gamma^m(k)$ for all $m \geq 1$. Thus second-order self-similarity implies that the correlation structure of $X(t)$ is exactly or asymptotically preserved under aggregation $X^m(i)$. The Hurst parameter H is used as a measure of the degree of self-similarity. The higher the H , the higher the degree of self-similarity. The form $\mathbf{g}(k) = \frac{\mathbf{S}^2}{2} ((k+1)^{2H} - 2k^{2H} + (k-1)^{2H})$ implies further structure of the stochastic process - the long-range dependence of $X(t)$.

For a second-order self-similar process, let $r(k) = \gamma(k)/\sigma^2$ denote the autocorrelation function. For $0 < H < 1$ and $H \neq 0.5$, it holds

$$r(k) \sim H \cdot (2H - 1) \cdot k^{2H-2}, \quad k \rightarrow \infty. \quad (2.4)$$

In particular, if $0.5 < H < 1$, for $0 < \mathbf{b} < 1$, $r(k)$ asymptotically becomes

$$r(k) = ck^{-\mathbf{b}} \quad (2.5)$$

where $c > 0$ is a constant and $\mathbf{b} = 2-2H$.

This also results in

$$\sum_{k=-\infty}^{\infty} r(k) = \infty. \quad (2.6)$$

Eq. (2.5) and (2.6) imply that the autocorrelation function of a self-similar process decays slowly and hyperbolically, which makes it non-summable. When $r(k)$ decays hyperbolically and its summation is unbounded, the corresponding stationary process $X(t)$ is long-range dependent. $X(t)$ is short-range dependent (SRD) if its autocorrelation function is summable. Note that the Poisson model is an example of the short-range dependent stochastic process.

If $H = 0.5$, then $r(k) = 0$ and $X(t)$ is SRD because it is completely uncorrelated. For $0 < H < 0.5$, the summation of $r(k)$ is 0, an artificial condition rarely occurring in SRD applications. If $H = 1$, then $r(k) = 1$, an uninteresting case where $X(t)$ is always perfectly correlated. $H > 1$ is prohibited because of the stationarity condition on $X(t)$.

Although self-similarity does not imply long-range dependence (LRD) and LRD does not imply self-similarity, in the case of asymptotic second-order self-similarity with the restriction $0.5 < H < 1$, self-similarity implies LRD and LRD implies self-similarity. Thus self-similarity and LRD are equivalent in this context. LRD and self-similarity are used interchangeably in the rest of this thesis [34].

There is also a close relationship between the heavy-tailed distribution and LRD. A random variable Z has a heavy-tailed distribution if

$$\Pr\{Z > x\} \sim cx^{-\alpha}, \quad x \rightarrow \infty \quad (2.7)$$

where $0 < \alpha < 2$ is called the tail index or the shape parameter and c is a positive constant. Heavy-tailed distribution is when the tail of a distribution asymptotically, decays hyperbolically. In contrast, light-tailed distribution, such as the exponential and the Gaussian distributions have exponentially decreasing tails. The distinguishing mark of the heavy-tailed distribution is that it has infinite variance for $0 < \alpha < 2$. If $0 < \alpha \leq 1$, it also has an unbounded mean. In the networking context, we are primarily interested in the case $1 < \alpha < 2$. The main characteristic of a random variable obeying a heavy-tailed

distribution is that it exhibits extreme variability. The convergence rate of the sample mean to the population mean is very slow due to this extreme variability in the samples.

The heavy-tailed distribution is the root of LRD and self-similarity; data bursts in data network may exhibit the heavy-tailed distribution and result in LRD and self-similarity. Although heavy-tailness is not necessary to generate LDR in aggregate traffic, empirical measurements provide strong evidence that heavy-tailness is an essential component to induce LRD in network traffic.

2.5.2. Traits and impacts of self-similar processes

As mentioned earlier, self-similarity and LRD are present in most network traffic. The existence of self-similarity and LRD has a significant impact on the network traffic modeling and the network performance. The combination of a large number of independent ON/OFF sources with the heavy-tailed distribution leads to self-similarity in the aggregate process with no reduction in burstiness or correlation [34]. Self-similar data traffic looks statistically similar over a wide range of time-scales, and thus burstiness can appear in all time-scales.

In contrast, in the Poisson and the Markovian traffic model, the probability of rare events (such as an occurrence of a very long data burst) is exponentially small and the stochastic process is SRD, characterized by exponentially decaying autocorrelation ($r(k) = \mathbf{r}^k$, $0 < \mathbf{r} < 1$). As a result, they underestimate the burstiness of traffic and aggregate traffic tends to smooth out [46]. When the traffic process is rescaled in time, the resulting coarsified process rapidly loses dependency. Thus, burstiness occurs mostly in the small time-scale only.

Figure 2.3 is an example of self-similar video traffic (traffic of the Star Wars MPEG video) and synthetic Poisson traffic to illustrate their difference in various time-scales. The four figures on the left are the self-similar MPEG traffic used in [5] and [26] over various time scales, and the four figures on the right are synthesized Poisson traffic with

the same mean. Self-similarity is manifested in MPEG traffic because it looks statistically similar over various time-scales. Moreover, burstiness is apparent in all time-scales for self-similar traffic but lost in the coarsified time-scales for Poisson traffic.

Self-similarity and LRD can have detrimental impacts on the network performance; one immediate impact is the degradation of the queuing performance. Congestion caused by self-similar traffic can build up more than that of SRD traffic in the Poisson model. Modest buffer sizes in the Poisson models cannot effectively absorb the long data burst in self-similar traffic; buffer sizes based on the Poisson model could result in overly optimistic QoS guarantees. Therefore, the presence of the self-similarity of traffic cannot be overlooked in the modeling and analysis of the network performance.

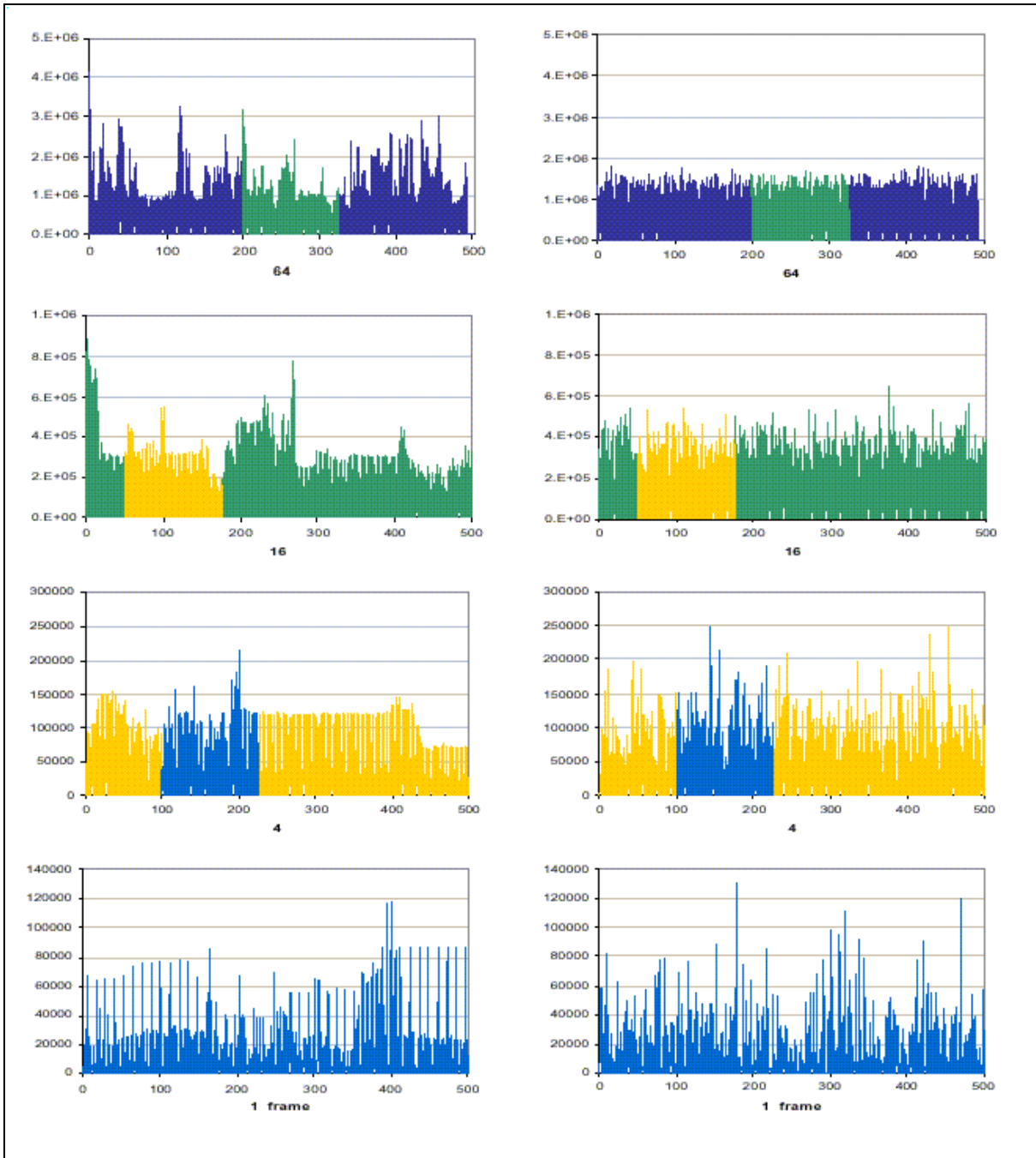


Figure 2.3. Self-similar (left) and Poisson traffic (right) in different time scales. The vertical axis is the number of packets and the horizontal is the time unit in number of frames. The two top plots start with a time-scale of 64 frames. Each subsequent plot is derived from the previous one by randomly choosing an interval of a quarter of its time-scale. These plots are taken from [19] and they were first used in [25].

Chapter 3

Simulation methodology

In this thesis project, the majority of work focuses on simulation. In this chapter, we describe the simulator we used, our simulation methods, and the simulation parameters.

3.1. Network simulator ns-2

We use network simulator ns-2 to simulate IP networks with video traffic. ns-2 is a packet-level, discrete event simulator, widely adopted in the network research community [31]. It evolved from the VINT (Virtual InterNet Testbed) project, a collaborative project among Lawrence Berkeley National Laboratory, University of California, Berkeley, University of South California, and Xerox PARC [49]. It is intended to provide a common reference and test suit for the analysis and development of Internet protocols based on simulation.

Using simulation to analyze data networks has several key advantages, i.e. simulation allows complete access of test data, which is often difficult in the real physical networks. Simulation also provides great flexibility in controlling the parameters in the analysis. In the real physical networks, many parameters of interests are difficult or impossible to control. For example, the time-stamp for every event that happens to a packet is difficult to obtain and the transmission speed of the router is difficult to control.

The ns-2 simulator is written in an object-oriented code using C++ and Object Tcl (OTcl). The C++ part enables high-performance simulation in the packet level and the OTcl part enables flexible simulation configuration and control. This combined structure compromises the complexity and speed of the simulator.

To run a simulation, ns-2 required an OTcl script that specifies the configuration and control of the simulation. A typical ns-2 OTcl script specifies the network topology, network technologies, protocols, applications that generate traffic, and the sequence of events to be executed during the simulation. The simulation results can be viewed graphically as animation using Network Animator (NAM) or stored in files as traces that include the data of interests collected during the simulation. Trace data are the events recorded during the simulation. The following is an example of the simulation trace recording the events occurred at a particular network node during the simulation.

```

+ 0.007594 0 101 udp 552 ----- 0 15 101.14 3 155
+ 0.007594 0 101 udp 552 ----- 0 86 101.85 3 156
D 0.007594 0 101 udp 552 ----- 0 86 101.85 3 156
- 0.007625 0 101 udp 552 ----- 0 51 101.5 1 53
R 0.00769 0 101 udp 552 ----- 0 90 101.89 0 2
- 0.007724 0 101 udp 552 ----- 0 87 101.86 1 54
R 0.007788 0 101 udp 552 ----- 0 21 101.2 0 3

```

There are twelve columns in this trace. The first column indicates the type of the event: a packet is enqueued (+), a packet is dequeued (-), a packet is dropped (D), or a packet is received by the next node (R). The second column is the time-stamp of the event. The third and fourth column are the two nodes between which the trace occurs. The fifth column indicates the type of the packet such as UDP and TCP. The sixth column indicates the size of the packet in bytes. The seventh column contains flags. The eighth column gives the IP flow identifier as defined in IP version 6 (IPv6). The ninth column indicates the source address of the packet. The tenth column indicates the destination address of the packet. The eleventh column gives the packet sequence number. The last column gives the unique packet id.

3.2. Simulation traces

In order to generate video traffic with self-similarity as in the real data network, we use genuine MPEG-1 traces in the simulation. Currently, there are two main sources of

MPEG traces available for the public on the Internet. Researchers at Institute of Computer Science in University of Wurzburg created an archive of MPEG-1 video traces (elementary streams) in 1995 [36], [48]. Researchers at Telecommunication Networks Group in Technical University Berlin created an archive of MPEG-4 video traces (elementary streams) in 2000 [15], [29]. We use ten different MPEG-1 traces from University of Wurzburg in the simulation. Each of these MPEG-1 traces contains the size of each frame in the MPEG-1 video. These MPEG-1 traces represent the video data sent by the application layer. For the transmission over IP networks, they need to be later converted into IP packets in the lower layers. All ten traces have the following properties:

Properties of MPEG-1 Traces from University of Wurzburg [36]:

- One slice per frame
- 25 frames per second
- GoP Pattern: IBBPBBPBBPBB (12 frames)
- Encoder Input: 384×288 pels with 12 bit color resolution
- Number of frames in each trace: 40000 (about half an hour of video)

Table 3.1 is a summary of the basic statistics of these ten traces.

Table 3.1. MPEG-1 traces from University of Wurzburg [36].

Trace	Mean Frame Size (1000 Bytes)	Mean Bit Rate (Mbps)	Peak Bit Rate (Mbps)	Hurst Parameter
The Silence of the Lambs	0.914	0.18	0.85	0.89
Terminator 2	1.363	0.27	0.74	0.89
MTV	2.472	0.49	2.71	0.89
Simpsons	2.322	0.46	1.49	0.89
German Talk Show	1.817	0.36	1.00	0.89
Jurassic Park I	1.634	0.33	1.01	0.88
Mr. Bean	2.205	0.44	1.76	0.85
German News	1.919	0.38	2.23	0.79
Star Wars	1.949	0.36	4.24	0.74
Political Discussion	2.239	0.49	1.40	0.73

As shown in Table 3.1, these ten traces are movies and TV programs. They have medium to high degrees of self-similarity according to their Hurst parameter values. The ratio of peak bit rate to mean bit rate indicates the burstiness of these traces. The average of the mean bit rates of these traces is 0.376 Mbps.

To transmit these videos over IP networks, they need to be packetized. We use a packetization method similar to the guidelines for RTP/UDP/IP packetization mentioned in Chapter 2.4. Here we give a detailed description of MPEG packetization used in the simulation.

According to the RTP/UDP/IP packetization guidelines, each slice of the MPEG video can be carried in one or more RTP packets, and an integer multiple of slices can be carried in one RTP packet. For example, a large slice is divided into several RTP packets and several small slices are combined in one RTP packet. Because in our MPEG traces each frame is a slice, the size of each slice is usually larger than the Maximum Transfer Unit (MTU) of typical networks, where MTU is the maximum packet size a particular network can accept without imposing any fragmentation. Thus larger slices (frames) are fragmented in order to conform to the MTU requirement when they are received by the router. In our simulation, the value of MTU we choose is 552 bytes, a common MTU in real networks [8], [45].

When a very large slice is sent directly to the network, the network fragments it into many packets because of the MTU constraint. Because the slice is too large, it will occupy a large amount of space in the router buffer immediately, causing a very congested router. This will cause significant network performance degradation in the simulation. In addition, ns-2 uses packet queues in several of its queuing schemes, where queue sizes are in packets regardless of the size of the packets. This limitation creates a buffer size fairness problem. For example, a very large packet can only occupy one space in the queue, whereas many very small packets will take a large number of spaces in the queue. As a result of the large MPEG slice problem and the ns-2 packet queue

limitation, the MPEG traces in the simulation have to be appropriately packetized before they are sent to the network. The following is our packetization method.

Every slice (which is equal to a frame) in an MPEG trace is packetized to an integer multiple of packets each of size 552 bytes (the MTU size). For example, if a frame is equal to 1800 bytes ($1800 = 552 \times 3.26$), 3 packets are created. If a frame is equal to 2100 bytes ($2100 = 552 \times 3.80$), 4 packets are created. Although such roundup and truncation cause unnecessary addition and deletion of bytes, they do not negatively affect the simulation result as to be shown in the later sections. If variable packet sizes are used in the simulation, the full buffer conditions can have different byte counts (even though the packet counts are the same), therefore, affecting the consistency and fairness of simulation results. Using a constant packet size not only overcomes the packet queue limitation in ns-2 but also simplifies the analysis and comparison of simulation results.

After each frame is packetized, the transmissions of packets belonging to the same frame are uniformly distributed in the first half of the frame duration (each frame duration is 40 milliseconds because the frame rate is 25 frames per second). Spreading packet transmissions helps avoid sudden congestion in the router buffer due to the large MPEG slice problem. The choice of the first half of the frame duration is an engineering choice. If the distribution duration is too long, it creates too much delay. If the distribution duration is too short, it creates congestion problems. Thus our choice is a compromise between delay and congestion. This packetization method was first introduced in [7] and similar packetization methods were used in [17] and [23]. At this point, we have to emphasize that packetization methods can have significant influences on the simulation results because packetization affects the flow of traffic and its statistical attributes. Figure 3.1 illustrates our packetization method.

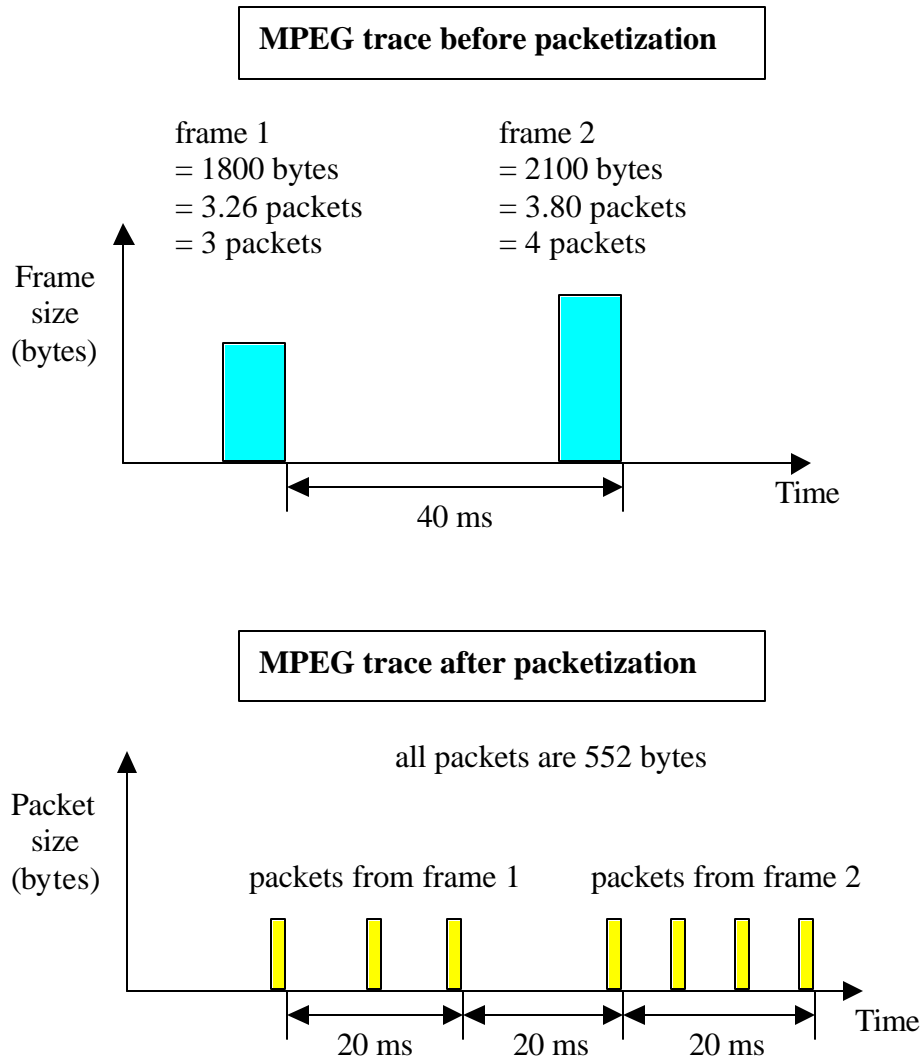


Figure 3.1. Packetization of MPEG traces.

3.3. Simulation configuration and parameters

Given the packetized MPEG traces, we now describe the details of our simulation configuration and parameters. As mentioned in Chapter 2, UDP is the primary protocol for real-time video applications. In our simulations, we transmit the packetized MPEG video via UDP over IP networks. Our packetization method is similar to the RTP packetization guidelines, although we do not utilize any service included in RTP. Each simulation runs for 30 minutes to cover the entire length of the MPEG trace. The

simulation topology consists of n sources generating MPEG video traffic to a common router connected to a traffic sink (destination), as shown in Figure 3.2. Traffic from a particular source is a flow; aggregate traffic received by the router consists of n flows.

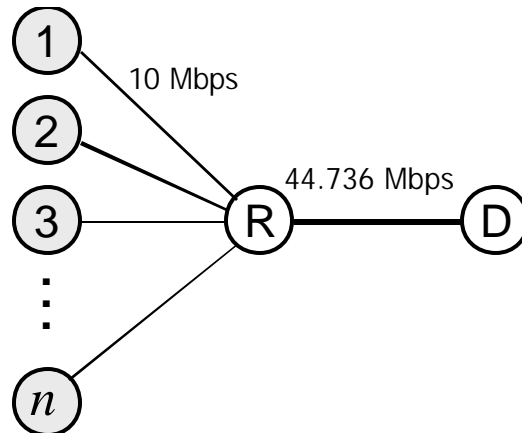


Figure 3.2. Network topology for the simulation: n sources, one router, and one destination [26].

In the majority of our simulations, among these n source nodes, every 10% of them transmit the same trace so that all ten traces are equally distributed. Table 3.2 shows the MPEG trace for each traffic source. During the simulation, each source selects a random point in the trace to start, and when the end of the trace is reached the source continues transmitting from the beginning of the trace. For the sources using the same trace, the random starting points are uniformly distributed in the trace. Note that even with random starting points, the cross-correlation between traffic from the same trace can still be relatively high if the trace has high degrees of LRD [17].

The link speed between the source and the common router is 10 Mbps, a common Ethernet speed, and the propagation delay is 1 millisecond. The link speed between the common router and the destination is 44.736 Mbps, based on the link speed of T-3/DS-3, and the propagation delay is 5 milliseconds. These settings are adopted from [26] in order to maintain the consistency of the simulation.

Table 3.2. MPEG trace for each traffic source. The total number of source is n , where n is an integer multiple of 10.

Traffic Source Number	MPEG Trace
1 to $0.1n$	The Silence of the Lambs
$0.1n$ to $0.2n$	Terminator 2
$0.2n$ to $0.3n$	MTV
$0.3n$ to $0.4n$	Simpsons
$0.4n$ to $0.5n$	German Talk Show
$0.5n$ to $0.6n$	Jurassic Park I
$0.6n$ to $0.7n$	Mr. Bean
$0.7n$ to $0.8n$	German News
$0.8n$ to $0.9n$	Star Wars
$0.9n$ to n	Political Discussion

Although T-3/DS-3 link speed is 45 Mbps in North America, 45 Mbps is the speed at the physical layer not at the IP layer. Because the simulation is only at the IP layer, the effective speed is approximately equal to 44.736 Mbps (by subtracting the speed required for the transmission of frame headers and trailers).

This network topology, as shown in Figure 3.2, mimics outgoing video traffic of a video server, where various videos are transmitted to the receivers (viewers) through IP networks. The source nodes and the common router correspond to a video server, and the destination node may represent the Internet. We do not explore more complex network topologies because they will introduce many additional parameters that are difficult to control independently and their simulation results may be difficult to compare.

Two different buffer sizes are used for the common router: 46 and 200 packets, which are approximately equal to 25 K and 100 K bytes given the constant packet size of 552 bytes. These buffer sizes yield maximum queuing delays of 5 and 20 milliseconds, which respectively represent a very stringent and a typical requirement for real-time video transmission. For high-end real-time video transmission, the delay requirement is within 10 milliseconds [14].

Five scheduling algorithms and queue management schemes are used in the simulations:

- FIFO/DropTail
- Random Early Drop (RED)
- Fair Queuing (FQ)
- Stochastic Fair Queuing (SFQ)
- Deficit Round Robin (DRR)

The focus of our research is FIFO/DropTail. A detailed description of each scheme is presented in Chapter 6.

Chapter 4

QoS Parameters

Real-time video applications have stringent requirements for loss, delay, and delay jitter. Both packet loss and packet delay can degrade the video quality at the receiver. Packet loss results in undesirable gaps and hiatus in video streaming. Large packet delay is equivalent to packet loss because in real-time applications new data overwrites old data. Large delay variation (jitter) degrades the performance of the video play-out buffer in the receiver and the smoothness of the video. There are many sources of packet loss and delay, such as loss due to link error and propagation delay. In our studies, we focus only on packet loss and packet delay due to buffer overflow and queuing in the router. Packet loss and delay also reflect the interaction between traffic and networks, such as congestion. In addition, we also examine other network performance measures such as the buffer occupancy probability and the throughput of the router. In this chapter, we explain how QoS parameters are measured and quantified in the simulation, and we describe our analysis approach.

4.1. Loss

The simplest way to quantify loss is by calculating the overall loss rate, which is equal to the total amount of lost traffic divided by the total amount of input traffic over the a certain period of time. Because we use a constant packet size in our simulation, the loss rate can be expressed as

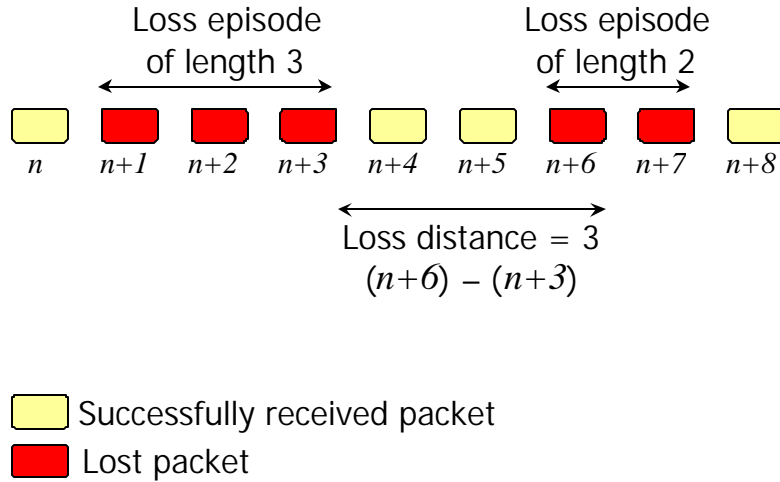
$$loss\ rate = \frac{total\ number\ of\ dropped\ packets}{total\ number\ of\ input\ packets}. \quad (4.1)$$

Related to loss rate is another network performance parameter - traffic load, which influences the loss rate. Traffic load indicates the overall degrees of congestion at a particular node in the network. We defined traffic load as

$$\text{traffic load} = \frac{\text{total number of arrived packets}}{\text{total number of packets that can be transmitted}}. \quad (4.2)$$

Although the packet loss rate is an important parameter, it cannot adequately capture the detailed loss pattern. For the same loss rate, loss patterns may be very different. To overcome the insufficiency of the loss rate, we adopt another measure called loss episodes to capture the loss pattern [26], [52].

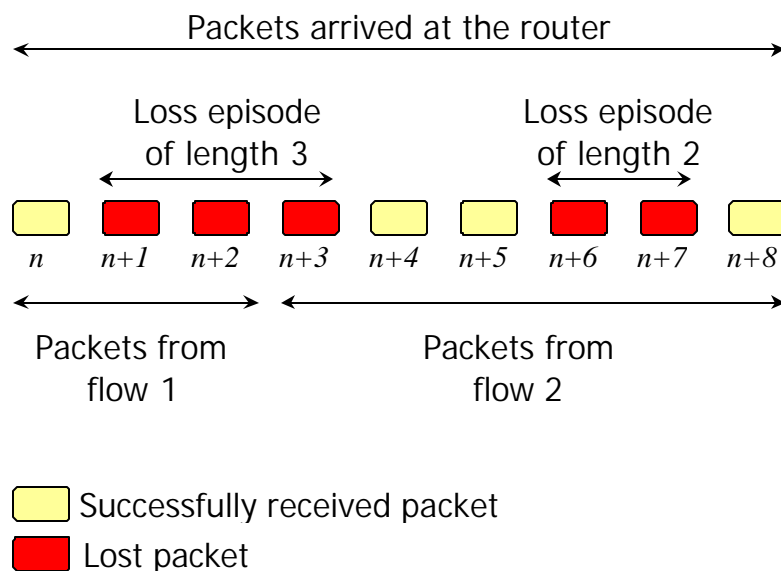
When one or more packets are lost consecutively, these lost packets constitute a loss episode. If k packets are lost in a row, then the corresponding loss episode is of length k . And the loss distance is a measure of spacing between two consecutively lost packets. Figure 4.1 is an illustration of loss episodes and loss distance.



$$npkt_k = k \cdot o_k \quad (4.4)$$

$$Npkt = \sum npkt_k \quad (4.5)$$

The definition of loss episode can be applied to the loss pattern of a particular traffic flow (per flow loss) received by the router or aggregate traffic from several flows received by the router (aggregate loss). The per-flow loss can be separated from the aggregate loss. In Figure 5.2, we illustrate the difference between the per-flow loss and the aggregate loss.



Aggregate loss: Two loss episodes (length 3 and length 2)

Per-flow loss:

Flow 1: One loss episode of length 2

Flow 2: Two loss episodes, one of length 1 the other of length 2

Figure 4.2. Illustration of per-flow and aggregate packet loss.

Using loss episodes, we can analyze the loss pattern by examining the distribution of loss episodes, which reflect how packets are dropped in the network. The loss episode

analysis is especially useful when it is applied to a particular video traffic source (per-flow loss), because real-time video applications are more susceptible to consecutive packet losses than sporadic single packet losses. We use two distributions in our analysis to characterize the packet loss pattern: the contribution of loss episodes of various lengths to the total number of loss episodes, and the contribution of lost packets from various loss episodes to the total number of lost packets. Using the notations Eqs. (4.3) – (4.5), these two distributions can be expressed as:

contribution of loss episodes of various lengths to the total number of loss episodes

$$\frac{o_k}{O_{total}} \quad (4.6)$$

contribution of lost packets from various loss episodes to the total number of lost packets

$$\frac{npkt_k}{Npkt} \quad (4.7)$$

The following example illustrates the difference between the contribution of loss episodes and the contribution of lost packets:

Total number of lost packets = 200
 Total number of loss episodes = 125

Loss episode of length	No. of occurrences	No. of lost packets
1	70	1 x 70 = 70
2	40	2 x 40 = 80
3	10	3 x 10 = 30
4	5	4 x 5 = 20

Contribution of loss episodes of length k to total number of loss episodes:

length 1: $70/125 = 0.56$ length 2: $40/125 = 0.32$
 length 3: $10/125 = 0.08$ length 4: $5/125 = 0.04$

Contribution of lost packets from loss episode of length k to total number of lost packets:

length 1: $70/200 = 0.35$ length 2: $80/200 = 0.40$
 length 3: $30/200 = 0.15$ length 4: $20/200 = 0.10$

For the per-flow packet loss, we are interested in not only the contribution of loss episode for each individual flow but also the average. The calculation of contribution of loss episodes of various lengths to overall number of loss episodes, averaged over all individual flows, can be illustrated using the following example.

Table 4.1. Example of per-flow packet loss contribution calculation.

Assume there are three flows in total to the router:		
Flow	No. of occurrences of loss episode of length k	
	1 packets	2 packets
#1	10	6
#2	8	6
#3	9	5
Contribution of loss episode of length i, averaged over all individual flows	$(10+8+9)$, $(10+8+9+6+6+5)$ = 0.6136	$(6+6+5)$, $(10+8+9+6+6+5)$ = 0.3863

4.2. Delay

Delay occurs when a packet waits in the buffer for the service (queuing delay) and when a packet is processed for transmission in the router (service delay). We use the term queuing delay to refer to both queuing and service delay. Thus, queuing delay is equivalent to the duration from the time a packet enters the router buffer until the time it leaves the router. Delay can be analyzed in two ways; the delay pattern can be examined by its distribution and by its autocorrelation function. The autocorrelation function of delay can be used to indicate how packet delays are correlated for a sequence of packets. It is defined as

$$ACF(d, l) = \frac{\sum_{i=1}^{n-l} [(d_i - m_d)(d_{i+l} - m_d)]}{\sum_{i=1}^n (d_i - m_d)^2} \quad (4.6)$$

where d_i is the delay of the i^{th} packet, n is the total number of packets measured, \bar{m}_i is the average packet delay, d is the packet delay random variable, l is the lag of correlation.

Packet delay cannot always be defined because of the packet loss. In our analysis, the delay of a lost packet is undefined and it is excluded from the calculation of delay distribution and autocorrelation function as suggested in [2]. Although such exclusion affects the accuracy of the delay distribution and the autocorrelation function, one possible method to assess the validity of this statistical analysis is to evaluate confidence intervals [41]. In addition, the exclusion of the delay of lost packets can also be considered as missing samples in the delay measurement. Lastly, delay jitter can be calculated by subtracting the delays of two consecutive packets [20]. We analyze the delay jitter pattern by plotting the distribution of the magnitude of delay jitter.

4.3. Other QoS and network performance parameters

In addition to loss and delay, we also investigate the buffer occupancy probability (buffer size probabilities) of the router, per-flow throughput, per-flow traffic load, and per-flow loss rate, all of them reflect how incoming traffic affects the router and how the router reacts to the traffic.

The buffer occupancy probability measures how frequently the size of the buffer is equal to i packets, and it can be calculated by measuring the size changes in the buffer. For a maximum buffer size of N packets, there are $N+1$ states, where *state 0* corresponds to the empty buffer and *state N* corresponds to the full buffer. Whenever the state (buffer size) changes, it is recorded. The buffer occupancy probability of state i is then defined as

$$P(i) = \frac{\text{total number of times state} = i \text{ (buffer contains } i \text{ packets)}}{\text{total number of state (occupancy) changes} + 1}. \quad (4.7)$$

The throughput of the router is the total number of packets delivered by the router. In order to compare the performance and the fairness of different scheduling and queue management schemes, we are interested in the per-flow throughput for each traffic source defined as

$$TP_i = \frac{\text{total number of packets from source } i \text{ that are delivered}}{\text{total number of packets delivered}}. \quad (4.8)$$

Similarly we define the per-flow traffic load $load_i$ and the per-flow loss rate $loss_i$.

$$load_i = \frac{\text{total number of packets arrived from source } i}{\text{total number of packets arrived}} \quad (4.9)$$

$$loss_i = \frac{\text{total number of packets from source } i \text{ that are lost}}{\text{total number of lost packets}} \quad (4.10)$$

By comparing TP_i , $load_i$, and $loss_i$, we can evaluate how fairly the scheduling and queue management scheme in the router allocates bandwidth to different traffic sources.

Chapter 5

Scheduling and queue management schemes

Scheduling and queue management algorithms are essential parts of router functionalities. Routers use scheduling algorithms to decide how and when packets are served. Routers employ queue management algorithms to determine when and which packets should be dropped from the buffer. The goal of good scheduling and queue manage schemes is to allocate bandwidth fairly among different traffic sources that may have different service requirements, while maximizing service utilization.

5.1. FIFO/DropTail

The combination of first-in-first-out (FIFO) scheduling and DropTail queue management is widely used in current network routers because of its simplicity. With FIFO, packets are served in the order that they are received. With DropTail, if the buffer is full when a packet arrives, the incoming packet is dropped.

5.2. Random early drop (RED)

Random early drop (RED) is a queue management scheme that monitors and controls buffer occupancy [16]. RED detects congestion by monitoring the average buffer size of the router. When the average buffer size is larger than the first threshold min_{th} but lower than the second threshold max_{th} , the incoming packets are dropped with probability Pa , which increases linearly as the average buffer size increases. When the average buffer size exceeds the second threshold max_{th} , the router drops randomly chosen packets from within the buffer with probability one. RED has two key objectives: one is to fairly distribute the effects of congestion across all traffic sources competing for the shared network capacity (as a result of the random packet drop); and the other is to create a

congestion avoidance mechanism by dropping packets when congestion is imminent (as a result of early packet drop) as a way to notify traffic sources the imminence of congestion before congestion actually occurs. The early drop action serves as a negative feedback signal to the traffic sources and congestion can be avoided if traffic sources reduce their traffic in time.

5.3. Fair queuing (FQ)

Fair queuing (FQ) was originally proposed by Nagle [30]. Nagle's FQ algorithm first divides the router buffer into sub-queues, one for each incoming traffic source (per-flow queuing). Then the router serves packets in the round-robin fashion (packet-by-packet round-robin scheduling). Nagle's FQ algorithm, however, assumes the packet size is constant, and thus it fails to provide throughput fairness when packets have different sizes. Demers, Keshav, and Shanker later proposed an improved version of FQ algorithm that solves the flaws in Nagle's FQ algorithm [10] (from this point we use FQ to refer to this algorithm and Nagle's algorithm to refer to Nagle's FQ algorithm). This FQ algorithm uses the same per-flow queuing mechanism, but instead of the packet-by-packet round-robin scheduling, it approximates the ideal bit-by-bit-round-robin scheduling in order to achieve throughput fairness under all conditions. The basic concept of FQ can be illustrated using the simplified FQ algorithm shown below.

FQ Algorithm

- Upon the reception of the k^{th} packet from flow i at time t , the router calculates the finish-time in bit-time for the packet using the equation:

$$F(i, k, t) = \max\{ F(i, k-1, t), R(t) \} + P(i, k, t) \quad (4.10)$$

where $F(i, k, t)$ is the finish-time of the current packet, $F(i, k-1, t)$ is the finish-time of the last packet from the same flow (thus, they use the same sub-queue), $R(t)$ is the round number which increments for every complete cycle of sending one bit per flow, and $P(i, k, t)$ is the size of the current packet in bits.

- When the router completes a packet transmission, the FQ scheduler checks the finish-time for the head-of-the-line packet in each sub-queue. The scheduler

finds the head-of-the line packet with the smallest finish-time and sends it to the outgoing link.

Although FQ achieves throughput fairness, it has very a high degree of computational complexity. Hence it is usually too expensive to be widely adopted in network routers [42].

5.4. Stochastic fair queuing (SFQ)

Stochastic fair queuing (SFQ) was proposed by McKenney [28] to address the inefficiencies of Nagle's algorithm, where the number of sub-queues must be equal to the number of traffic sources. Because the number of traffic sources can be very large and not all traffic sources are active at the same time, Nagle's is inefficient and has high computational complexity. SFQ uses hashing, which has much less computational complexity than Nagle's algorithm, to map packets into corresponding queues and the total number of queues only has to be larger than the total number of active flows. The throughput fairness of SFQ is stochastic; that is, packets from different traffic sources will collide into the same sub-queue when the number of active flows is larger than the number of allocated sub-queues. The probability of such unfairness due to sub-queue collision can be minimized by setting the hashing index to be a small multiple of the number of active flows [42]. SFQ, however, is still a packet-by-packet round-robin scheduling scheme and thus it inherits the same flaws in Nagle's algorithm. The other key functionality of SFQ is its packet dropping policy. When the buffer is full, SFQ drops packets from the longest sub-queue (longest sub-queue packet drop policy) instead of dropping the arriving packets

5.5. Deficit round robin (DRR)

Deficit round robin (DRR) scheduling algorithm was proposed by Shreedhar and Varghese [42] to approximate the performance of FQ using a less complex computational

structure. DRR uses the same hashing mechanism and longest sub-queue packet drop policy as in SFQ. DRR serves sub-queues in the round-robin fashion. For each sub-queue, a deficit counter (in bytes) is assigned. In each round of service, the deficit counter is incremented by a quantum (in bytes). Each sub-queue, when served, is allowed to send its packets one by one if the packet size is smaller than the deficit counter. The deficit counter is decremented by the packet size after a packet is sent. When the deficit counter is depleted, the DRR scheduler moves to the next sub-queue. The follow is a simplified DRR algorithm for illustration.

DRR Algorithm

Let D_i be the deficit counter for flow i , Q_i be the quantum size for sub-queue i , and P_i be the head of line packet for sub-queue of flow i .

```

 $D_i = D_i + Q_i$ 
While ( $D_i > 0$ )
  If ( $D_i \geq \text{size}(P_i)$ )
    Then send  $P_i$ 
     $D_i = D_i - \text{size}(P_i)$ 
  Else exit the While Loop
End of the While Loop
Go to the next non-empty sub-queue

```

DRR is capable of achieving nearly perfect throughput fairness as FQ with much less computational complexity than FQ. Figure 5.1 illustrates the common structure of FQ, SFQ, and DRR.

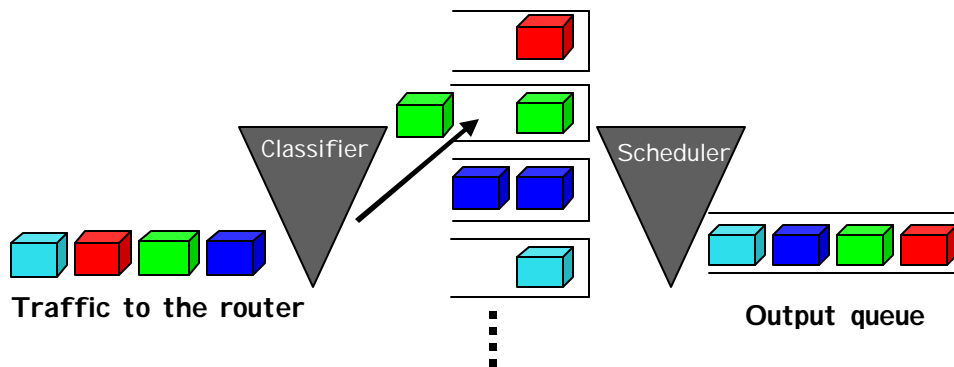


Figure 5.1. Common structure of FQ, SFQ, and DRR.

Chapter 6

FIFO/DropTail simulation results and analysis

In this chapter, we present the FIFO/DropTail simulation results and their analysis. We first start with a comparison of the effects of different packetization methods. We use the results to explain our packetization choice as mentioned in Chapter 3.2. Then we present a detailed discussion on the characteristics of packet loss, packet delay, and other QoS and network performance parameters.

6.1. Comparison of packetization methods and the addition of RTP header

As mentioned previously in Chapter 3.2, in simulation we use a constant packet size of 552 bytes. The reason for using constant packet is to overcome the packet-queue limitation of ns-2, and to simplify analysis. To study the effect of packetization and the addition of RTP headers on the simulation results, we examine the aggregate packet loss pattern in the common router. The aggregate packet loss pattern can reflect the characteristics of the network traffic, because it is closely related to packet arrival pattern of the traffic.

The four methods to be compared are

- Method 1: Constant packet size of 552 bytes.
- Method 2: Constant packet size of 552 bytes, including a 16 byte RTP header and a 4 byte RTP MPEG specific header.
- Method 3: Maximum packet size of 552 bytes (no truncation or round-up as mentioned in Chapter 3.2).

- Method 4: Maximum packet size of 552 bytes, including a 16 byte RTP header and a 4 byte RTP MPEG specific header.

Using these methods, we can examine the effect of the addition of RTP headers, and the effect of data addition and deletion caused by constant packet size packetization. In these simulations, we use 100 traffic sources to create a heavily congested network at the router in order to amplify the packet loss and simplify the analysis. To simplify the analysis, we use only the Terminator trace. All MPEG video traffic is delivered using UDP/IP and the router buffer size is 46 packets. Table 6.1 shows the basic statistics of the simulation results and Figure 6.1 shows packet loss patterns in the router (aggregate loss) using the contribution of loss episodes analysis defined in Eq. (4.6).

Table 6.1. Summary of simulation results for the comparison of the effect of different packetization methods and RTP header addition.

Packetization Method	Number of Packets Arrived	Number of Packets Transmitted	Number of Packets Dropped	Packet Loss Rate (%)
1	11330442	11074880	255554	2.255
2	11456413	11236252	220151	1.922
3	13322766	13105722	217028	1.629
4	13831755	13623609	208125	1.505

As shown in Table 6.1, the packet loss rate is not proportional to the number of packets in the traffic. Method 1 generates the least amount of packets but it has the highest packet loss rate. The traffic pattern influences the packet loss rate more than the total number of packets arrived at the router in the simulation. Table 6.1 alone, however, is an insufficient comparison. The loss characteristics can be reflected in Figure 6.1. As shown in Figure 6.1, the effects on the contribution of loss episodes by all four methods are very similar. For loss episodes smaller than 10 packets, the distributions are almost identical. The impacts of different numbers of packets generated by these methods are concentrated in the region of long loss episodes, where probabilities are much smaller.

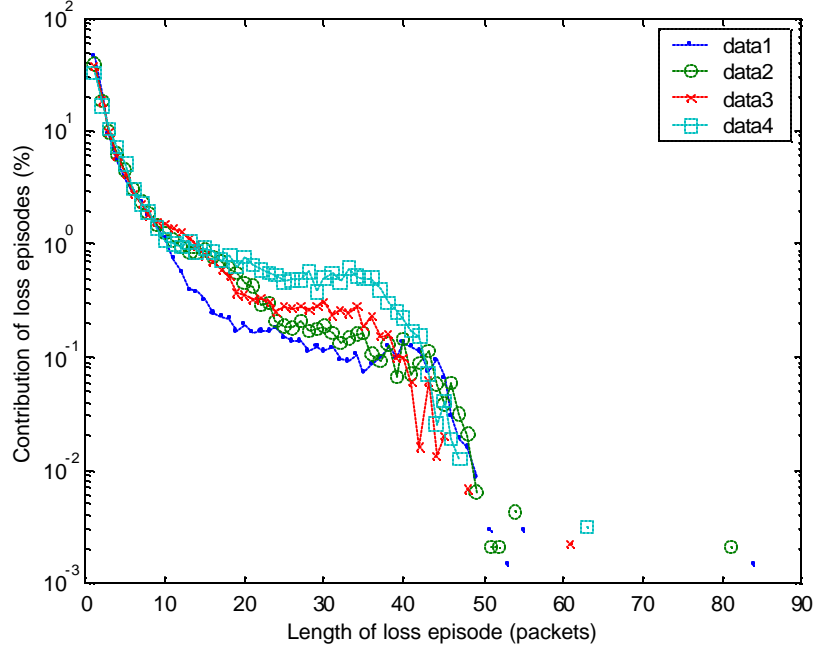


Figure 6.1. Contribution of loss episodes of various lengths to the overall number of loss episodes. Simulation with different packetization methods with or without RTP headers, using the Terminator trace. Data i corresponds to method i .

From Table 6.1 and Figure 6.1, we believe that using method 1 does not result in significant changes in the simulation results. Thus, throughout most of our studies and in the remaining sections of this report, we adopt method 1, which uses constant packet size of 552 bytes and does not include the RTP header. We choose not to include the RTP headers because no information from the RTP header is used in the simulation.

6.2. Packet loss

6.2.1. Aggregate packet loss

The last section presented a preliminary look at the aggregate loss behavior in the common router. In this section, we focus on a more detailed analysis of aggregate packet loss in the router (aggregate loss). The simulation in this section and the remaining sections in this report uses all ten MPEG traces as described in Chapter 3.3. In the following simulation, we use 100 traffic sources to create a heavily congested network. All MPEG video traffic is delivered using UDP/IP. The router buffer sizes are 46 and 200 packets. Table 6.2 shows the basic statistics of the simulation results, while Figure 6.2 shows the contribution of loss episodes of the aggregate packet loss at the router.

Table 6.2. Summary of simulation results for the FIFO/DropTail simulation with 100 traffic sources.

Buffer size (packets)	Number of Packets Arrived	Number of Packets Transmitted	Number of Packets Dropped	Traffic Load (%)	Packet Loss Rate (%)
46	14997184	14128986	868186	82.24	5.789
200	14997184	14796732	200499	82.24	1.337

As shown in Table 6.2, the traffic load with 100 traffic sources is 82%, indicating high network congestion at the router. Figure 6.2 shows that as a result of such high congestion, we observe loss episodes of very long lengths. In the log-scale graph, the distribution is approximately linear in the short loss episode region, but as the loss episode length increases the linearity is no longer apparent. The lack of overall linearity in the log scale indicates that the overall probability distribution does not decay exponentially. Related to Figure 6.2 is the contribution of lost packets from loss episodes of length i to the overall lost packets as shown in Figure 6.3. The contribution of lost packets is defined in Eq. (4.7). Figure 6.3 shows that loss episodes of length two have the highest contribution to the total amount of lost packets even though loss episodes of length one is the most common loss episode. Again, overall linearity is absent in this log-scale distribution.

In Figure 6.5 we plot the aggregate packet loss process used to derive the distribution of loss episodes and lost packets shown in Figure 6.2 and 6.3. For FIFO/DropTail queue, the aggregate packet loss process at the router reflects some aspects of the packet arrival process of incoming traffic, because packets are delivered in their arrival order and incoming packets are lost when the buffer is full. Both burstiness of the traffic (as shown by the long loss episode) and presence of congestion (as shown by the dense region) can be inferred from Figure 6.5. The studies in [26] and [51] and have shown that the packet loss process in a similar simulation setting, in fact, exhibits self-similarity.

We also observe similar aggregate packet loss process patterns in simulations where all traffic sources use only a single MPEG trace. Several sample simulation results are included in the Appendix A for reference. In Figure 6.4 we plot the distribution of the packet arrival process, in which we record the number of packets arriving in one millisecond intervals. The analysis of the traffic arrival and the loss process is not trivial because of their complexities such as the influence of different time-scales and self-similarity. Figure 6.4, alone, cannot adequately describe the packet arrival process of video traffic. Although the in-depth analysis of arrival and loss processes is beyond the scope of our research, we note that the wavelet analysis has been shown as a useful tool for the analysis of self-similarity and the effect of time-scales [51].

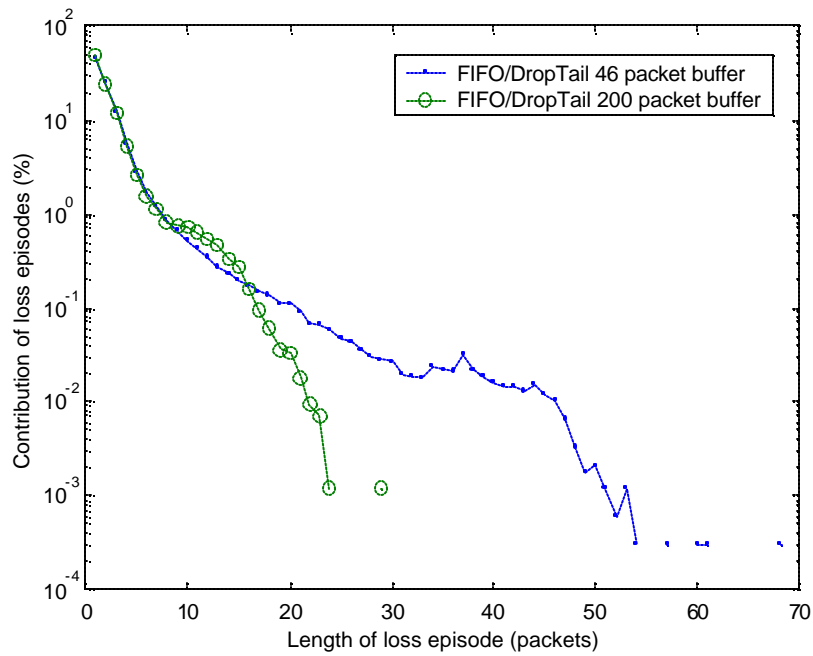
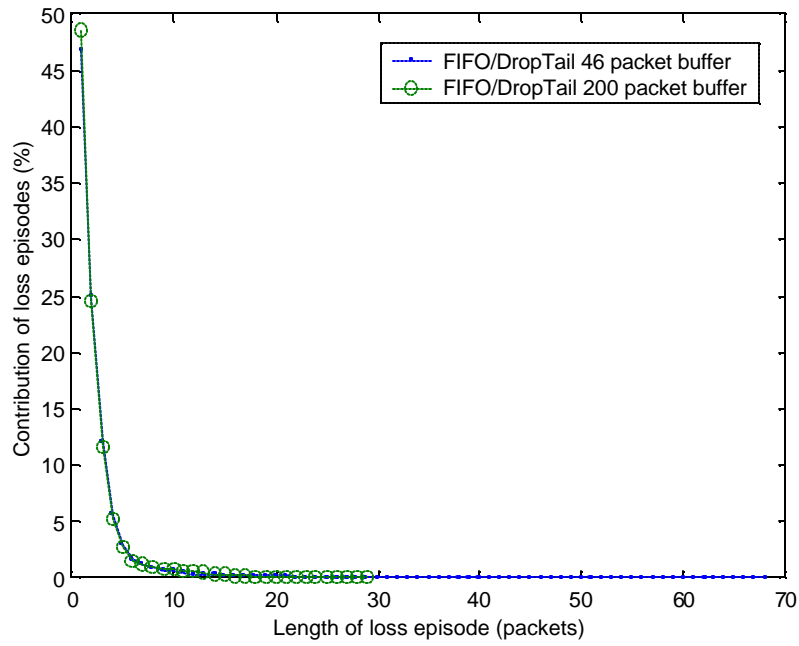


Figure 6.2. Contribution of loss episodes of various lengths to the overall number of loss episodes, linear (top) and log scale (bottom). Simulation with 100 traffic sources, using 10 different MPEG traces.

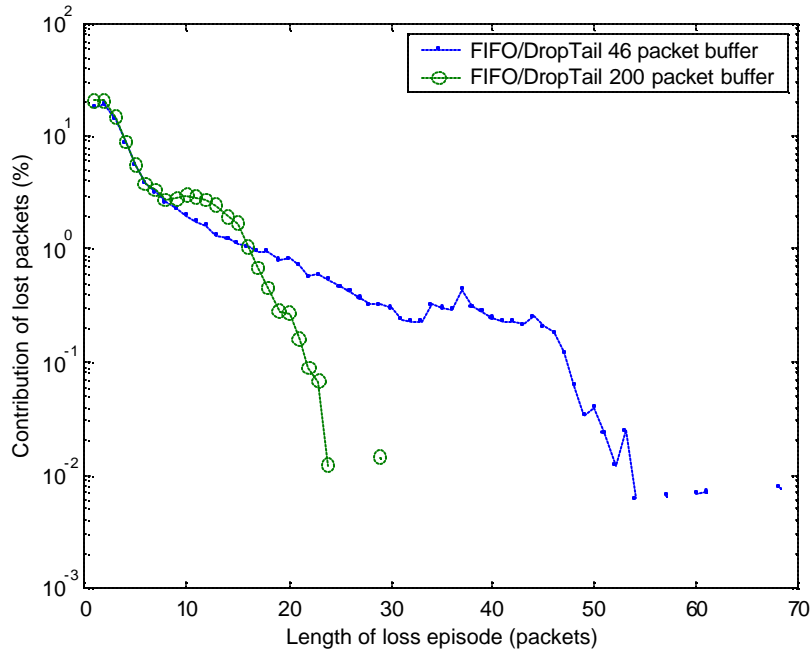
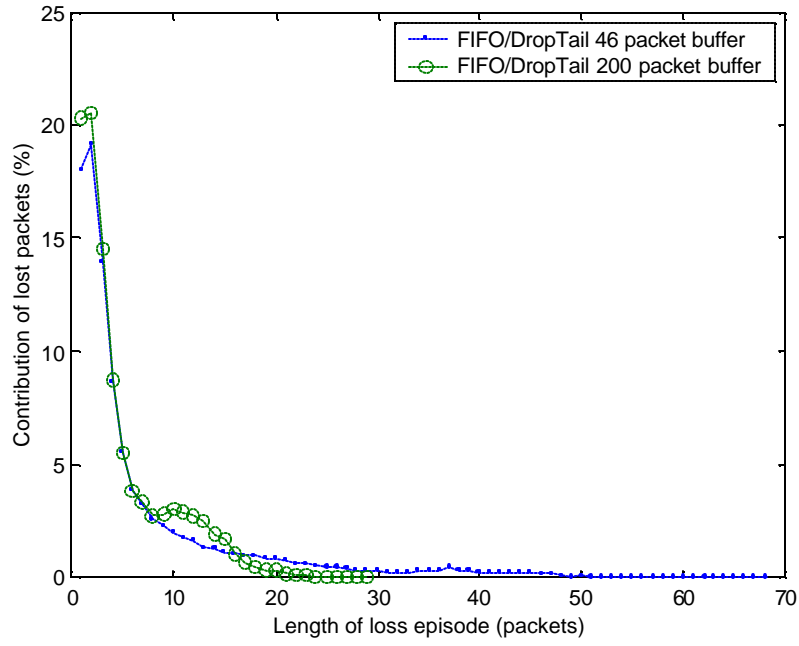


Figure 6.3. Contribution of lost packets from loss episodes of i to the overall number of lost packets, linear (top) and log scale (bottom). Simulation with 100 traffic sources, using 10 different MPEG traces.

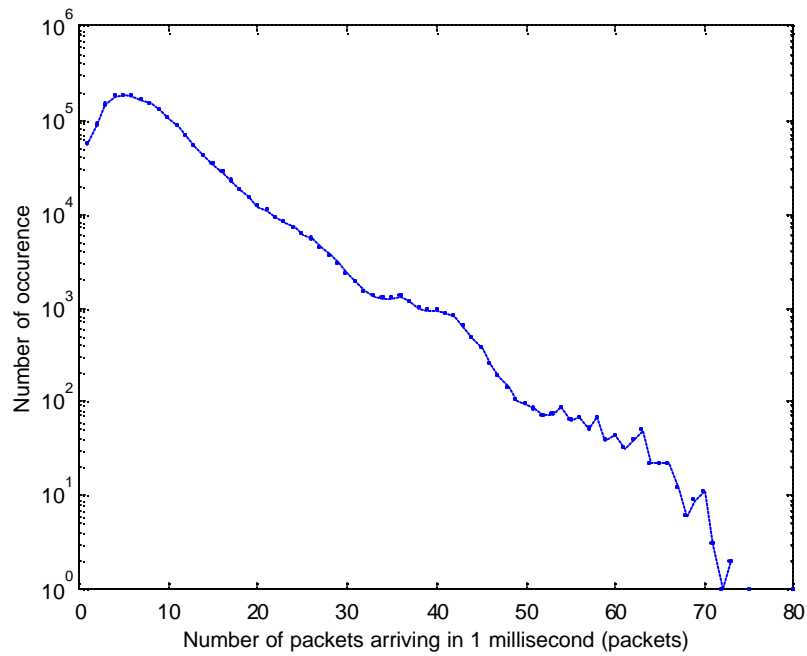
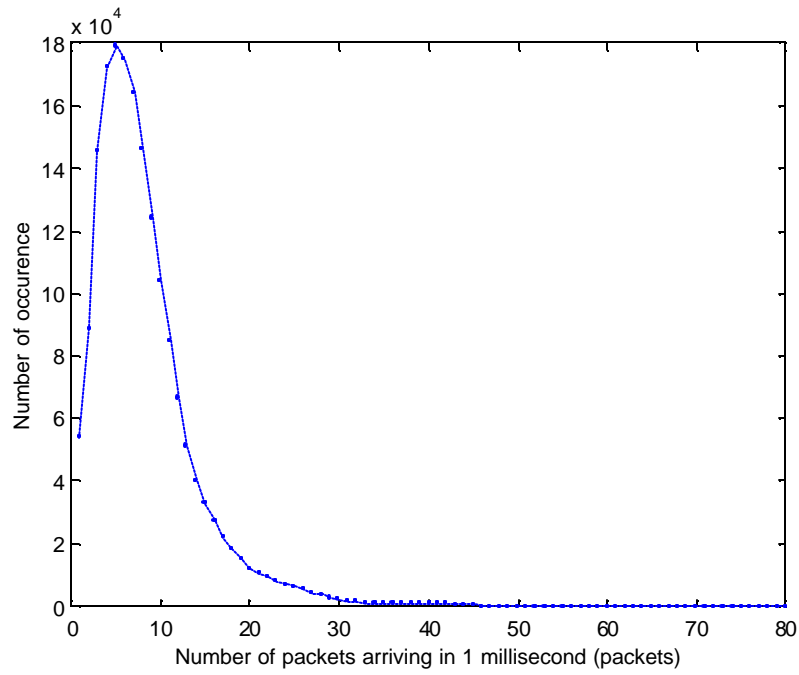


Figure 6.4. Distribution of packet arrival process measured in one-millisecond intervals, linear (top) and log (bottom) scale. Simulation with 100 traffic sources, using 10 different MPEG traces.

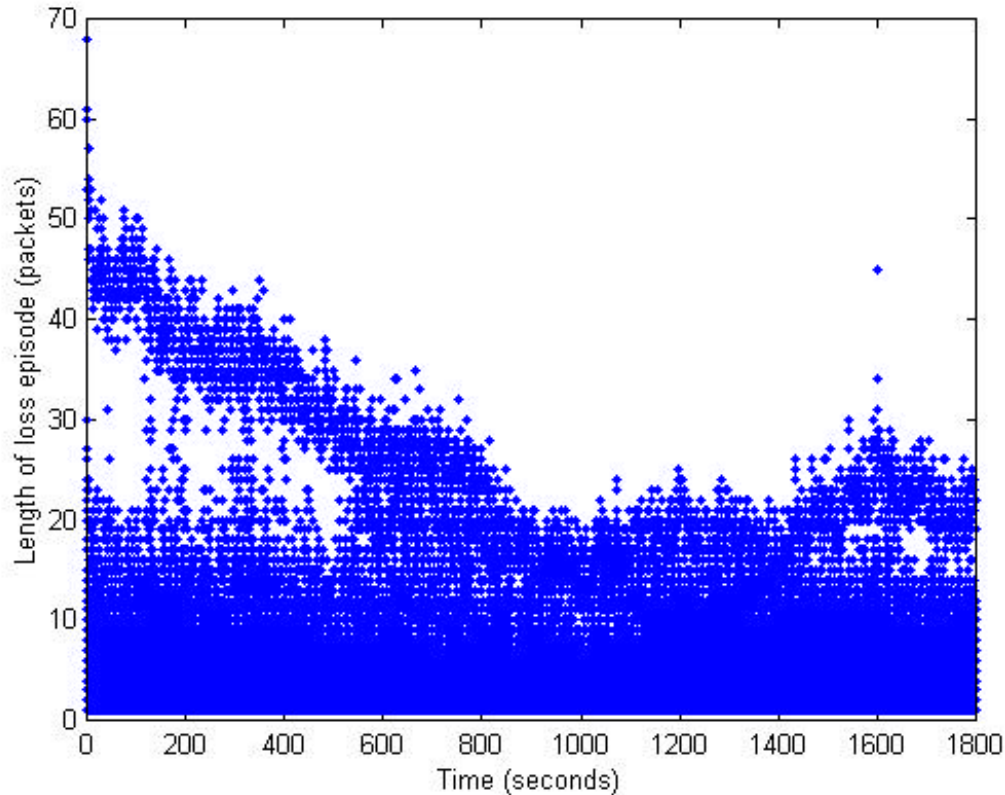


Figure 6.5. Aggregate packet loss process. Simulation with 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets. Each point in the graph corresponds to a particular loss episode. The occurrence time for each loss episode is the occurrence time of the first lost packet in the loss episode.

6.2.2. Effects of traffic load on aggregate packet loss

In this section, we investigate how traffic load affects the characteristics of aggregate packet loss. Our objective is to determine if the same characteristics shown in the last section can be observed under different traffic load conditions. To simulate different traffic loads, we vary the total number of traffic sources while keeping the original 10 MPEG traces equally distributed among all traffic sources. We present the results for the simulation with the buffer size equal to 46 packets. The basic simulation results are summarized in Table 6.3

Table 6.3. Summary of simulation results for the FIFO/DropTail simulation with various numbers of traffic sources, buffer size = 46 packets.

Number of Traffic Sources	Number of Packets Arrived	Number of Packets Transmitted	Number of Packets Dropped	Traffic Load (%)	Packet Loss Rate (%)
100	14997184	14128986	868186	82.24	5.789
90	13501765	13002996	498763	74.04	3.694
80	12041042	11593871	447171	66.03	3.714
70	10535949	10401653	134296	57.78	1.275
60	9007789	8936442	71347	49.40	0.7921
50	7536633	7530650	5983	41.33	0.07939
40	6008029	6007942	87	32.95	0.001448

As shown in Table 6.3, the number of traffic source from 40 to 100 corresponds to traffic load from 33% to 82%. When the number of sources is 30 or below, no packet loss is observed. Figure 6.6 shows the contribution of loss episodes. In the region with short loss episodes (up to 20 packets), the distribution curves for all traffic loads are rather similar. In the region with long loss episodes, the shape of each distribution curve depends on the traffic load since the degrees of burstiness vary with traffic load.

In Figure 6.7, we focus our analysis on the distribution in the region with loss episodes of lengths one to three packets. The contribution of loss episode of length one decreases as traffic load increases, whereas the contribution of loss episode of length three increases as traffic load increase. For loss episodes of length two, the contributions are very similar for all traffic loads. This phenomenon has also been observed in [26]. We repeat the simulation similar to the simulation in [26] and include the results in the Appendix B for reference.

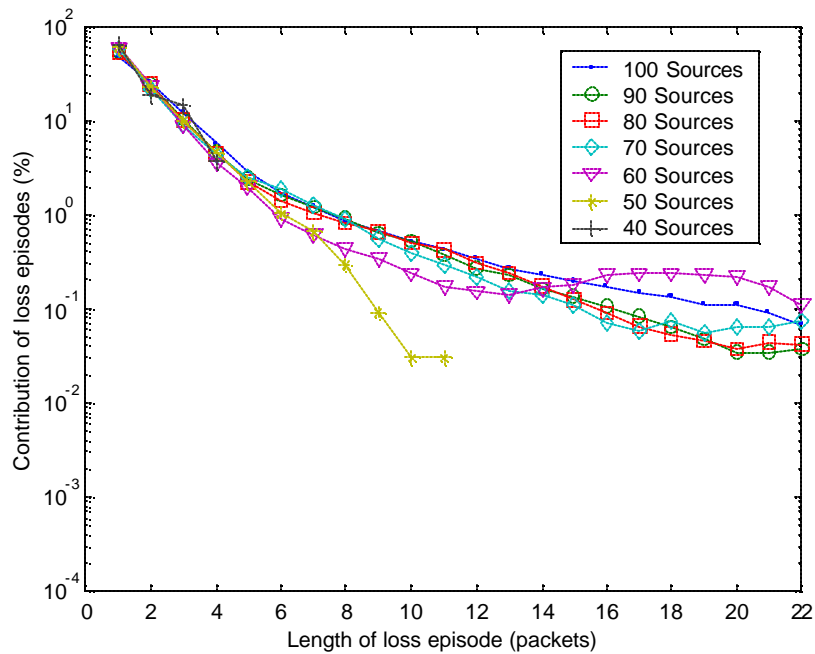
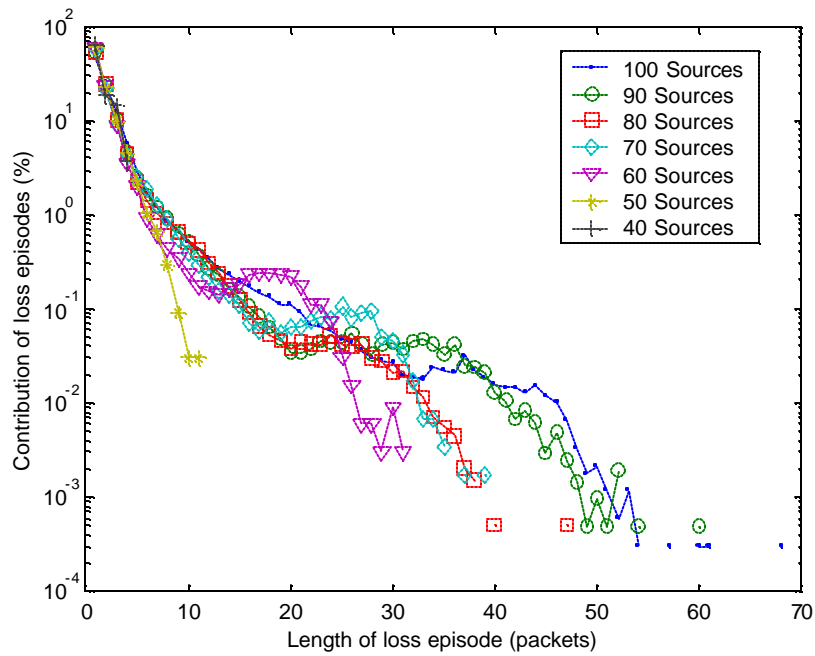


Figure 6.6. Contribution of loss episodes of various lengths to the overall number of loss episodes, entire episode length range (top) and episode length up to 22 packets (bottom). Simulation with 40 to 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets.

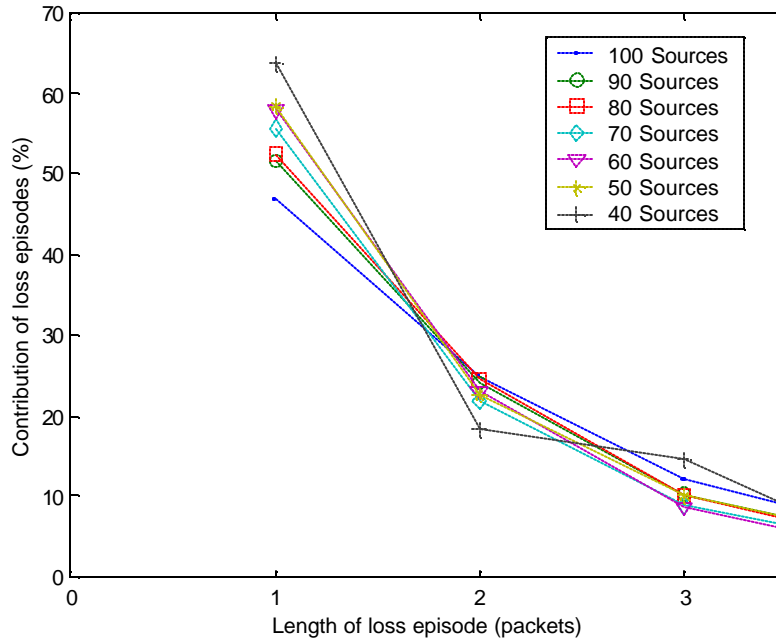


Figure 6.7. Contribution of loss episodes of various lengths to the overall number of loss episodes, episode length from 1 and to 3. Simulation with 40 to 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets.

6.2.3. Per-flow packet loss

Besides the aggregate packet loss at the router, the per-flow packet loss is also important to characteristics of packet loss. Per-flow packet loss reflects the end user video reception quality. For example, consecutive packet losses (or loss episode of length larger or equal to two) degrade video quality more than sporadic single packet losses. We apply the same analysis to the per-flow packet loss as to the aggregate packet loss. We use 100 traffic sources in the simulation, and for each flow, we extract the contribution of loss episodes of various lengths to its overall number of loss episodes as defined in Chapter 4.1. The results for each of the 100 flows are included in the Appendix C. Figure 6.8 shows the contribution of loss episodes of various lengths to the overall number of loss episodes, averaged over all individual flows.

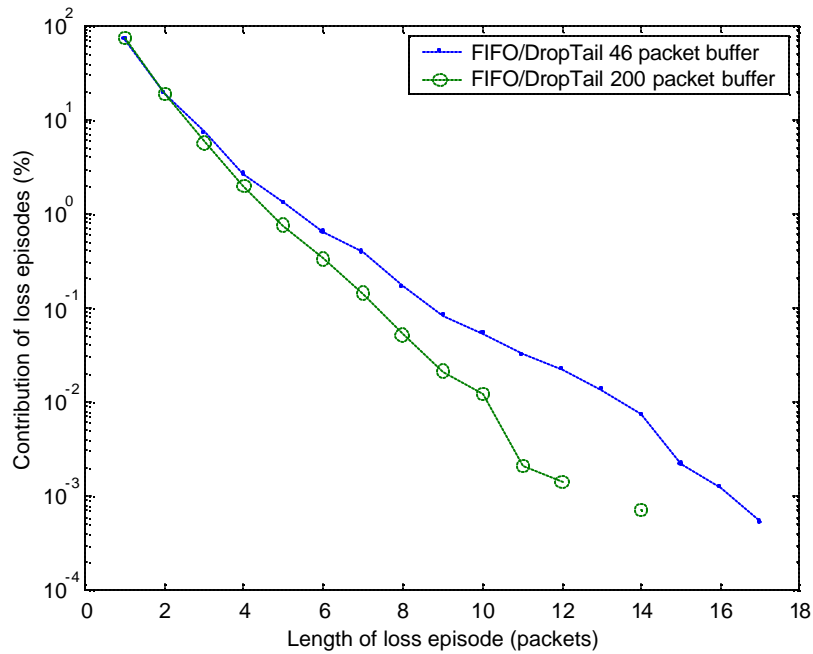
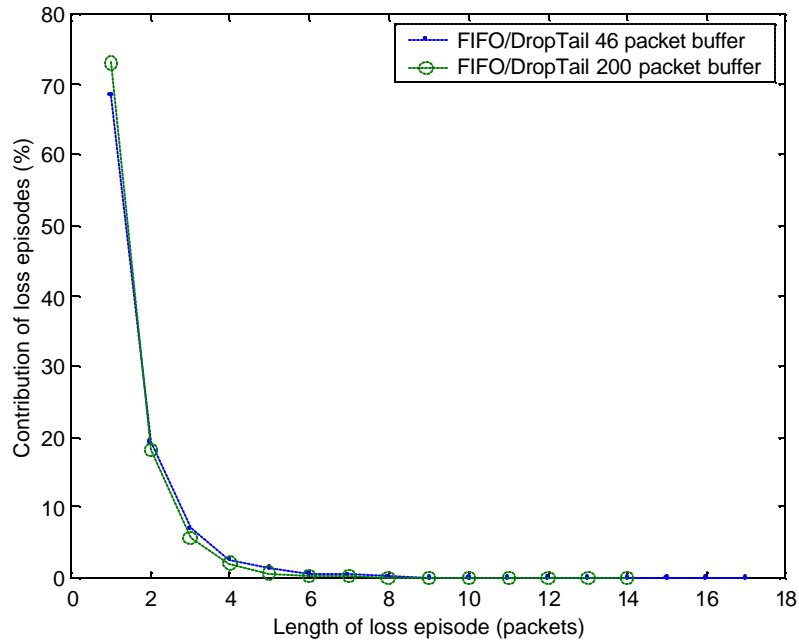


Figure 6.8. The contribution of loss episodes of various lengths to the overall number of loss episodes, averaged over all individual flows (per-flow loss), linear scale (top) and log scale (bottom). Simulation with 100 traffic sources, using 10 different MPEG traces.

As shown in Figure 6.8, the distribution is almost linear on the log scale, indicating that the distribution decays exponentially as the length of loss episode increases. This exponentially decaying characteristic is more preferable than the pattern observed in the aggregate loss because it implies that long consecutive packet losses are very rare and single packet losses are more prevalent. Thus, error-concealment techniques, which are more effective against short data losses, can be adopted to improve the video quality.

6.3. Packet delay

6.3.1. Packet delay distribution

Using the same simulation scenario with 100 traffic sources, we analyze the packet delay characteristics using the delay distribution and the autocorrelation function. Figure 6.9 shows the packet delay distribution for all packets delivered by the router to the destination.

As shown in Figure 6.9, delay distribution not only decays very slowly in the medium and the high delay region, it also increases rapidly in the region close to the maximum delays that are about 4.5 milliseconds for 46 packet buffer simulation and 19.7 milliseconds for 200 packet buffer simulation. This indicates a high possibility that the buffer size is close to the maximum as a result of frequent congestion (as to be shown in the analysis of buffer occupancy probability). Therefore, many packets experience queuing delay close to the maximum value of 4.5 and 19.7 milliseconds.

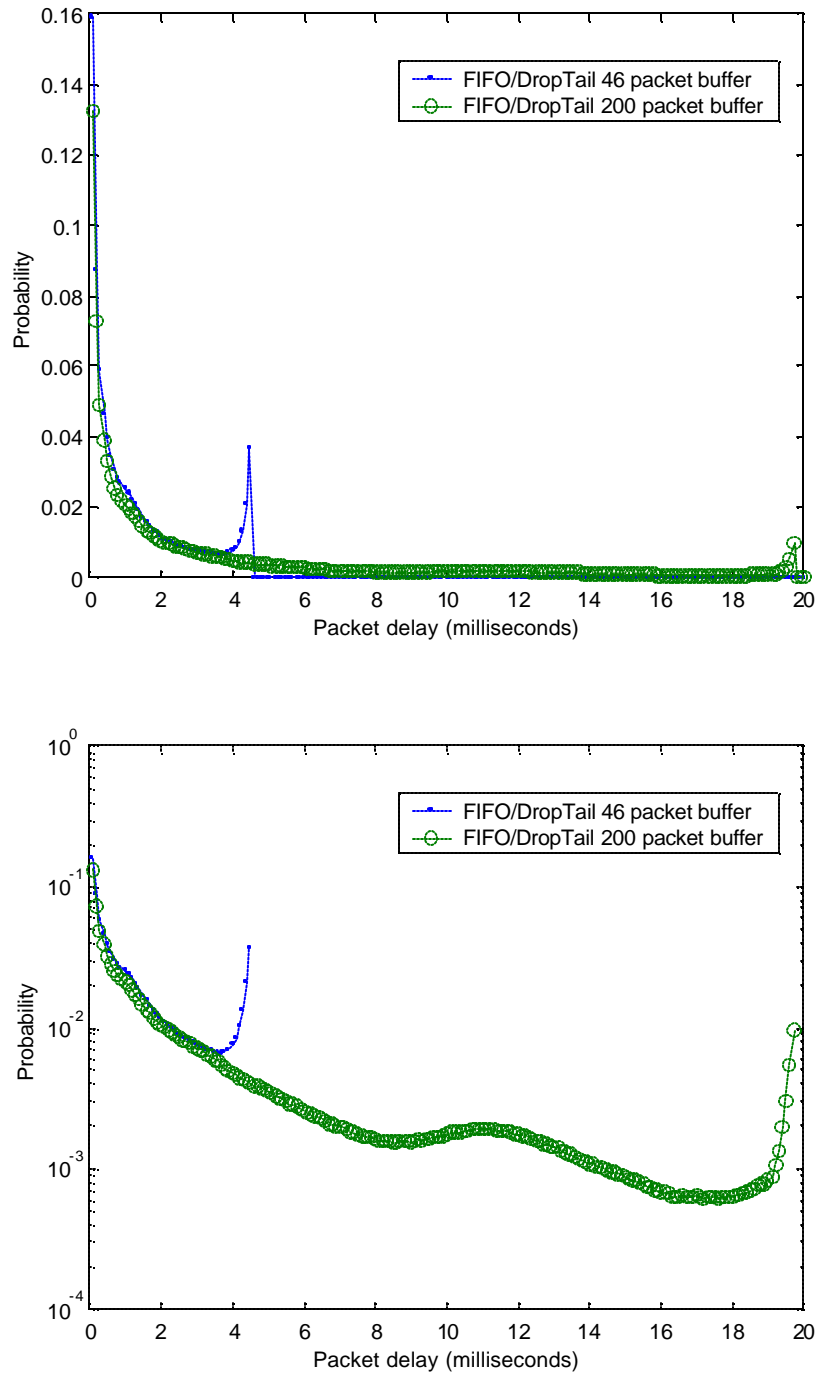


Figure 6.9. Probability distribution of packet delay for all delivered packets, linear scale (top) and log scale (bottom). Simulation with 100 traffic sources, using 10 different MPEG traces.

6.3.2. Packet delay autocorrelation function

To calculate the autocorrelation function for packet delays, the packets are ordered as they are delivered. For FIFO/DropTail, the packet arrival order is the same as the packet delivery order. As previously shown in Table 6.3, in a 30 minutes long simulation with 100 traffic sources, approximately 15 million packets arrive to the router and approximately 14 million packets are delivered from the router to the destination. Because of such a large number of packets, we are not able to calculate the autocorrelation function for all 14 million packets (a 14 million long packet delay sequence). Instead, we evaluate the autocorrelations function for selected data partitions covering the entire packet sequence as follows:

- Partition 1: packet no. 1 to packet no. 200,000
- Partition 2: packet no. 1,000,001 to packet no. 1,200,000
- Partition 3: packet no. 2,000,001 to packet no. 2,200,000
- Partition 4: packet no. 4,000,001 to packet no. 4,200,000
- Partition 5: packet no. 6,000,001 to packet no. 6,200,000
- Partition 6: packet no. 8,000,001 to packet no. 8,200,000
- Partition 7: packet no. 10,000,001 to packet no. 10,200,000
- Partition 8: packet no. 12,000,001 to packet no. 12,200,000
- Partition 9: packet no. 14,000,001 to packet no. 14,200,000

Using the autocorrelation function for these partitions enable us to approximate the autocorrelation function for the entire packet delay sequence. The autocorrelation functions for the simulation with the buffer size equal to 46 packets are shown in Figures 6.10, 6.11, and 6.12.

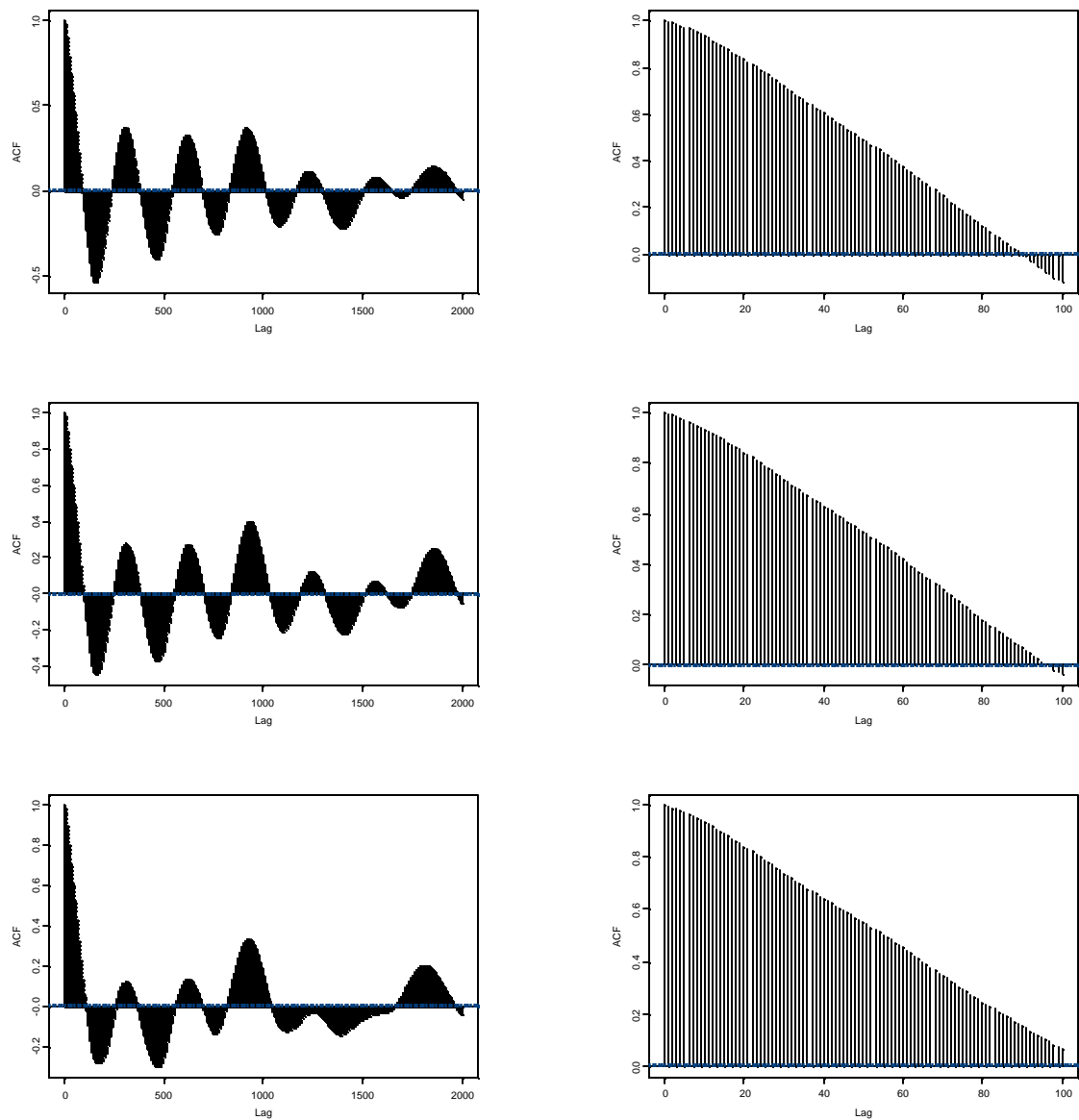


Figure 6.10. Autocorrelation function for packet delays. Top: packet no. 1 to 200,000. Middle: packet no. 1,000,001 to 1,200,000. Bottom: packet no. 2,000,001 to 2,200,000. Left: 2000 lag scale. Right: 100 lag scale. Simulation with 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets.

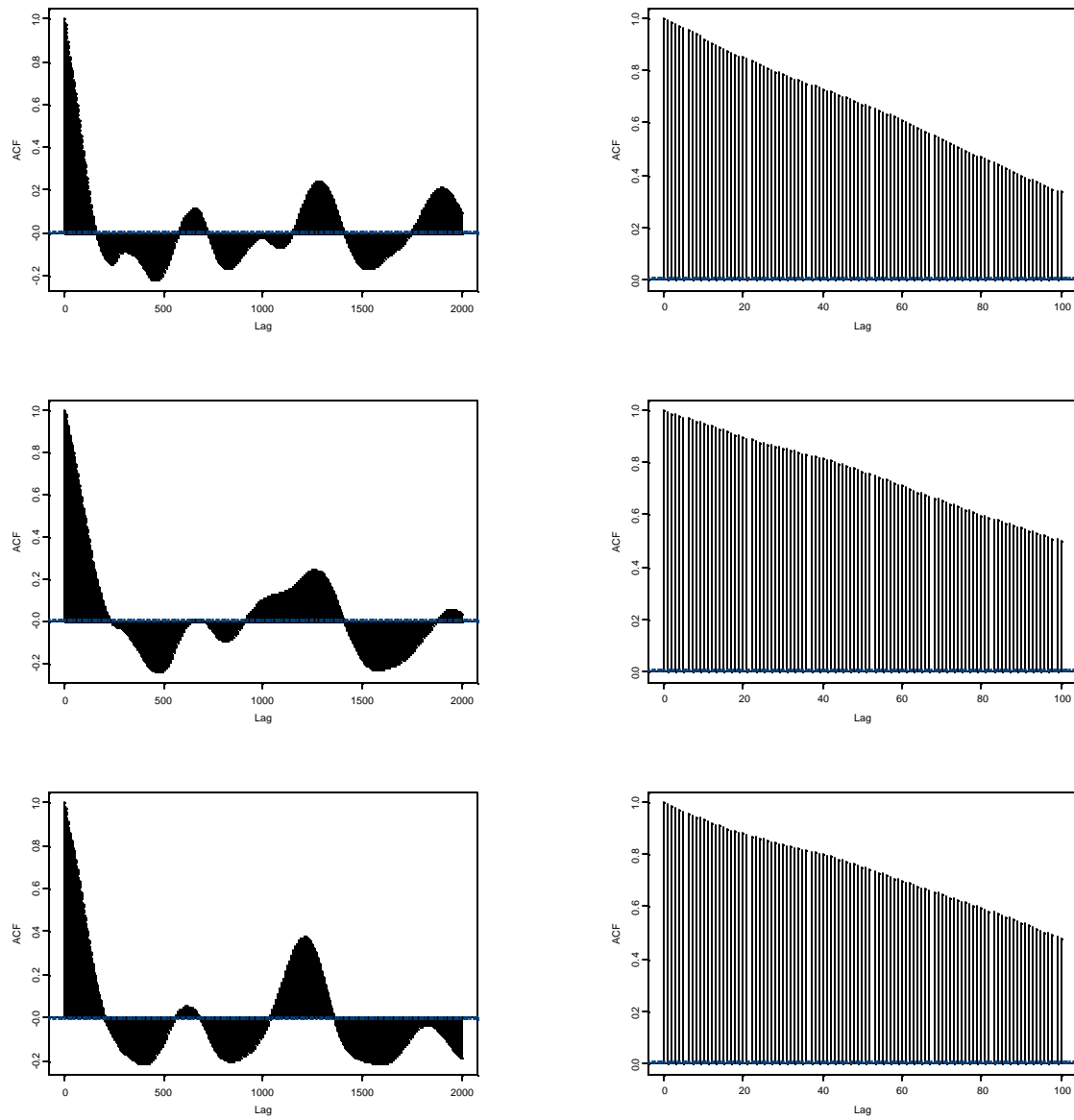


Figure 6.11. Autocorrelation function for packet delays. Top: packet no. 4,000,001 to 4,200,000. Middle: packet no. 6,000,001 to 6,200,000. Bottom: packet no. 8,000,001 to 8,200,000. Left: 2000 lag scale. Right: 100 lag scale. Simulation with 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets.

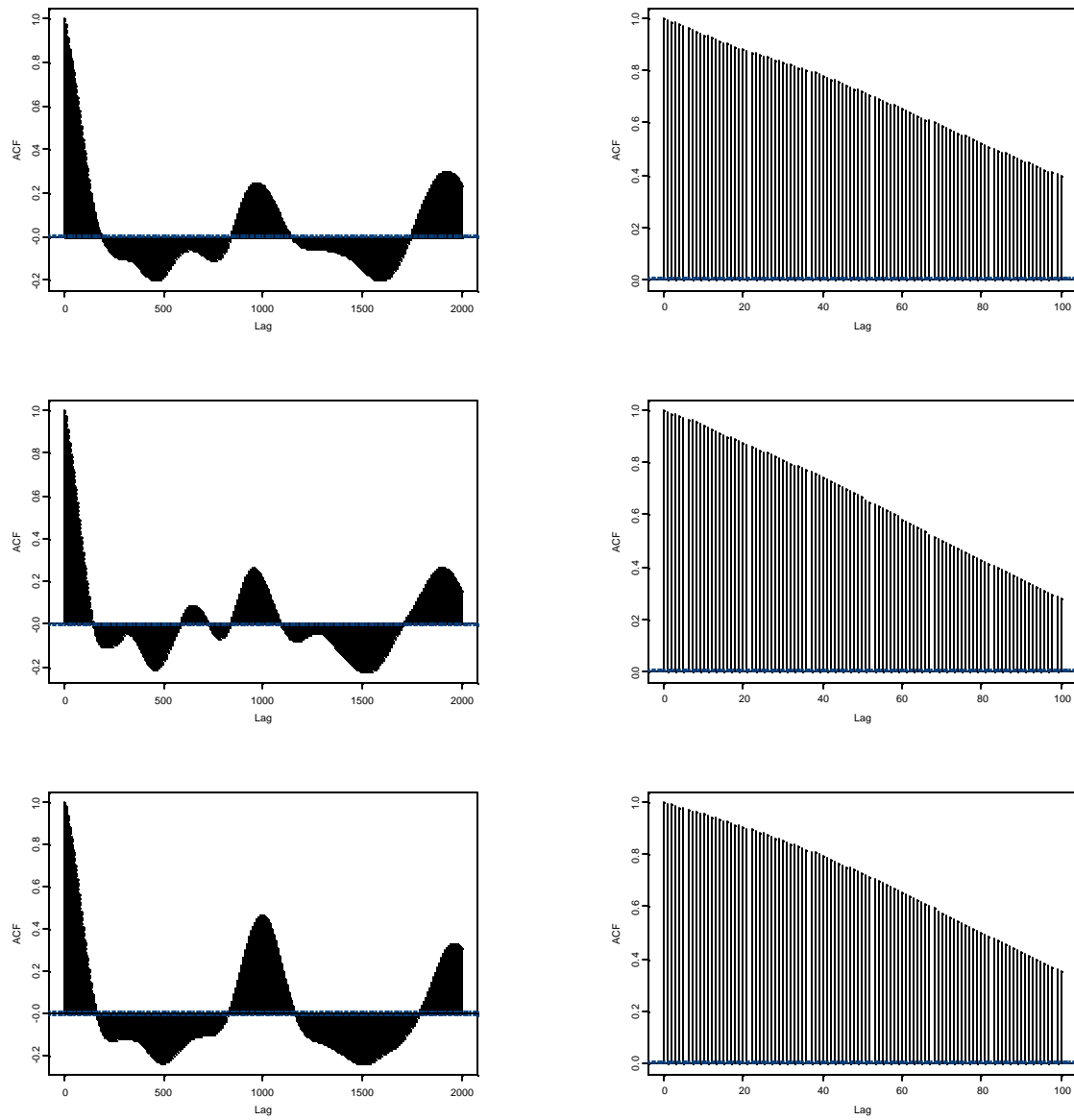


Figure 6.12. Autocorrelation function for packet delays. Top: packet no. 10,000,001 to 10,200,000. Middle: packet no. 12,000,001 to 12,200,000. Bottom: packet no. 14,000,001 to 14,200,000. Left: 2000 lag scale. Right: 100 lag scale. Simulation with 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets.

As shown in Figure 6.10 to 6.12, the autocorrelation functions in the large lag scale (2000 lags) all exhibit some periodic or oscillatory pattern. The autocorrelation functions in the small lag scale (100 lags) all exhibit close to a linear pattern. We conjecture that the autocorrelation function of the packet delay might be related to the correlation of the random process of packet inter-arrival time. That is, when the traffic becomes burstier, the packet inter-arrival time tends to decrease gradually, more packets arrive to the router, and the packet delay gradually increases. When the traffic becomes less bursty the packet inter-arrival time tends to increase gradually, fewer packets arrive to the router, and the packet delay gradually decreases. As we noted in Chapter 4.2, we exclude the lost packets in our analysis of packet delay and such data exclusion may have some non-negligible effects on our delay analysis.

6.3.3. Packet delay jitter

In addition to the delay distribution and the autocorrelation functions, the characteristics of delay jitter are also an important aspect of packet delay. We determine delay jitter by calculating the magnitude of the difference in the delay of two consecutive packets. We analyze the delay jitter by plotting the delay jitter distribution as shown in Figure 6.13 for the simulation with the buffer size equal to 46 packets. As shown in Figure 6.13, most of the delay jitters are within 0.2 milliseconds, although delay jitters larger than 1 millisecond are still possible. The delay jitter has an approximately exponential-like distribution because the distribution in the log scale has a linear-like shape despite some oscillation. The simulation with the buffer size equal to 200 packets yields similar results, which are included in the Appendix D.

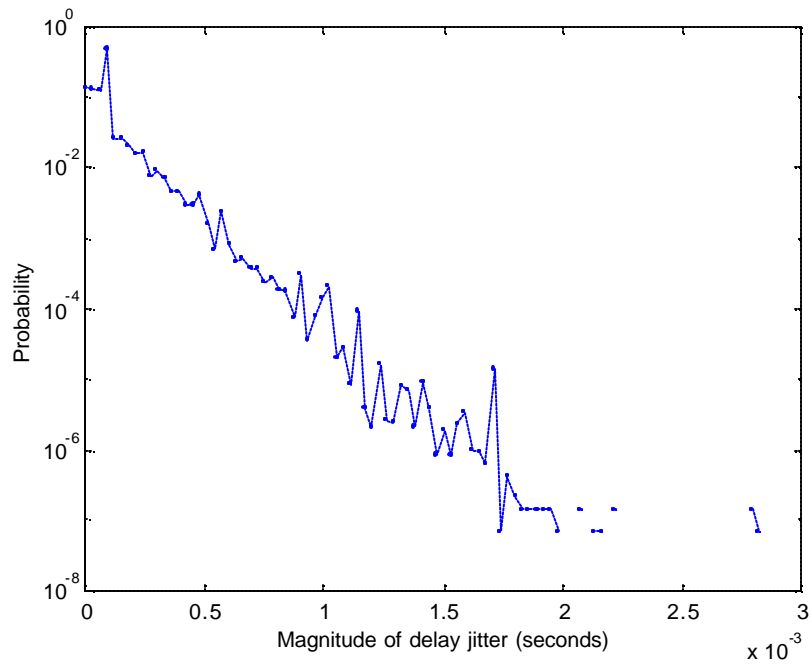
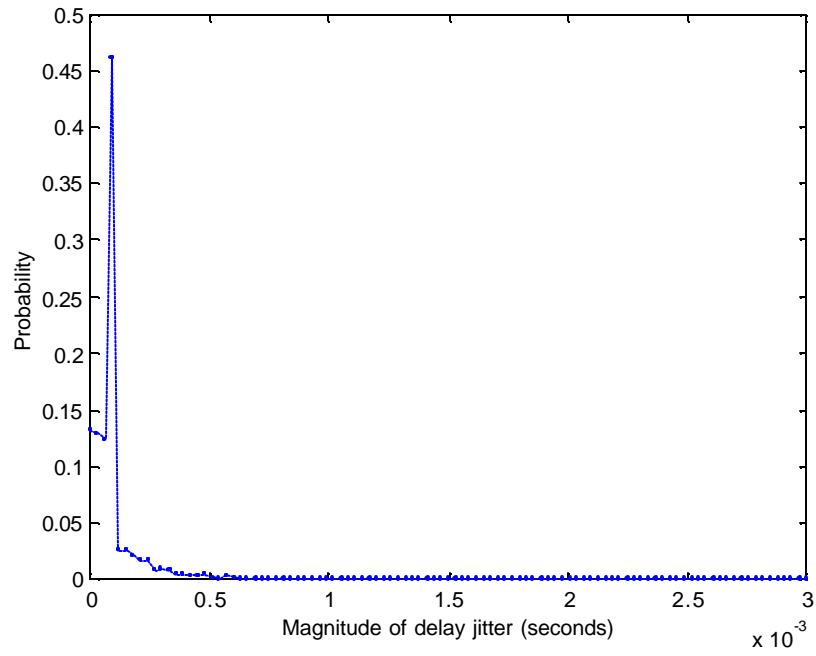


Figure 6.13. Distribution of the magnitude of packet delay jitter, linear scale (top) and log scale (bottom). Simulation with 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets.

6.3.4. Per-flow average packet delay and standard deviation

Lastly, for the packet delay analysis, we investigate the average packet delay and the standard deviation of packet delay for packets from the same flow. Figure 6.14 and 6.15 are the per-flow average packet delay and the per-flow packet delay standard deviation respectively for the simulation with buffer size equal to 46 packets. As shown in Figure 6.14 and 6.15, there is no particular pattern for the average delay and the delay standard deviation. Such observation is expected for FIFO/DropTail queuing mechanism because all flows share the same queue, resulting in the lack of fairness and regulation in the packet delay performance. The simulation with the buffer size equal to 200 packets yields similar results. They are included in the Appendix D.

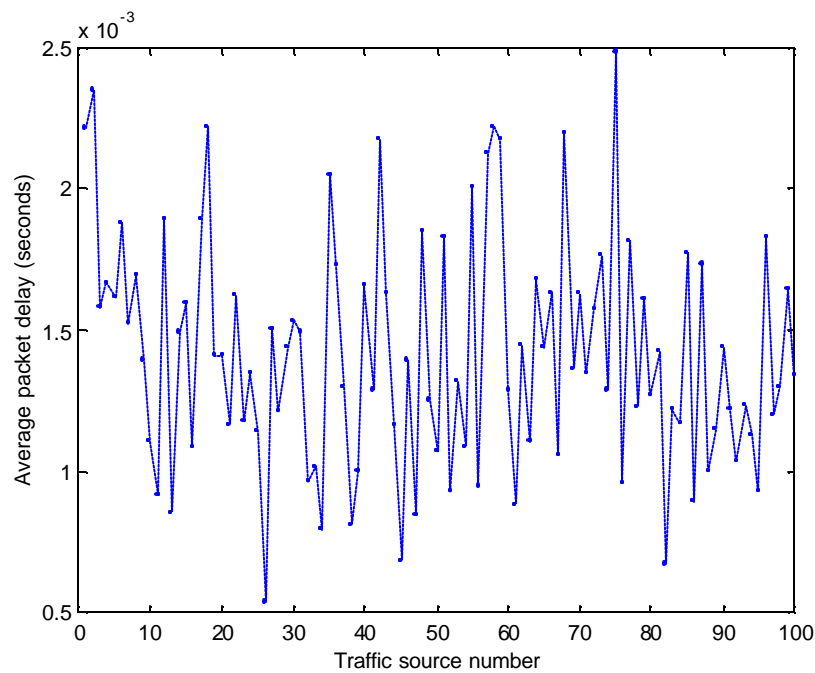


Figure 6.14. Average packet delay for packets from the same flow. Simulation with 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets.

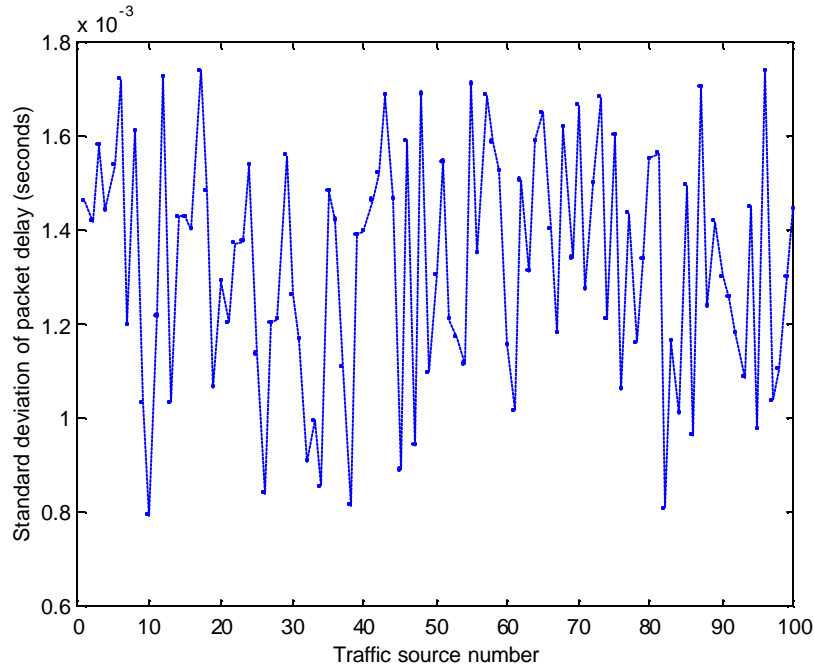


Figure 6.15. Standard deviation of packet delay for packets from the same flow. Simulation with 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets.

6.4. QoS and network performance parameters

In addition to packet loss and packet delay, we investigate four other QoS and network performance parameters: buffer occupancy probability, per-flow traffic load, throughput, and loss rate.

6.4.1. Buffer occupancy probability

Buffer occupancy (buffer size) probability measures how frequently the buffer size is equal to i packets, as defined in Eq. (4.7). Figure 6.16 shows the buffer occupancy probability distribution. The buffer occupancy probability distribution is very similar to the packet delay distribution. For FIFO/DropTail queuing mechanism, because packets are served in the same order as their arrival order and the packets are dropped from the end of the buffer when the buffer is full, the buffer occupancy reflects not only the buffer size but also how long packets have to wait in the buffer. As in the case of the packet

delay distribution, the buffer occupancy probability distribution decays very slowly and even increases rapidly in the region close to the maximum buffer size, indicating high probabilities of full buffer due to frequent heavy network congestion.

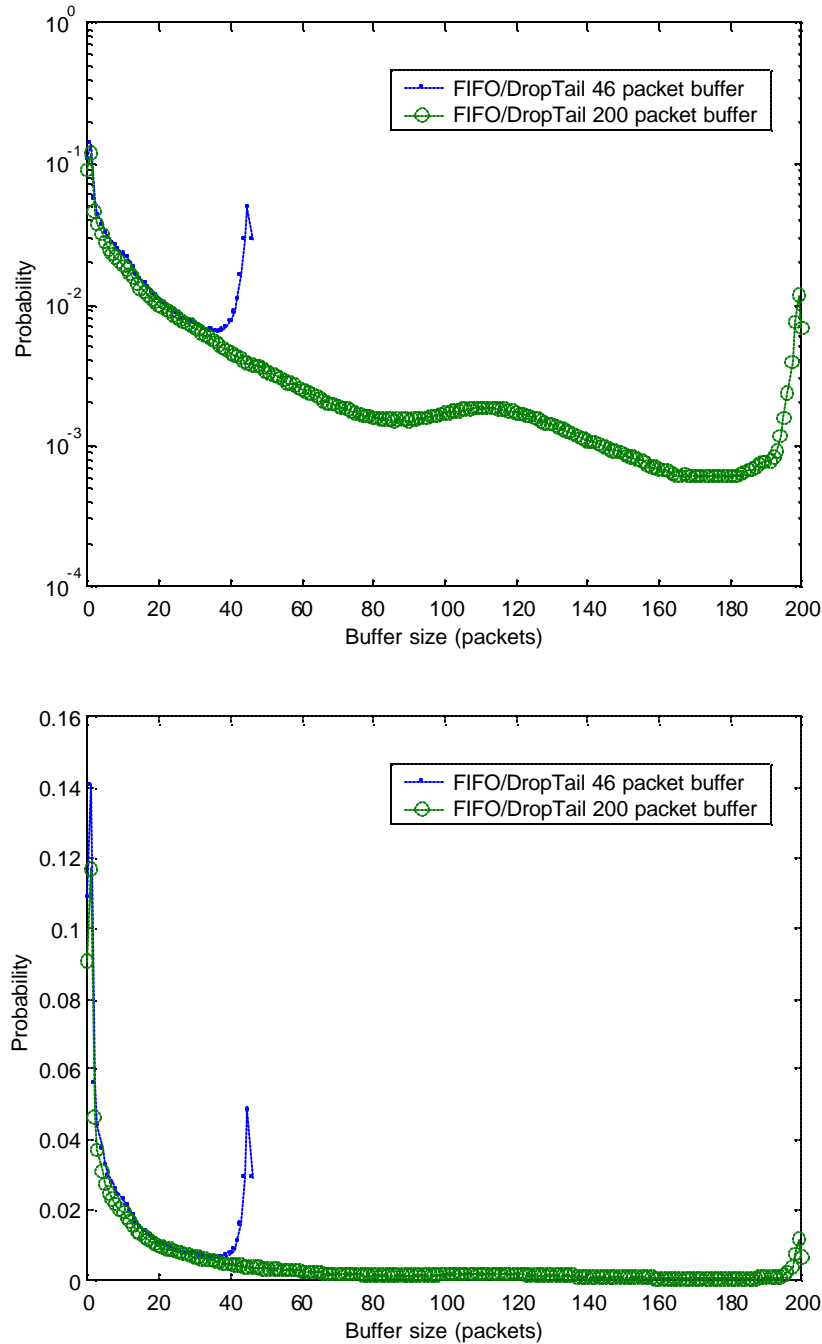


Figure 6.16. Router buffer occupancy probability distribution, linear scale (top) and log scale (bottom). The size of the router buffer is 46 packets. Simulation with 100 traffic sources, using 10 different MPEG traces.

6.4.2. Per-flow traffic load, throughput, and loss rate

In order to assess the fairness of FIFO/DropTail queuing mechanism, we analyze the per-flow traffic load, throughput, and loss rate as defined in Chapter 4.3. Figure 6.17 shows the per-flow traffic load, throughput, and loss rate for each of the 100 traffic sources for the simulation with buffer size equal to 46 packets. As shown in Figure 6.17, although the per-flow throughput is almost identical to the per-flow traffic load, FIFO/DropTail is not a fair queuing mechanism because the per-flow loss rate is not proportional to the per-flow traffic load. For example, source 1 to source 10 have the lowest traffic load (generating the least amount of packets) because their average bit rate is the lowest (as indicated in Table 3.1). However, they are served unfairly because they have more packet losses than many other flows with higher traffic load. The simulation with the buffer size equal to 200 packets yields similar results, which are included in the Appendix D.

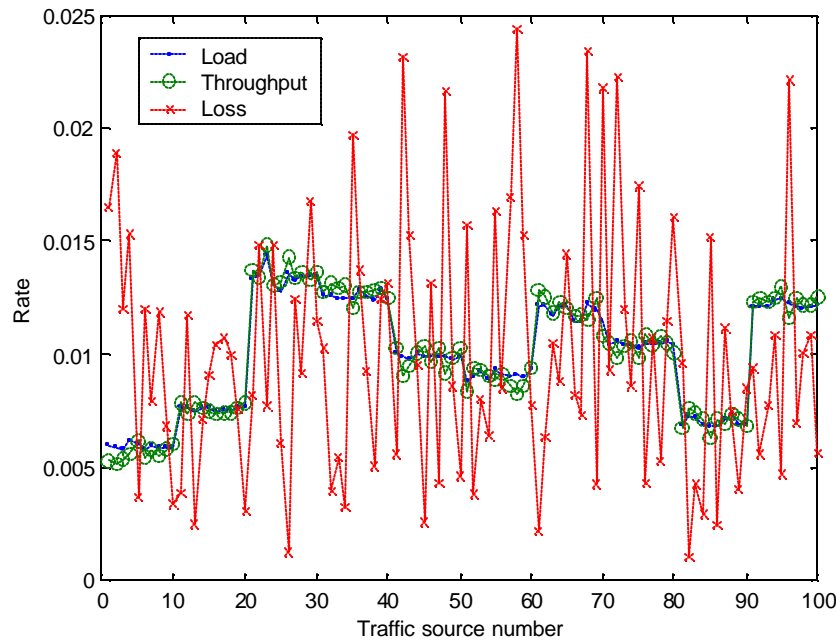


Figure 6.17. Per-flow load, throughput, and loss, calculated with respect to the total traffic load, throughput, and packet loss. Simulation with 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets. The total traffic load, throughput and packet loss are shown in Table 6.2.

Chapter 7

RED, FQ, SFQ, and DRR simulations

In this Chapter, we present the RED, FQ, SFQ, and DRR simulation results. Because the focus of this research is on FIFO/DropTail, the analysis for RED, FQ, SFQ, and DRR is only preliminary and it is not as extensive as the analysis presented in Chapter 6. The objective of the RED, FQ, SFQ, and DRR simulations is to compare how different queuing mechanisms affect the QoS and the network performance. The comparison is separated into two parts: comparison between FIFO/DropTail and RED, and comparison among FQ, SFQ, and DRR. We compare FQ, SFQ, and DRR because their mechanisms are different from that of FIFO/DropTail and RED, and, hence, their simulations require different simulation parameter settings. Therefore, in order to keep the consistency of the analysis, we do not compare FIFO/DropTail and RED directly to FQ, SFQ, and DRR. In addition, due to the ns-2 limitations such as the packet queue constraint mentioned earlier, the functionalities of FQ, SFQ, and DRR cannot be fully exploited in our simulations. Therefore, we only present a survey analysis for these three queuing mechanisms.

7.1. RED

Random early drop (RED) [16] is a congestion avoidance queuing mechanisms; it monitors the average buffer size and drops both the incoming packets and the packets inside the buffer when the congestion starts to build up. RED has many configuration parameters such as the buffer threshold size and the packet drop probability. In our RED simulation we use the basic configuration parameter values documented in [5]. As in the FIFO/DropTail simulations, we use 100 traffic sources, 10 MPEG traces, and 46 and 200 packet buffers in the RED simulation. We compare the performance of RED to FIFO/DropTail by examining per-flow packet loss pattern, delay, and fairness.

Table 7.1 is a summary of the basic statistics of the RED simulation results. As expected, the RED configuration parameters that we used result in more packet losses than FIFO/DropTail. To see how RED's packet drop policy affects the loss pattern, we examine the contribution of loss episodes averaged over all individual flows shown in Figure 7.1. As shown in Figure 7.1, RED has a better per-flow packet loss pattern because the contribution of loss episodes not only decays exponentially but also decays faster than FIFO/DropTail. Therefore, by dropping packets before the buffer is full and by dropping packets from within the buffer instead from the incoming packets, RED is capable of reducing the occurrence of long loss episodes for individual flows. With RED, the end-user video quality may be improved despite some increase of packet losses. In Figure 7.2, we plot the length of the longest loss episode for each traffic flow for the simulation with buffer size equal to 46 packets. Compared to FIFO/DropTail, the longest loss episode for a particular flow tends to be shorter with RED. The simulation with the buffer size equal to 200 packets yields similar results, which are included in the Appendix D.

Table 7.1. Summary of simulation results for the FIFO/DropTail and RED simulation with 100 traffic sources.

Queuing Mechanism	Buffer Size (packets)	Number of Traffic Sources	Number of Packets Arrived	Number of Packets Transmitted	Number of Packets Dropped	Traffic Load (%)	Packet Loss Rate (%)
FIFO/DropTail	46	100	14997184	14128986	868186	82.24	5.789
FIFO/DropTail	200	100	14997184	14796732	200499	82.24	1.337
RED	46	100	14997184	14128980	868192	82.24	5.789
RED	200	100	14997184	14485653	511519	82.24	3.411

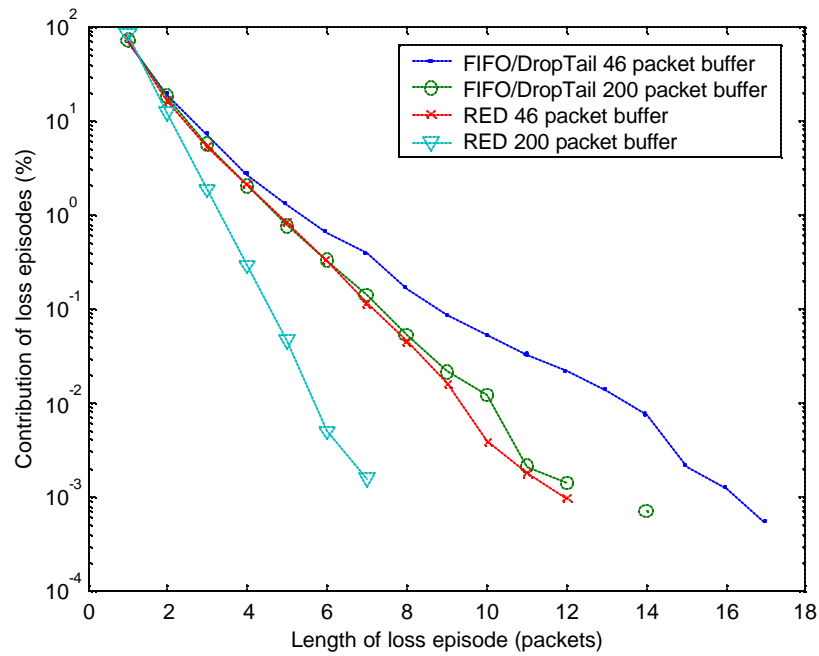
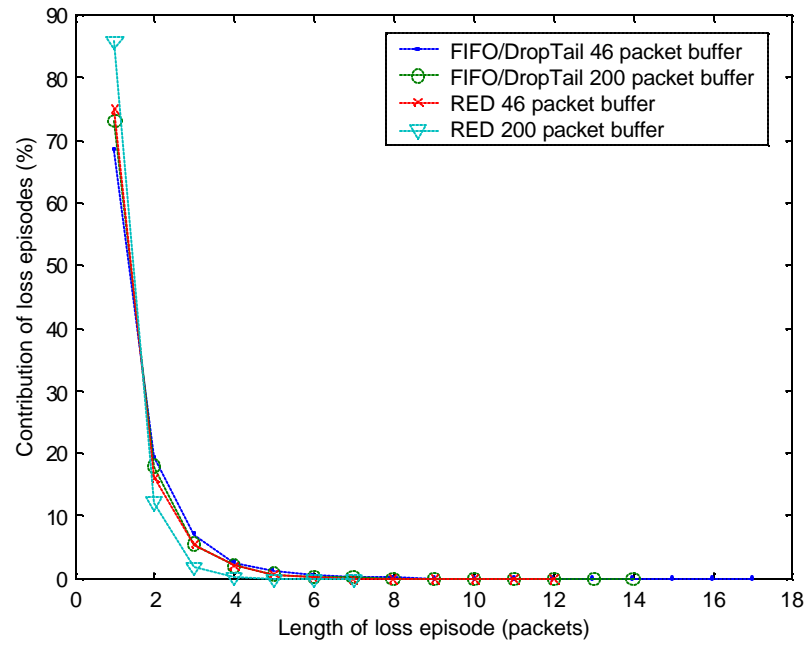


Figure 7.1. The contribution of loss episodes of various lengths to the overall number of loss episodes, averaged over all individual flows (per-flow loss), linear scale (top) and log scale (bottom). Simulation with FIFO/DropTail and RED, 100 traffic sources, using 10 different MPEG traces.

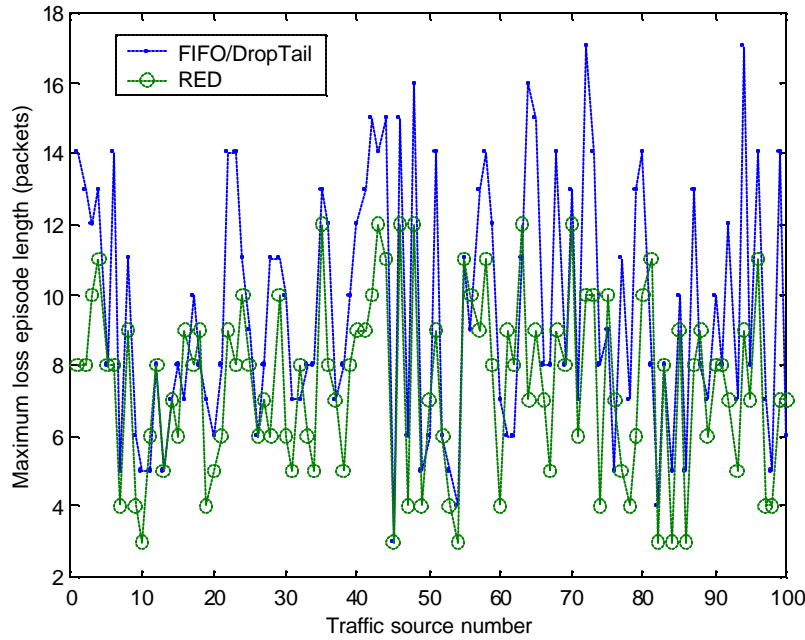


Figure 7.2. The length of the longest loss episode for each flow. Simulation with FIFO/DropTail and RED, 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets.

To characterize packet delay, we analyze the packet delay distribution shown in Figure 7.3. In the small and medium packet delay region, RED and FIFO/DropTail have identical distribution. In the packet delay region larger than 4.0 milliseconds (for 46 packet buffer simulation) and 19.0 milliseconds (for 46 packet buffer simulation), the RED distribution decays rapidly, whereas the FIFO/DropTail distribution increases rapidly. The maximum packet delay for RED is also bounded by the maximum packet delay for FIFO/DropTail. RED has much smaller probabilities of packet delay close to the maximum value because when a packet arrives to the buffer whose size is close to full, RED drops packets from within the buffer to reduce the buffer size. As a result, the newly arrived packet will not experience long queuing delay. Because of the absence of the rapid increase in delay probability near the maximum packet region, RED yields better delay characteristics than FIFO/DropTail in the heavily congested network.

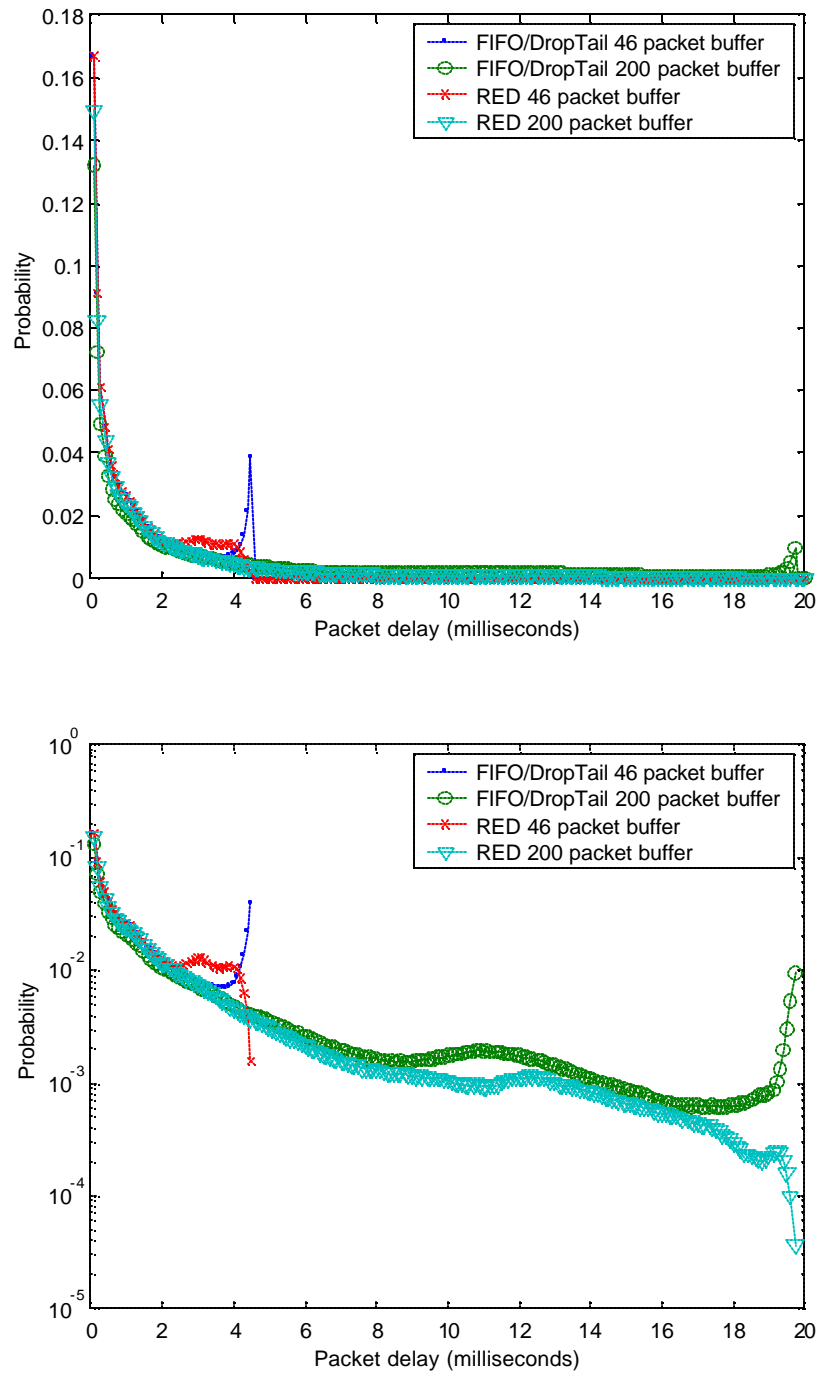


Figure 7.3. Probability distribution of packet delay for all delivered packets, linear scale (top) and log scale (bottom). Simulation with FIFO/DropTail and RED, 100 traffic sources, using 10 different MPEG traces.

In addition to the packet delay distribution, we also plot in Figure 7.4 and 7.5 the average packet delay and the standard deviation respectively for each flow for the simulation with the buffer size equal to 46 packets. As shown in Figure 7.4 and 7.5, RED has smaller average delay and standard deviation than FIFO/DropTail. However, as in the case of FIFO/DropTail, because all flows share the same queue, there is no particular pattern for the average delay and the delay standard deviation, reflecting the lack of fairness and regulation in the packet delay performance. The simulation with the buffer size equal to 200 packets yields similar results, which are included in the Appendix D.

In order to compare the fairness of RED, we examine the per-flow traffic load, throughput, and loss rate for the simulation with the buffer size equal to 46 packets. We plot the results in Figure 7.6. Our simulation results show that RED exhibits small degrees of fairness, as the loss rate is roughly proportional to the traffic load for individual flows. Even RED drops packet randomly, its packet drop policy can still improve the fairness of packet drops. For example, the traffic flows with higher bit rate and burstiness occupy more buffer space than flows with lower bit rate and burstiness. When RED drops packet from within the buffer to reduce the buffer size to avoid congestion, packets from flows with higher traffic load are more likely to be dropped than packets from flows with smaller traffic load because most of the buffer is occupied by packets from flows with higher traffic load. The simulation with the buffer size equal to 200 packets yields similar results, which are included in the Appendix D.

In summary, compared to FIFO/DropTail, RED has a better per-flow packet loss pattern because the contribution of loss episodes decays faster. RED also has better packet delay characteristics because the probability of maximum packet delay is much smaller than in the case of FIFO/DropTail. The random drop mechanism in RED also results in better fairness where the loss rate is roughly proportional to the traffic load for individual traffic flows.

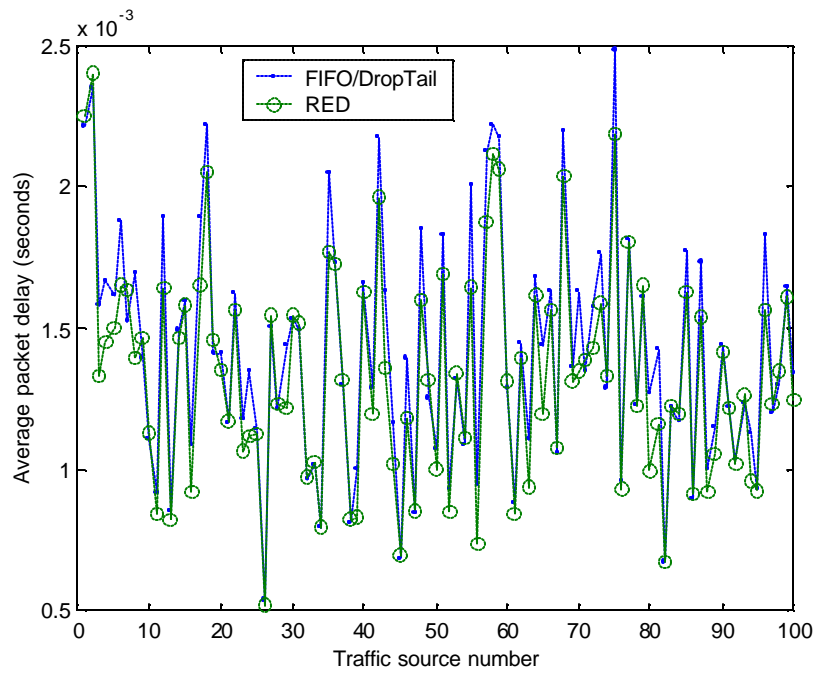


Figure 7.4. Average packet delay for packets from the same flow. Simulation with FIFO/DropTail and RED, 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets.

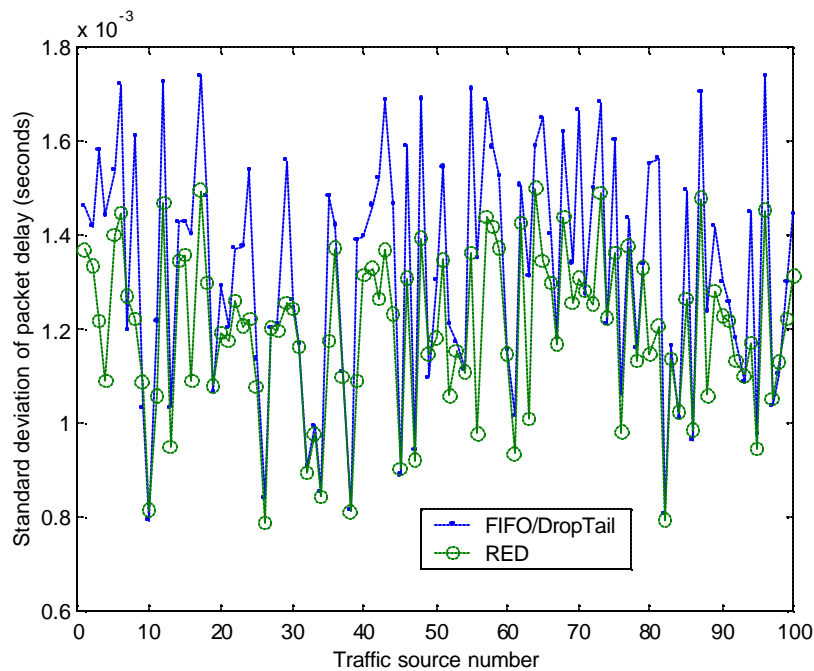


Figure 7.5. Standard deviation of packet delay for packets from the same flow. Simulation with FIFO/DropTail and RED, 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets.

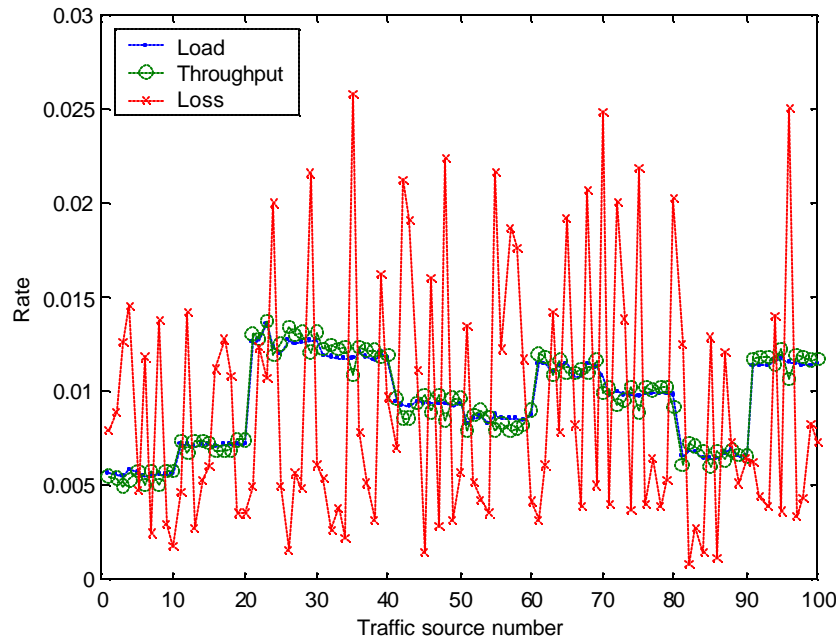


Figure 7.6. Per-flow load, throughput, and loss, calculated with respect to the total traffic load, throughput, and packet loss. Simulation with RED, 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets.

7.2. FQ, SFQ, and DRR

Fair queuing (FQ) [30] approximates the bit-by-bit round robin scheduling scheme by using the finish-time equation, Eq. (4.1) to achieve nearly perfect throughput fairness for all traffic flows to the router. FQ allocates one sub-queue for each flow. For example, let flow A and flow B have the same average bit rate, with flow A using 1000-byte packets and flow B using 10-byte packets. The router uses different sub-queues to store packets from flow A and flow B. When the router receives one packet from flow A first, immediately followed by 100 packets from flow B, the flow B packets will have shorter finish-time than the one flow A packet because of their small packet size. Consequently, the flow B packets will be delivered before the flow A packets. Therefore, small packets will not be delayed by large packets, and flow A and flow B will have the same throughput on average. Stochastic fair queuing (SFQ) [28] is simply a packet-by-packet round robin scheduling scheme. SFQ uses hashing to allocate sub-queues for traffic

flows. Just as SFQ, deficit round robin (DRR) [42] uses hashing to allocate sub-queues for traffic flows. DRR uses a simpler algorithm that approximates the throughput fairness of FQ.

In ns-2, the implementation of FQ and SFQ is not identical to the original FQ and SFQ algorithms. For FQ, ns-2 uses a slightly different finish-time equation, and when the buffer is full incoming packets are dropped as in DropTail. For SFQ, ns-2 does not use the longest sub-queue packet drop policy (packets from the longest sub-queue are dropped when the buffer is full). Instead, each sub-queue is allocated the same amount of storage space, called the fair-share. For example, if the buffer size is 200 packets and there are 100 traffic flows to the router, then each sub-queue is allowed to hold at most two packets when the buffer is close to full. In addition, in ns-2, both FQ and SFQ use packet queues as in FIFO/DropTail, whereas DRR uses byte queues.

As mentioned in Chapter 3.2, in order to maintain the consistency of simulation, we use a constant packet size for MPEG traffic. Although using the constant packet size results in a fair comparison of buffer sizes among FQ, SFQ, and DRR, the constant packet size limits functionalities of FQ and DRR, whose main advantage is to provide throughput fairness when traffic flows generate packets with different sizes. Therefore our simulation and analysis of FQ and DRR do not reveal their full performance. Nevertheless, we present our simplified results and analysis as a preliminary study and comparison of these queuing mechanisms and their effects on QoS in video traffic.

In our FQ, SFQ, and DRR simulations, we use 100 traffic sources and 10 MPEG traces. Instead of the 46 and 200 packet buffer as in FIFO/DropTail and RED simulation, we set the maximum buffer size to only 200 packets. Because FQ, SFQ, and DRR all utilize per-flow queuing, using a buffer size smaller than the potential number of active traffic flows will significantly degrade their performance. For example, if the buffer size is only 46 packets, packets from many traffic flows will collide into the same sub-queues, reducing the fairness of the queuing mechanism. Therefore, we select our buffer size to be 200 packets so that the sub-queue for each flow can ideally hold two packets. As in

the comparison between FIFO/DropTail and RED, we compare the performance of FQ, SFQ, and DRR by examining per-flow packet loss pattern, delay, and fairness.

Table 7.2 is a summary of the basic statistics of the FQ, SFQ, and DRR simulation results. FQ and DRR have the same loss rate because they both drop packets only when the entire buffer is full. SFQ has a much higher loss rate because each sub-queue is allowed to hold at most two packets when the buffer is close to be full (i.e., packets can be dropped before the entire buffer is full). Therefore, bursty traffic sources will lose a considerable amount of packets due to the buffer fair-share policy. Figure 7.7 shows the contribution of loss episodes, averaged over all individual flows, reflecting the per-flow packet loss characteristics.

Table 7.2. Summary of simulation results for the FQ, SFQ, and DRR simulations with 100 traffic sources.

Queuing Mechanism	Number of Traffic Sources	Number of Packets Arrived	Number of Packets Transmitted	Number of Packets Dropped	Traffic Load (%)	Packet Loss Rate (%)
FQ	100	14997184	14796723	200449	82.24	1.337
SFQ	100	14997184	13898322	1098852	82.24	7.327
DRR	100	14997184	13898322	1098852	82.24	1.337

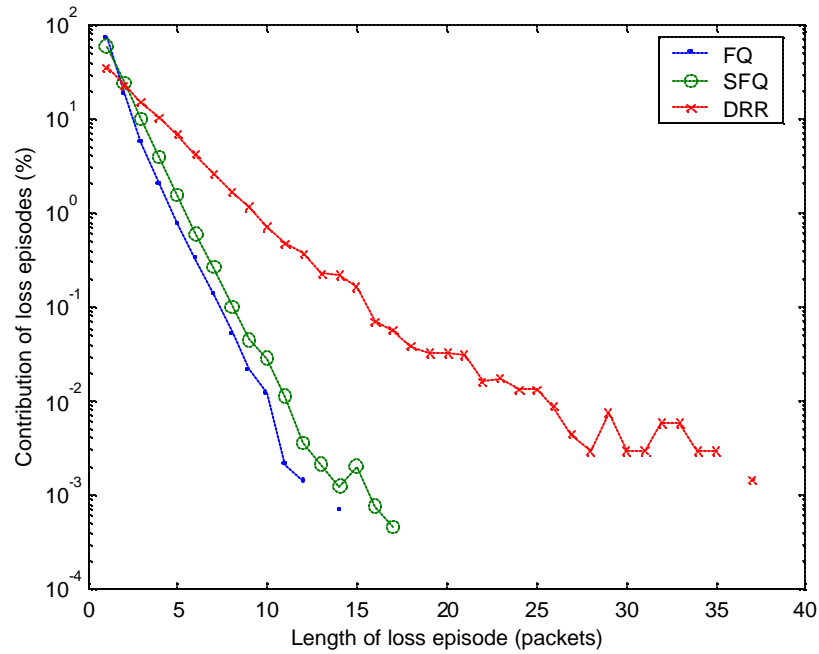
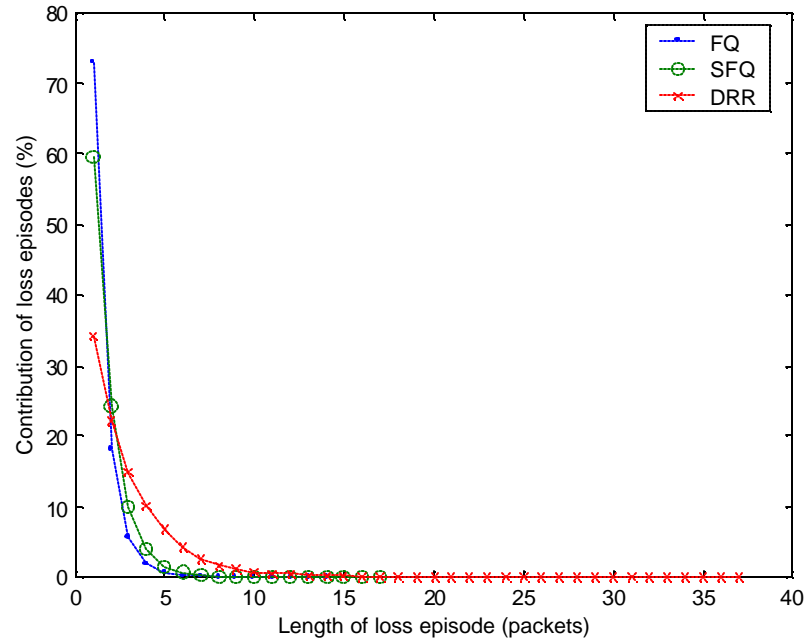


Figure 7.7. The contribution of loss episodes of various lengths to the overall number of loss episodes, averaged over all individual flows (per-flow loss), linear scale (top) and log scale (bottom). Simulation with FQ, SFQ, and DRR, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

As shown in Figure 7.7, FQ, SFQ, and DRR all exhibit approximately exponentially decaying patterns in the per-flow loss episode contribution (because of the linear-like pattern in the log scale distribution). The pattern for FQ is the same as FIFO/DropTail because FQ drops packets in the same way as FIFO/DropTail. For SFQ, the distribution decays slightly more slowly than FQ because its fair-share packet drop policy results in much more packet losses than FQ, as shown in Table 7.2. For DRR, the distribution decays much more slowly, although it is still approximately exponential. The higher contribution of long loss episodes in DRR is the result of DRR's longest sub-queue packet drop policy. For example, when a traffic source sends a bursty packet stream to the router, resulting a full buffer, its sub-queue will most likely have the longest length. To accommodate the incoming packets, DRR will drop packets from the end of the same sub-queue until this sub-queue is no longer the longest sub-queue. Therefore a long sequence of packets from the same sub-queue will be dropped and make up a long loss episode. Such occurrence of long loss episodes, however, indicates the better fairness in DRR. When a traffic flow is very bursty or has very high traffic load, DRR punishes it when congestion occurs by dropping a long sequence of its packets from the buffer, resulting in a long loss episode.

For SFQ, on the other hand, although each sub-queue can only hold two packets when the buffer is congested, long loss episodes do not occur often. For example, when a traffic source sends a bursty packet stream to the router, its sub-queue will reach its maximum size (the fair-share) immediately. But because SFQ drops only incoming packets rather than packets already stored in the sub-queue as in DRR, some packets from the bursty incoming stream can still be accepted when the router delivers packets from the same sub-queue. As a result, long loss episodes seldom occur. Figure 7.8 shows the longest loss episode for each flow for FQ, SFQ, and DRR simulation. The effect of different packet drop policy is apparent in Figure 7.8.

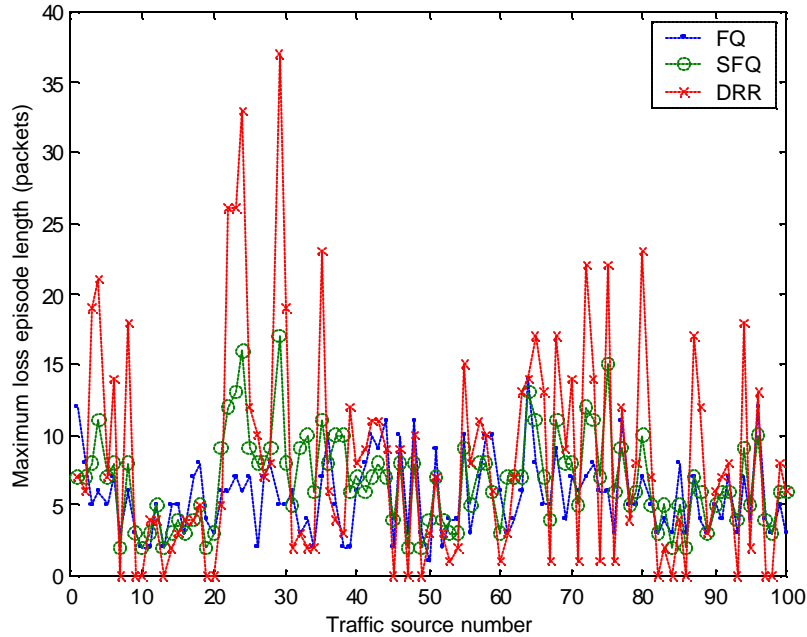


Figure 7.8. The length of the longest loss episode for each flow. Simulation with FQ, SFQ, and DRR, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

To characterize packet delay for FQ, SFQ, and DRR, we analyze the packet delay distribution in Figure 7.9, and the length of the longest packet delay for each flow in Figure 7.10. As shown in Figure 7.9, the packet delay distribution for FQ is very similar to FIFO/DropTail, where the probability distribution increases rapidly near the maximum packet delay region. The maximum packet delay for FQ is equal to the maximum packet delay for FIFO/DropTail, because in our simulation FQ is simply a packet-by-packet round robin scheduling scheme due to the constant packet size. With the given router capacity, a 200 packet buffer and a 552 byte packet size result in a maximum packet delay approximately equal to 20 milliseconds, as shown in Figure 7.10.

For DRR, extremely long packet delays can occur as shown in Figure 7.10. The packet delay probability of DRR at 20 milliseconds in Figure 7.9 is the cumulative probability for packet delay larger than 20 milliseconds. Like the occurrence of very long loss episodes, the occurrence of very long packet delay is also caused by the DRR's

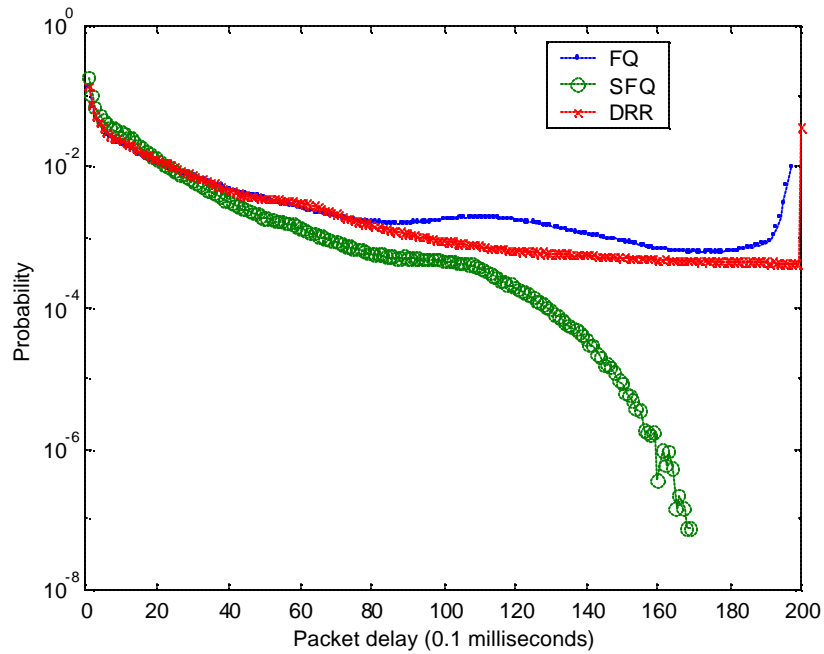
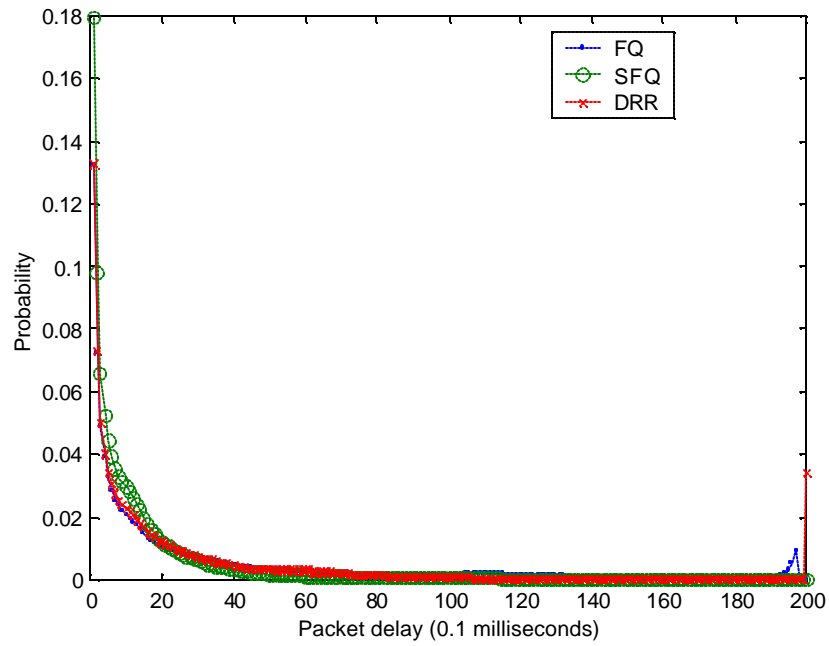


Figure 7.9. Probability distribution of packet delay for all delivered packets, linear scale (top) and log scale (bottom). Simulation with FQ, SFQ, and DRR, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

packet drop policy. For example, assume the buffer is not full and there is one long sub-queue with all other sub-queues being very short. Very long packet delay can occur for packets stored at the end of the longest sub-queue because they will be served only after packets ahead of them and all packets from the shorter sub-queues are served. Furthermore, all newly arrived packets to the shorter sub-queues, assuming they always have shorter queue length, will also have to be served first. However, as in the per-flow loss episode characteristics, the occurrence of very long packet delays indicates better fairness of DRR as well. Very long packet delays will mostly happen to the traffic flow with very high traffic load or high degrees of burstiness.

For SFQ, the packet delay distribution decays faster than for both FQ and DRR, especially in the long packet delay region, as shown in Figure 7.9. In addition, the longest packet delay is bounded by the maximum packet delay for FQ, as shown in Figure 7.10. The packet delay for SFQ tends to be shorter because, when the router is congested, each sub-queue usually can have at most two packets and the queuing delay is short. The worst packet delay occurs when one traffic source is very bursty and all the other sources are inactive. Nevertheless, the worst packet delay is still bounded by the maximum packet delay of FQ.

In Figure 7.11 and 7.12, we plot the average packet delay and the standard deviation of packet delay for each flow. Again, SFQ has the best delay characteristics, with the smallest average packet delay and standard deviation, at the cost of more packet losses. DRR, on the other hand, has the worst delay characteristics, with the largest average packet delay and standard deviation.

Lastly, we compare the fairness of FQ, SFQ, and DRR by plotting the per-flow traffic load, throughput, and loss rate as shown in Figure 7.13, 7.14, and 7.15. As shown in Figure 7.14, FQ lacks fairness because, when the buffer is full, it drops packets in the way same as DropTail. The loss rate is not proportional to traffic load as the result of the unfair packet drop policy in DropTail. As shown in Figure 7.14, SFQ has better fairness: the loss rate is roughly proportional to traffic load. Because of the fair-share buffer

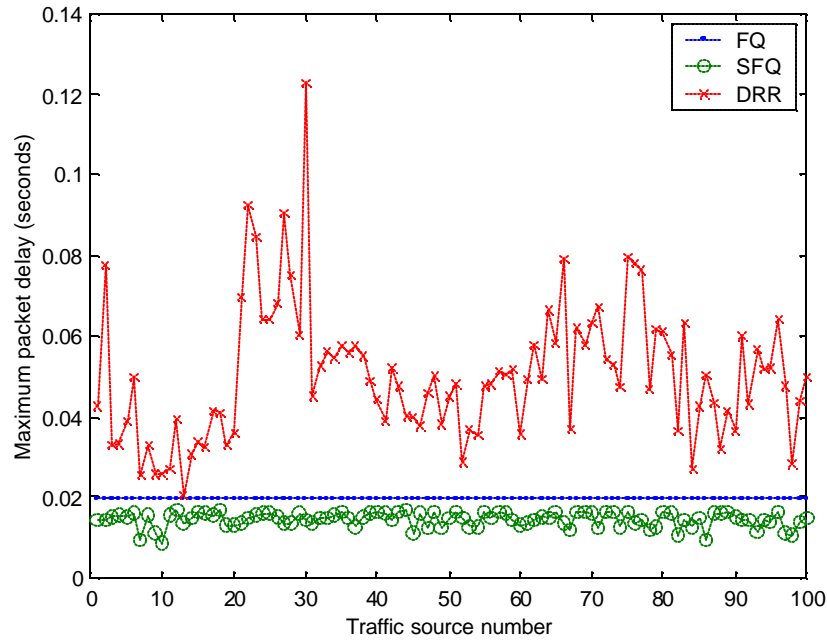


Figure 7.10. The length of the longest packet delay for each flow. Simulation with FQ, SFQ, and DRR, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

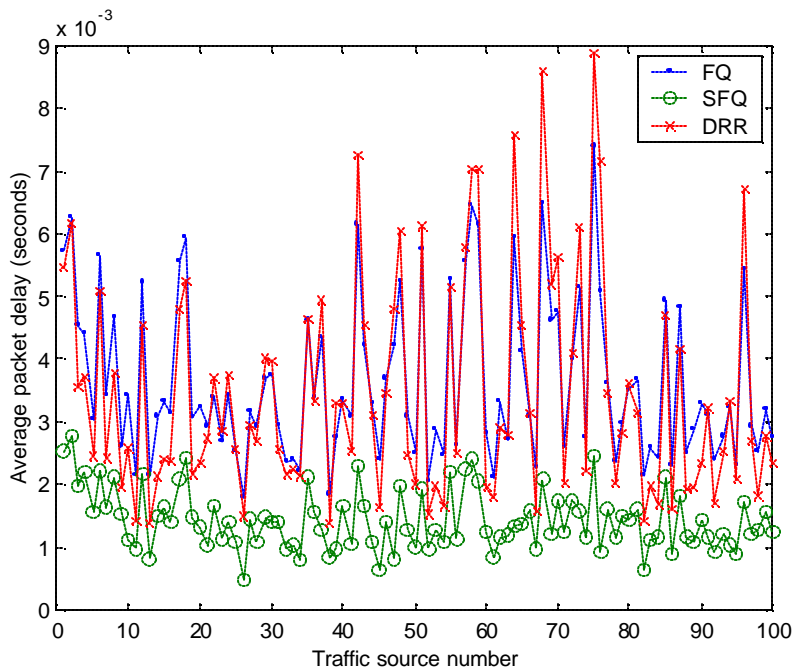


Figure 7.11. Average packet delay for packets from the same flow. Simulation with FQ, SFQ, and RED, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

policy in SFQ, all flows have the same amount of buffer space when the router is congested. Flows with high traffic load, therefore, have more packet losses than flows with lower traffic load. DRR, as shown in Figure 7.15, has the best fairness. The longest sub-queue packet drop policy in DRR punishes bursty flows when congestion occurs. Bursty flows will have the longest sub-queue when the router is congested and, therefore, packets are always dropped from their sub-queues.

In summary, in terms of the per-flow packet loss, DRR has the worst characteristics because its distribution decays very slowly, although it is still exponential. In terms of packet delay, DRR also has the worst characteristics because extremely long packet delays occur and their cumulative probability is non-negligible. On the other hand, SFQ has the best packet delay characteristics at the cost of more packet losses; the packet delay is mostly short and it is bounded by the maximum buffer size. In terms of fairness, FQ is unfair because of its DropTail packet drop policy. DRR has the best fairness: the loss rate for each flow is proportional to its burstiness and traffic load, and bursty flows and flows with high traffic load will experience longer packet delays and longer loss episodes.

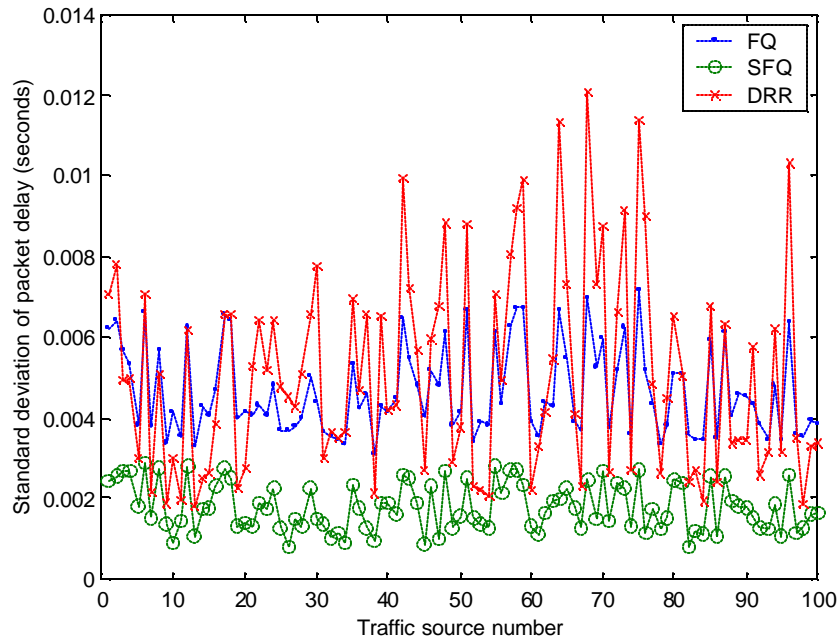


Figure 7.12. Standard deviation of packet delay for packets from the same flow. Simulation with FQ, SFQ, and DRR, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

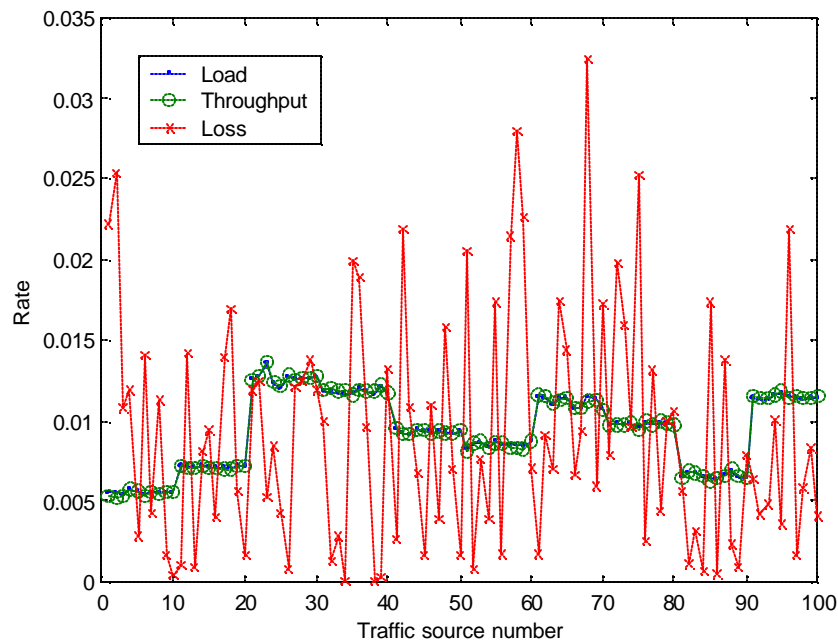


Figure 7.13. Per-flow load, throughput, and loss, calculated with respect to the total traffic load, throughput, and packet loss. Simulation with FQ, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

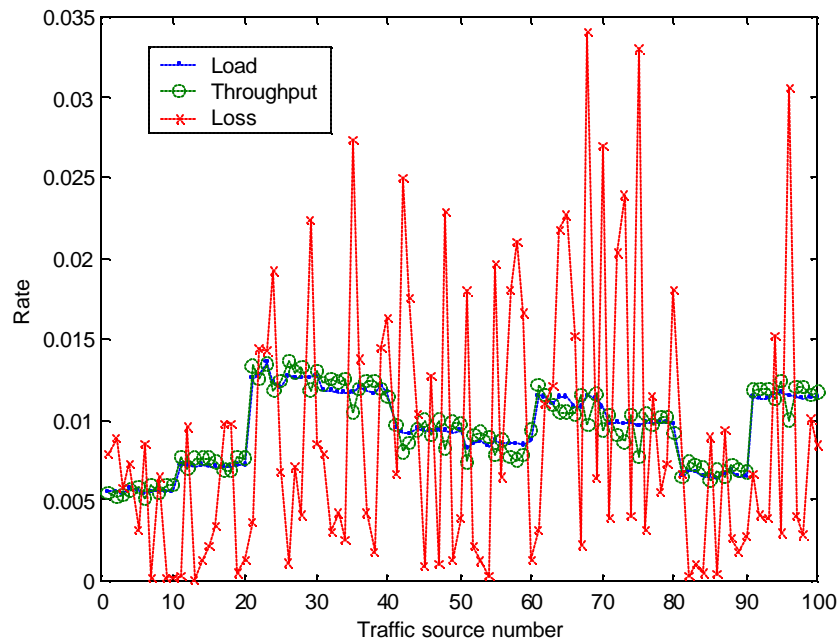


Figure 7.14. Per-flow load, throughput, and loss, calculated with respect to the total traffic load, throughput, and packet loss. Simulation with SFQ, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

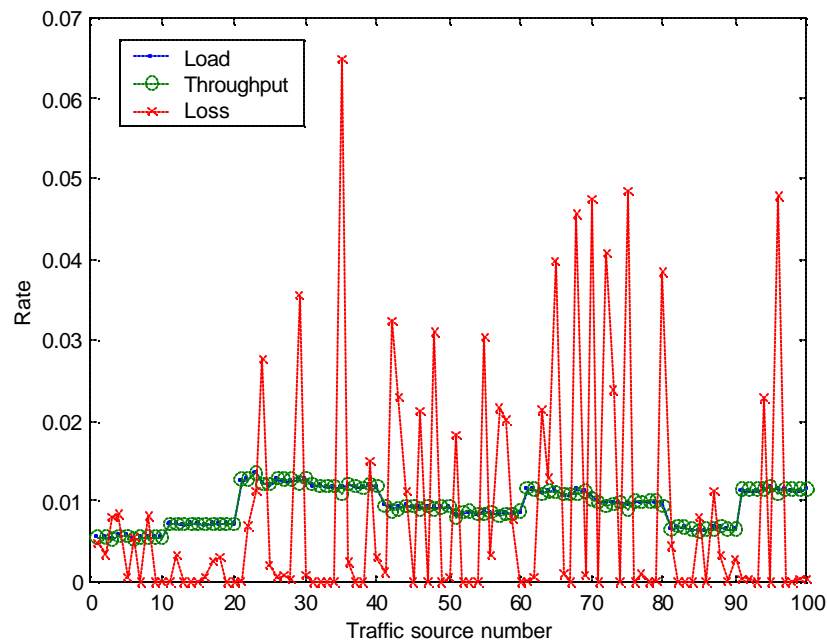


Figure 7.15. Per-flow load, throughput, and loss, calculated with respect to the total traffic load, throughput, and packet loss. Simulation with DRR, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

Chapter 8

Conclusions and future work

In this research, we used computer simulation to analyze the QoS in IP networks with video traffic based on the studies in [5], [26], and [52]. We used the ns-2 network simulator to simulate a network topology that mimics a video server in IP networks. We used genuine MPEG- video traces to generate video traffic transmitted over IP networks using UDP. The selected MPEG-1 video traces exhibit medium to high degrees of self-similarity, a statistical property widely found in variable bit rate video and in the Internet traffic. The use of genuine video trace with self-similarity helps the simulations to more accurately capture the self-similar characteristics in the real network traffic. The main objective of our research is to simulate the video traffic with self-similarity and analyze how video traffic affects various QoS parameters.

We focused our simulation and analysis on the FIFO/DropTail queuing mechanism. We simulated a heavily congested network scenario to capture the QoS performance. We characterized packet loss using packet loss episodes. Our simulation results showed that the contribution of loss episodes of the aggregate packet loss at the router decays slowly, whereas the contribution of loss episodes of the per-flow packet loss decays faster and in an exponential-like pattern. In addition to packet loss, we analyzed packet delay using the packet delay distribution and autocorrelation function. Our results showed that the packet delay distribution not only decays very slowly, but also exhibits a rapid increase pattern near the tail of the distribution. The autocorrelation function of packet delay exhibits some degree of periodicity and oscillation in the large lag scale, and close to a linear decay pattern in the small lag scale. We also found that the delay jitter distribution has an exponential-like decay pattern.

In addition to the FIFO/DropTail queuing mechanism, we also experimented with other queuing mechanisms in our simulation. Our goal is to compare the performance of different queuing mechanisms and analyze their impact on QoS. We compared RED to

FIFO/DropTail, and found that RED, at the cost of some increase in loss rate, resulted in better QoS characteristics because of its different packet drop policy. We also compared FQ, SFQ, and DR, but only among themselves due to their distinct functionalities and the limitations of ns-2. Our preliminary studies showed that SFQ, at the cost of a large increase in loss rate, resulted in the best QoS characteristics. DRR, on the other hand, results in the best fairness despite the worst QoS characteristics.

We hope that from the insight of our QoS analysis a more accurate QoS model for video traffic could be developed to help design future networks with improved performance in the presence of video traffic. Our studies of QoS, although detailed, are still incomplete. Future extensions of this research includes enhancing ns-2 to permit more accurate simulations, analysis of the correlation between packet loss and delay, characterization of packetized traffic, investigation of the effect of time-scales, and better understanding of the effect of self-similarity and LRD on the network performance.

Appendix A.

Aggregate packet loss process for simulations with a single MPEG trace

As mentioned in Section 6.2.1, we observe similar patterns for the aggregate packet loss process in most of our simulations. In this section, we present several sample results for the aggregate packet loss process. We use only one MPEG trace in the simulation. The simulation configuration is the same as the one mentioned in Chapter 6.2.1: 100 traffic sources, 46 packet buffer, FIFO/DropTail queuing mechanism, and 30 minute simulation. Table A1 is a summary of the simulation results. Figures A1 to A4 show the aggregate loss process for four simulations, each with a different MPEG trace.

Table A1. Summary of simulation results for the FIFO/DropTail simulation with various single MPEG traces.

MPEG Trace	Number of Packets Arrived	Number of Packets Transmitted	Number of Packets Dropped	Traffic Load (%)	Packet Loss Rate (%)
Terminator 2	11330442	11074880	255554	62.14	2.255
Simpsons	18961695	15773222	3188436	103.99	16.82
Jurassic Park 1	13513313	12399446	1113867	74.11	8.243
Star Wars	10487413	10390351	97061	57.52	0.9255

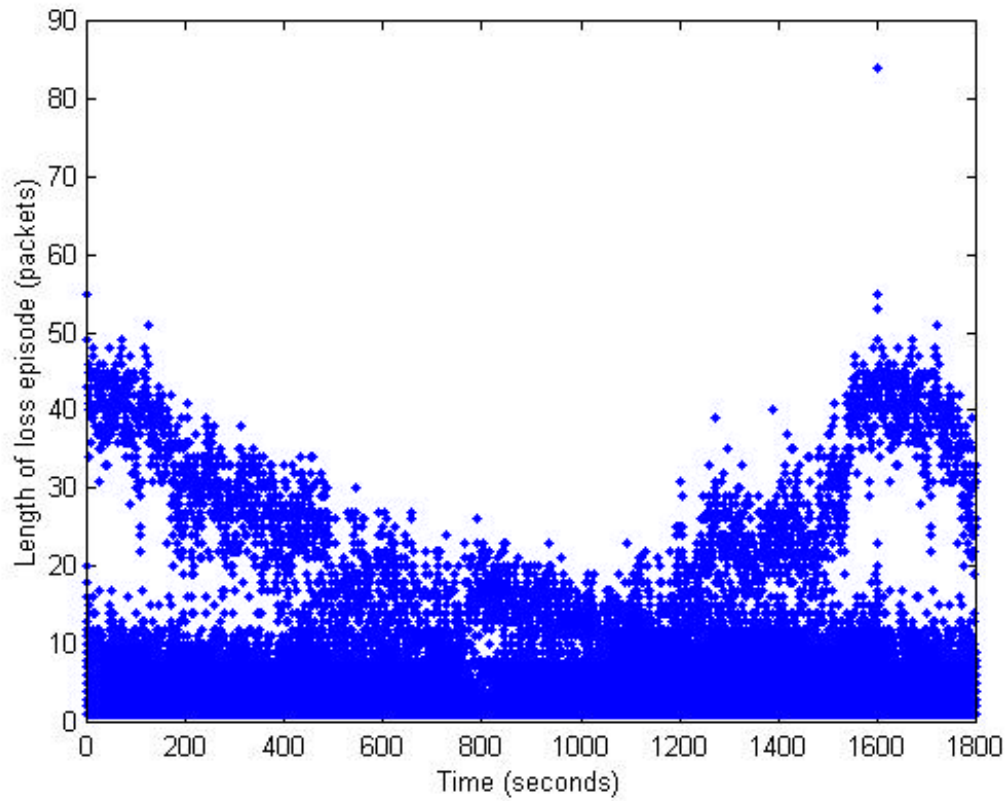


Figure A1. Aggregate packet loss process. Simulation with 100 traffic sources, using one MPEG trace (Terminator 2). Each point in the graph corresponds to a particular loss episode. The occurrence time for each loss episode is the occurrence time of the first lost packet in the loss episode.

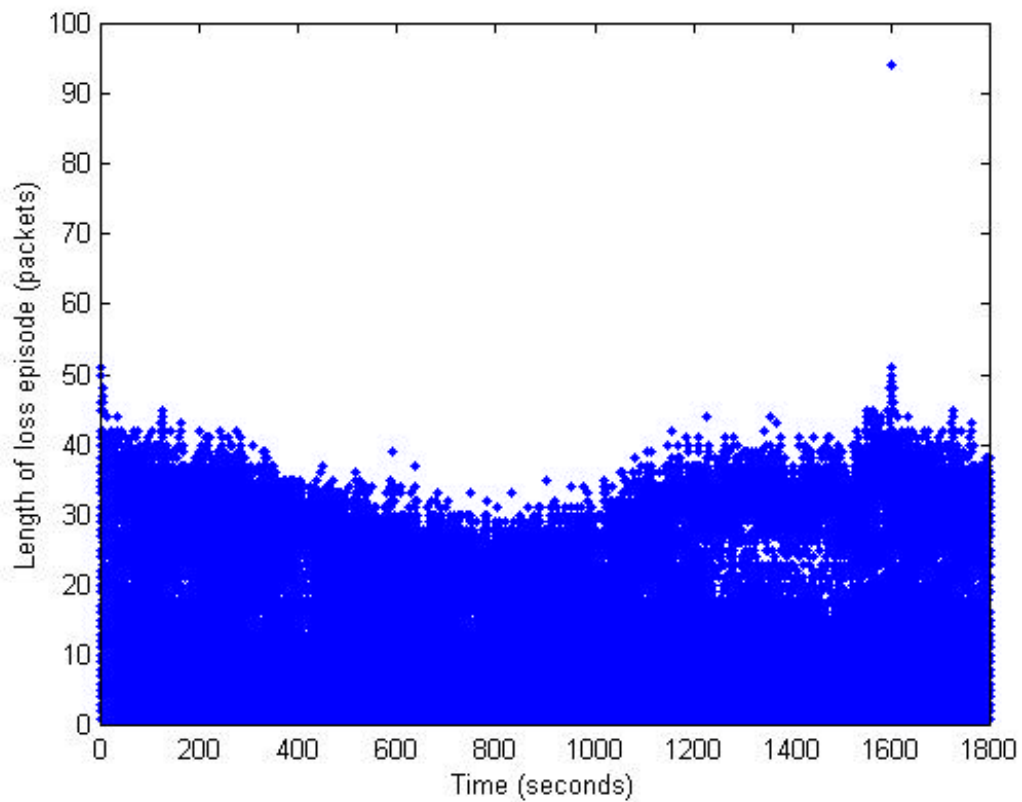


Figure A2. Aggregate packet loss process. Simulation with 100 traffic sources, using one MPEG trace (Simpsons). Each point in the graph corresponds to a particular loss episode. The occurrence time for each loss episode is the occurrence time of the first lost packet in the loss episode.

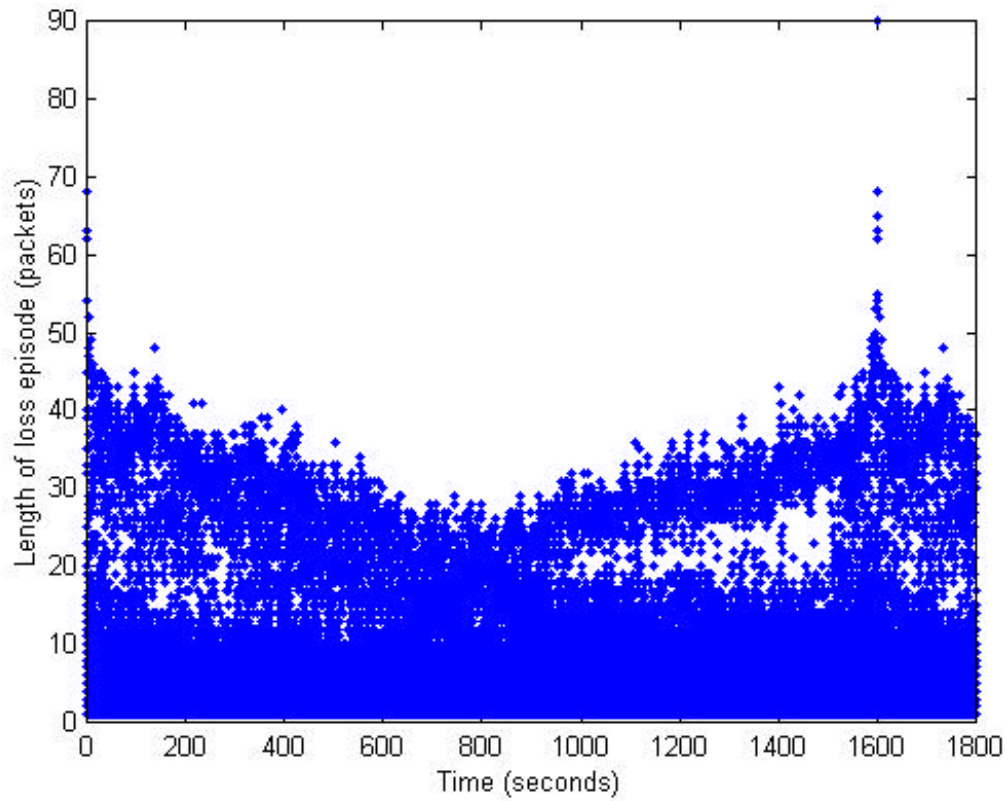


Figure A3. Aggregate packet loss process. Simulation with 100 traffic sources, using one MPEG trace (Jurassic Park 1). Each point in the graph corresponds to a particular loss episode. The occurrence time for each loss episode is the occurrence time of the first lost packet in the loss episode.

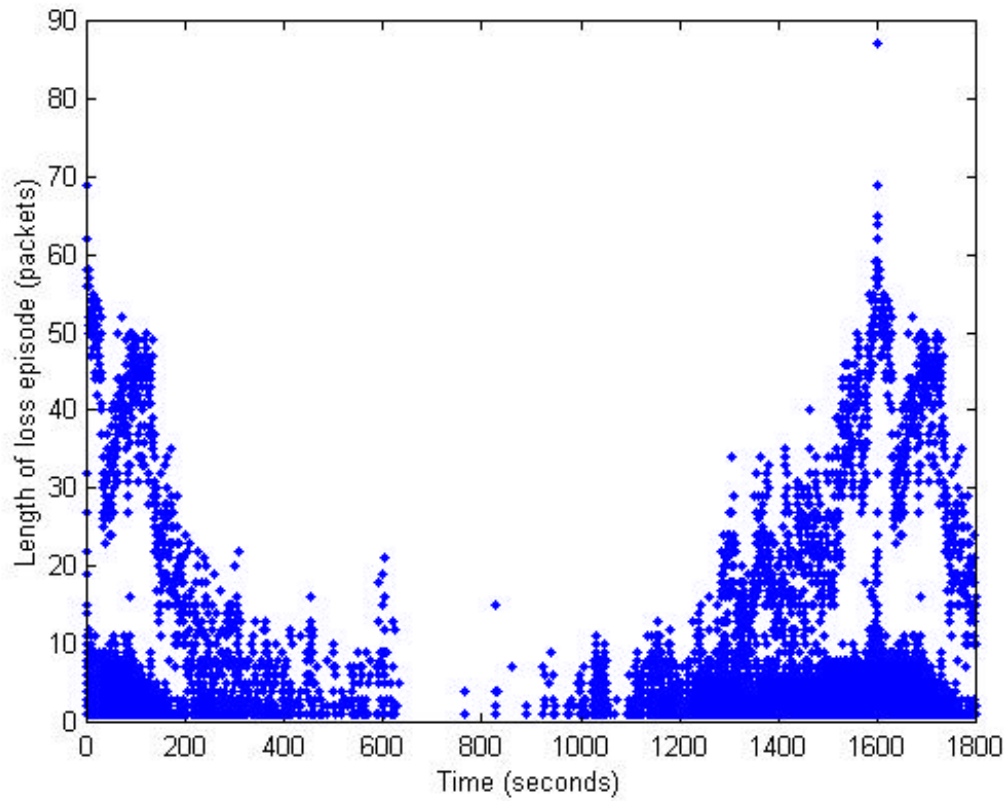


Figure A4. Aggregate packet loss process. Simulation with 100 traffic sources, using one MPEG trace (Star Wars). Each point in the graph corresponds to a particular loss episode. The occurrence time for each loss episode is the occurrence time of the first lost packet in the loss episode.

Appendix B.

Effect of traffic load on aggregate packet loss, simulation with M. Garrett's Star Wars MPEG-1 trace

As mentioned in Section 6.2.2, the contribution of loss episodes of length two is almost identical for all traffic loads in our simulation. This phenomenon has also been observed in [26], where the MPEG trace is the MPEG-1 Star Wars trace originally created by Mark Garrett [17], [43], [44]. We repeat the simulations as in [26]. Table B1 is a summary of the statistics of the trace. Figure B1 and B2 show the contribution of loss episodes. Figure B3 shows the contribution of loss episodes for episodes of length one to three packets. As shown in Figure B3, the contribution of loss episode length two is the same for all traffic loads.

Table B1. Summary of statistics of Garrett's MPEG-1 Star Wars trace [26].

Encoder input (pel)	480×504
Resolution (bits/pel)	8
GoP pattern	IBBPBBPBBPBB
GoP size	12
Frame rate (frames/second)	24
Number of video frames	174,136
Peak bit rate (Mbps)	4.446
Mean bit rate (Mbps)	0.3744

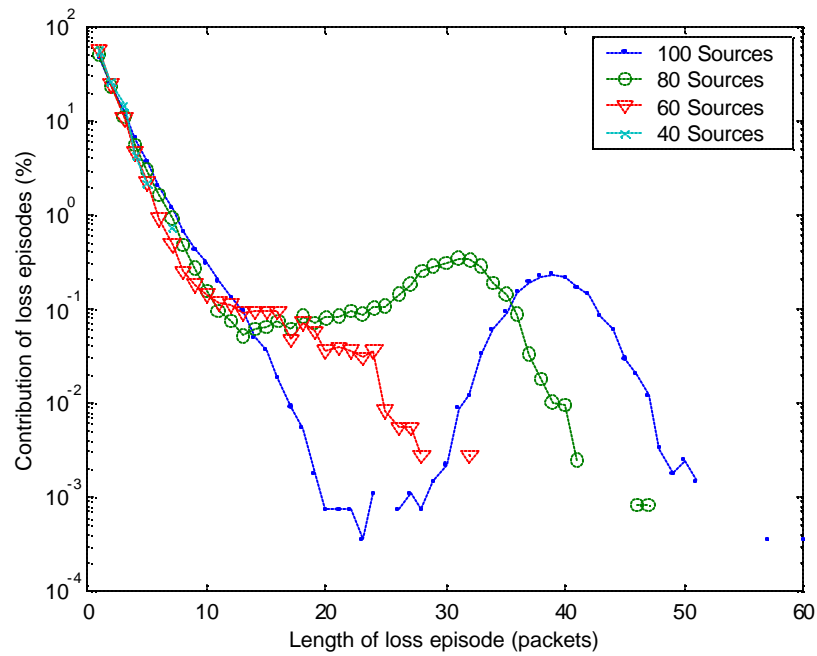
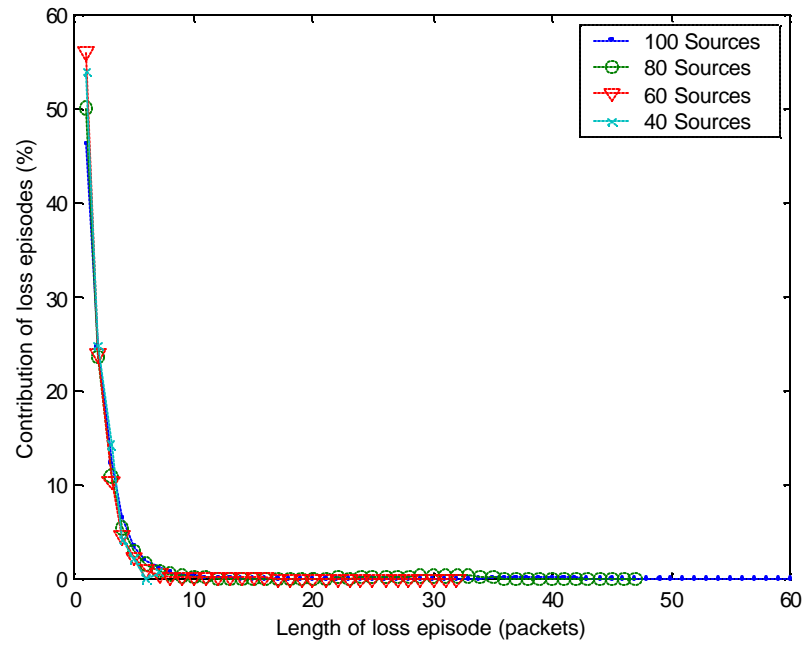


Figure B1. Contribution of loss episodes of various lengths to the overall number of loss episodes, linear (top) and log (bottom) scale. Simulation with 40 to 100 traffic sources, using Garrett's Star Wars MPEG traces, buffer size = 46 packets.

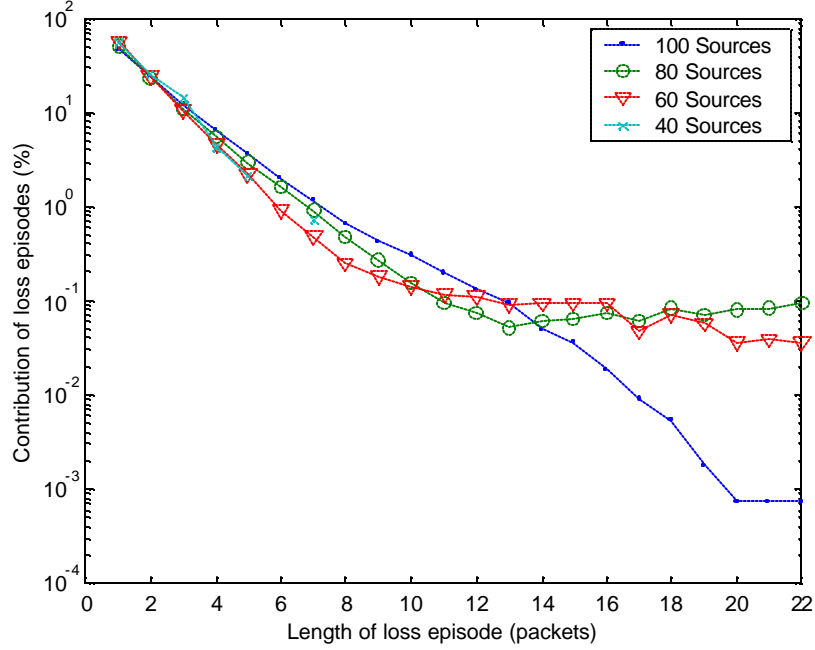


Figure B2. Contribution of loss episodes of various lengths to the overall number of loss episodes, episode length up to 22 packets. Simulation with 40 to 100 traffic sources, using Garrett's Star Wars MPEG traces, buffer size = 46 packets.

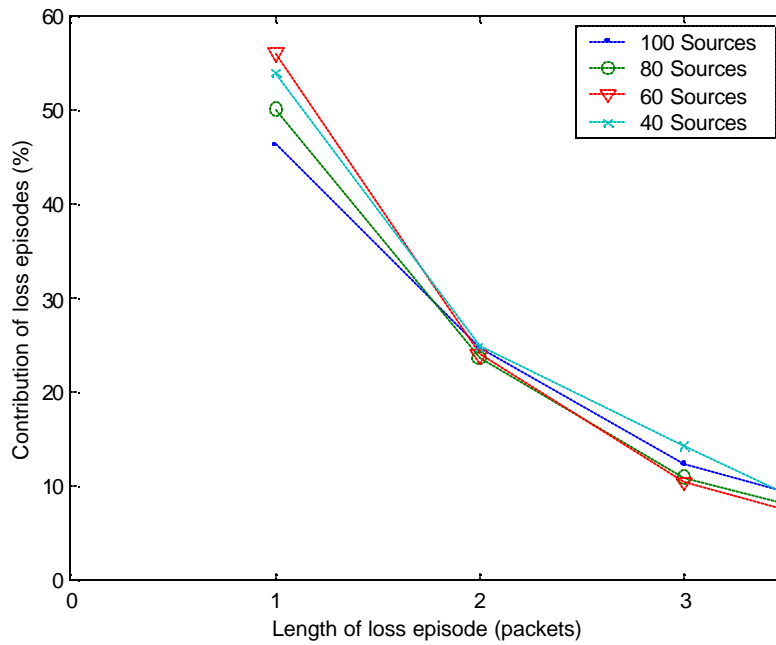


Figure B3. Contribution of loss episodes of various lengths to the overall number of loss episodes, episode length from 1 and to 3. Simulation with 40 to 100 traffic sources, using Garrett's Star Wars MPEG traces, buffer size = 46 packets.

Appendix C.

Contribution of loss episodes for each individual flow

In Section 6.2.3, we showed the contribution of loss episodes of various lengths to the overall number of loss episodes, averaged over all individual flows (in Figure 6.8). In this section, we plot the contribution of loss episodes for each individual flow in Figures C1 to C10 for the simulation with buffer size = 46 packets. Figure 6.8 can be derived from Figures C1 to C10.

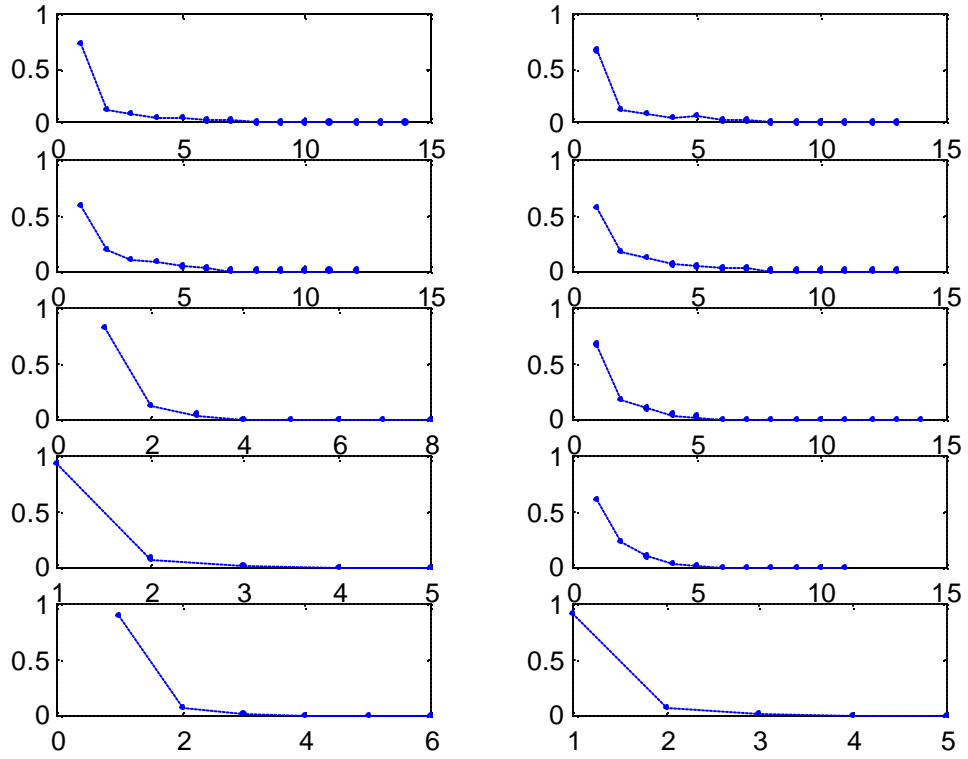


Figure C1. The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 1 to 10. The x-axis is the contribution of loss episodes and the y-axis is the length of loss episode in packets. Simulation with 100 traffic sources, using 10 different MPEG traces.

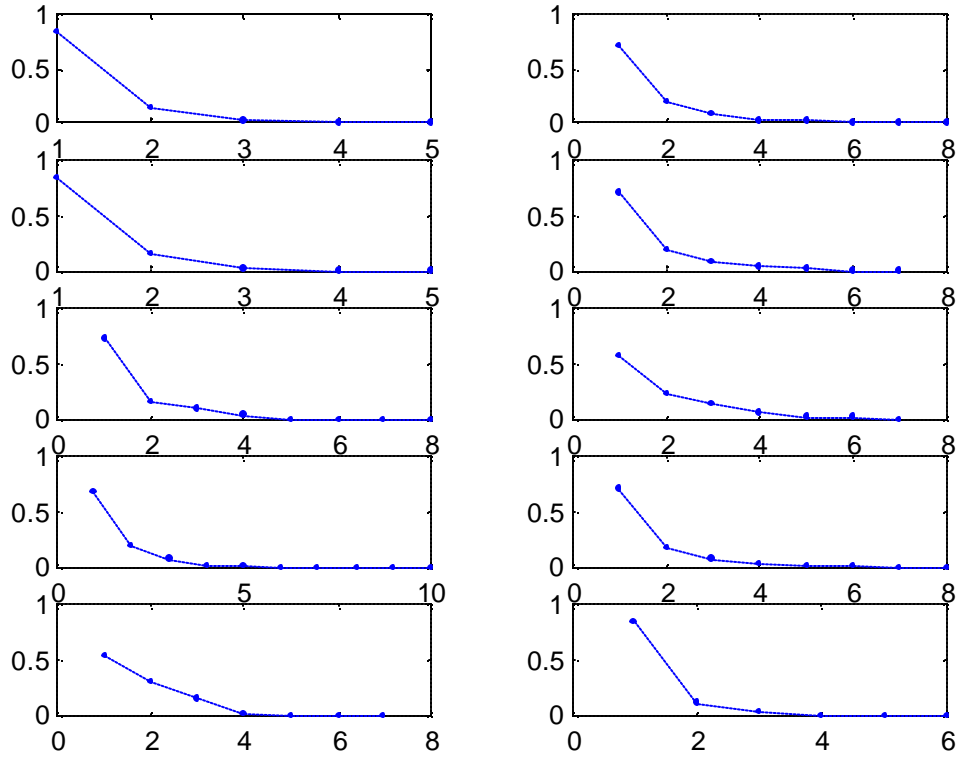


Figure C2. The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 11 to 20. The x-axis is the contribution of loss episodes and the y-axis is the length of loss episode in packets. Simulation with 100 traffic sources, using 10 different MPEG traces.

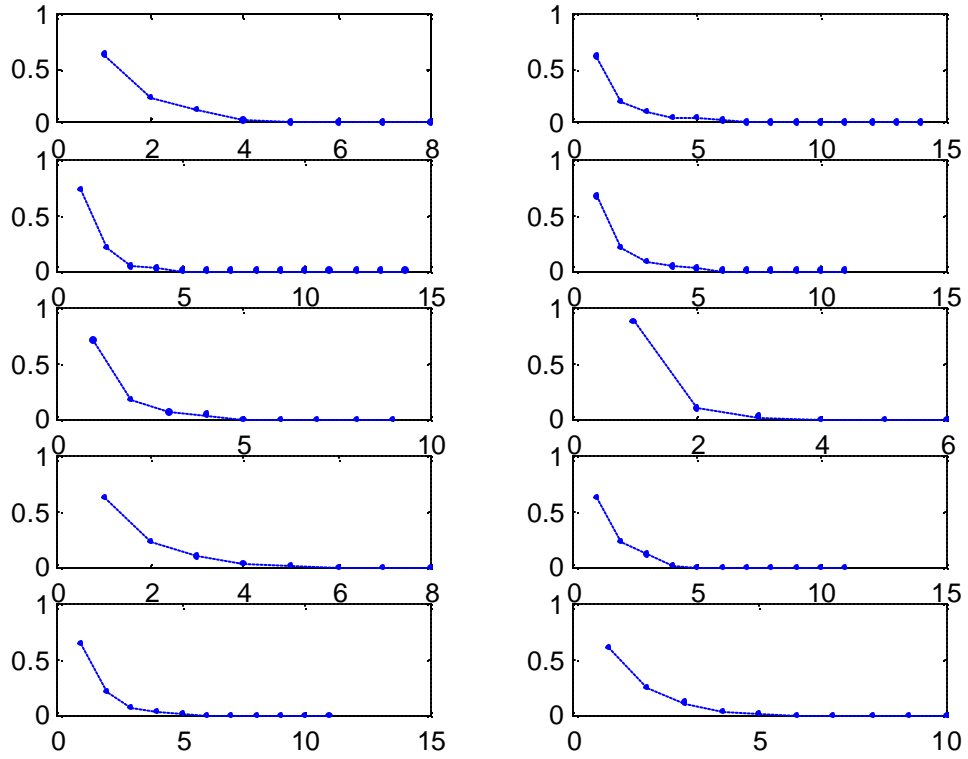


Figure C3. The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 21 to 30. The x-axis is the contribution of loss episodes and the y-axis is the length of loss episode in packets. Simulation with 100 traffic sources, using 10 different MPEG traces.

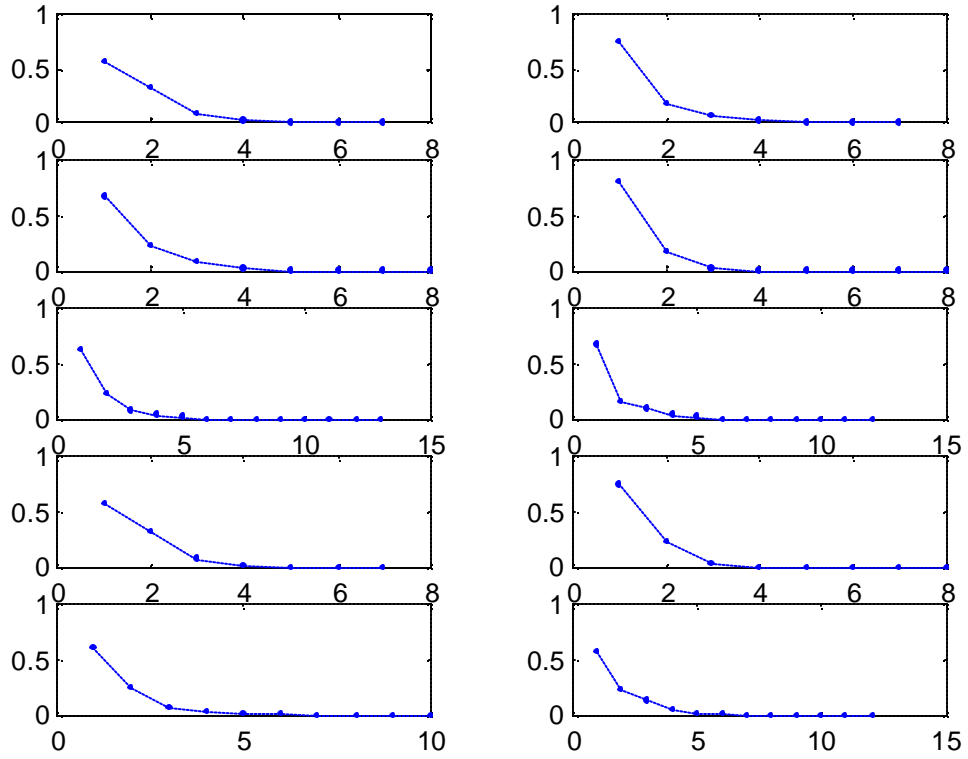


Figure C4. The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 31 to 40. The x-axis is the contribution of loss episodes and the y-axis is the length of loss episode in packets. Simulation with 100 traffic sources, using 10 different MPEG traces.

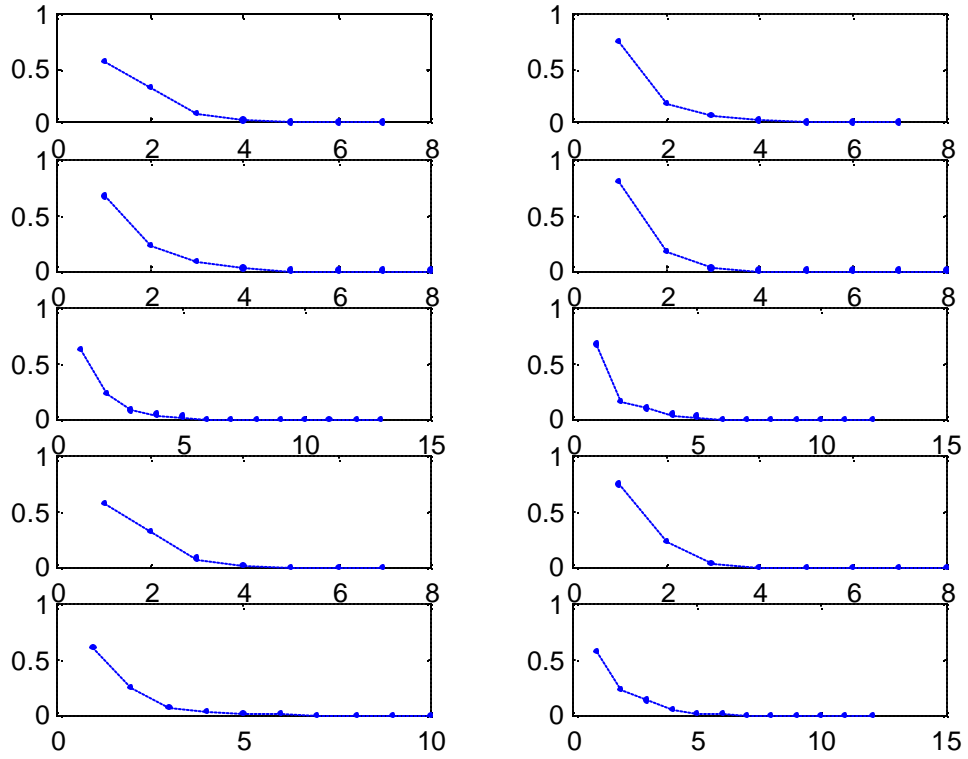


Figure C5. The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 41 to 50. The x-axis is the contribution of loss episodes and the y-axis is the length of loss episode in packets. Simulation with 100 traffic sources, using 10 different MPEG traces.

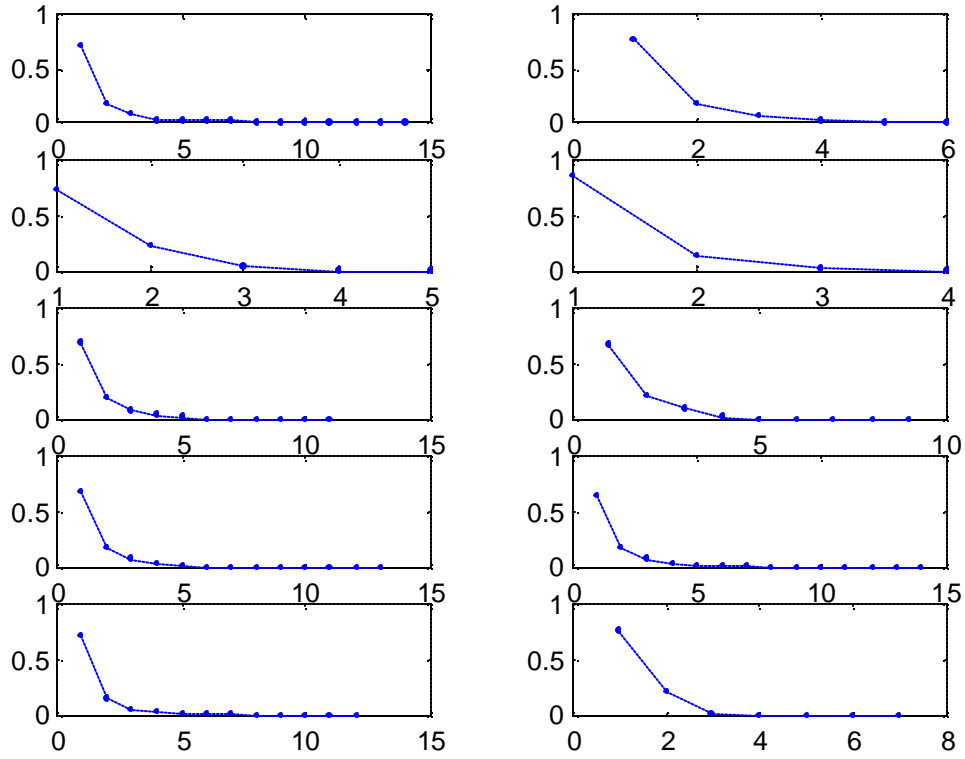


Figure C6. The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 51 to 60. The x-axis is the contribution of loss episodes and the y-axis is the length of loss episode in packets. Simulation with 100 traffic sources, using 10 different MPEG traces.

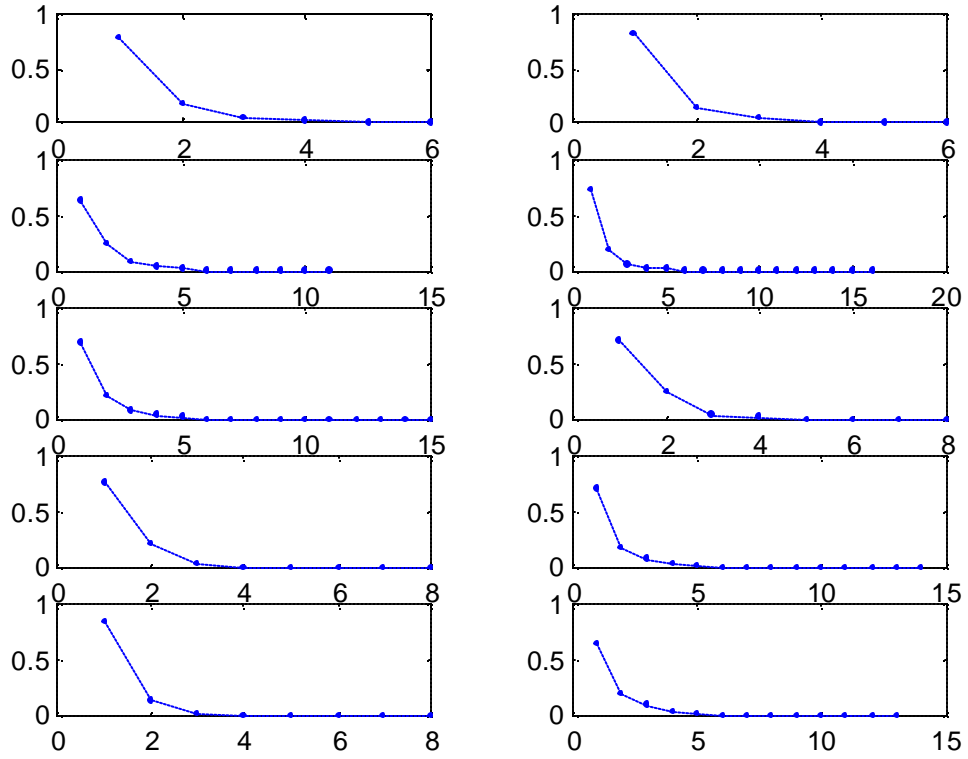


Figure C7. The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 61 to 70. The x-axis is the contribution of loss episodes and the y-axis is the length of loss episode in packets. Simulation with 100 traffic sources, using 10 different MPEG traces.

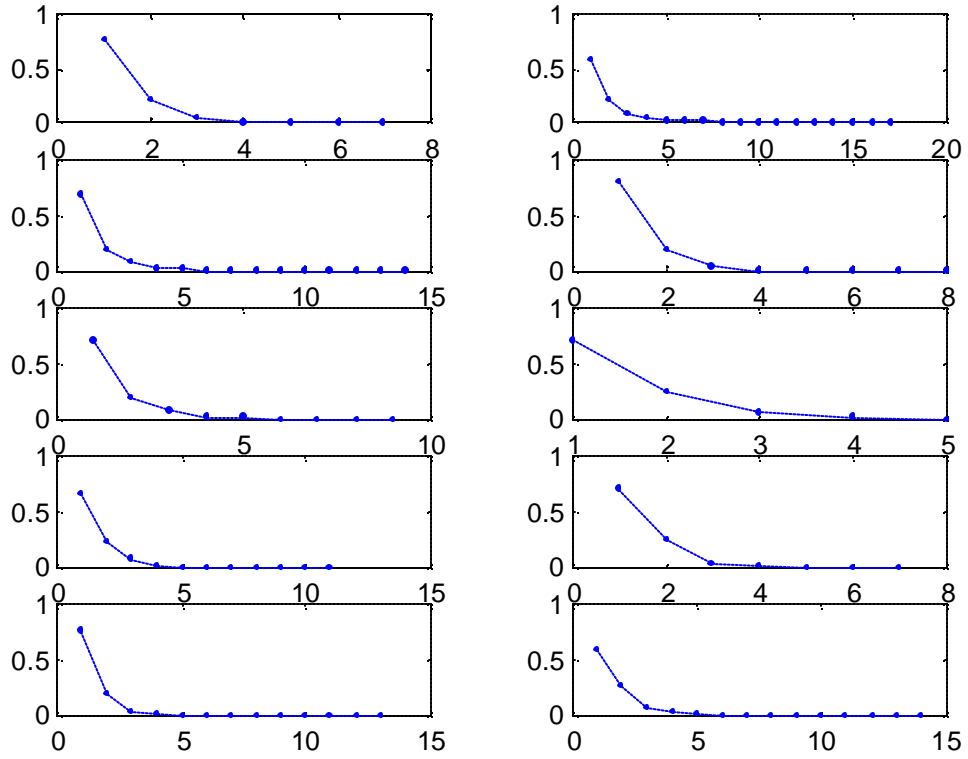


Figure C8. The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 71 to 80. The x-axis is the contribution of loss episodes and the y-axis is the length of loss episode in packets. Simulation with 100 traffic sources, using 10 different MPEG traces.

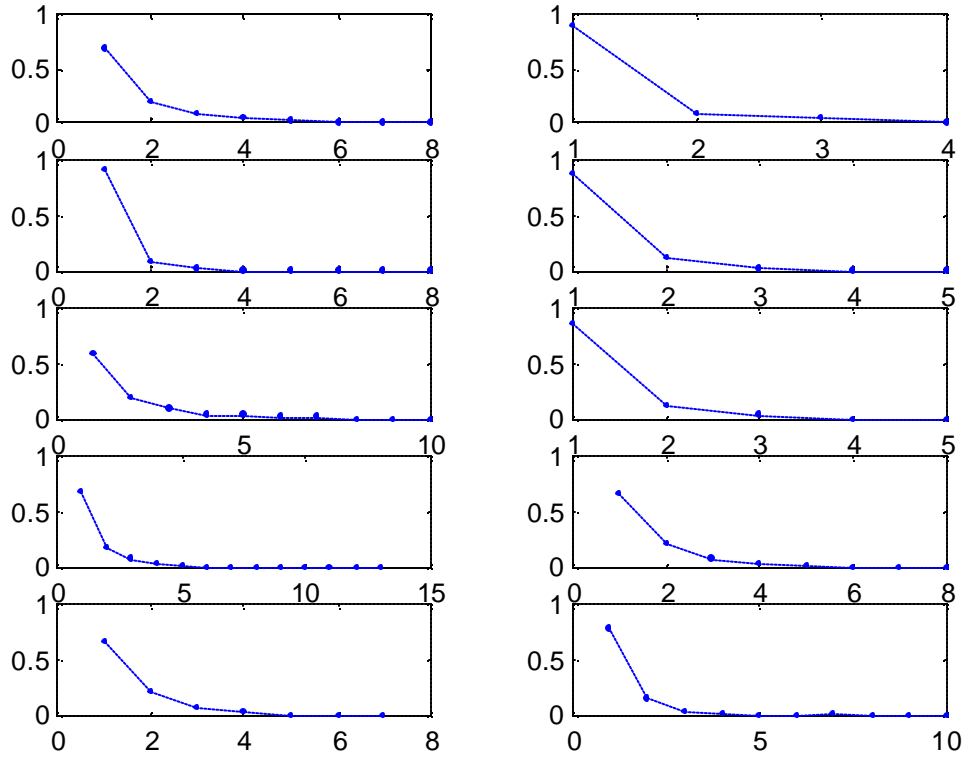


Figure C9. The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 81 to 90. The x-axis is the contribution of loss episodes and the y-axis is the length of loss episode in packets. Simulation with 100 traffic sources, using 10 different MPEG traces.

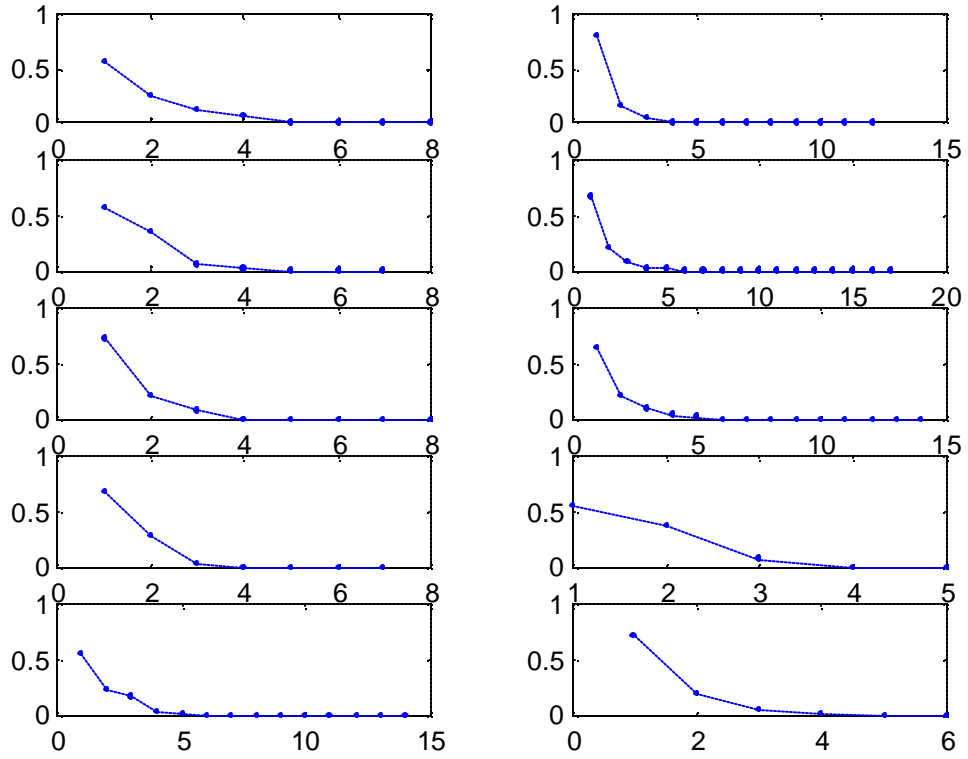


Figure C10. The contribution of loss episodes of various lengths to overall number of loss episodes for traffic source number 91 to 100. The x-axis is the contribution of loss episodes and the y-axis is the length of loss episode in packets. Simulation with 100 traffic sources, using 10 different MPEG traces.

Appendix D.

Additional simulation results

In this Appendix, we include additional simulation results previously mentioned in this thesis.

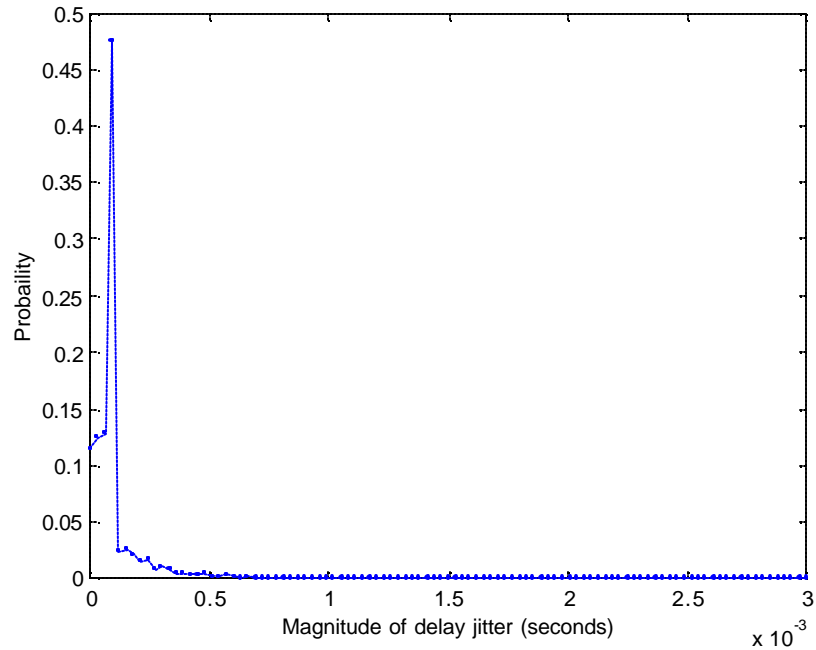
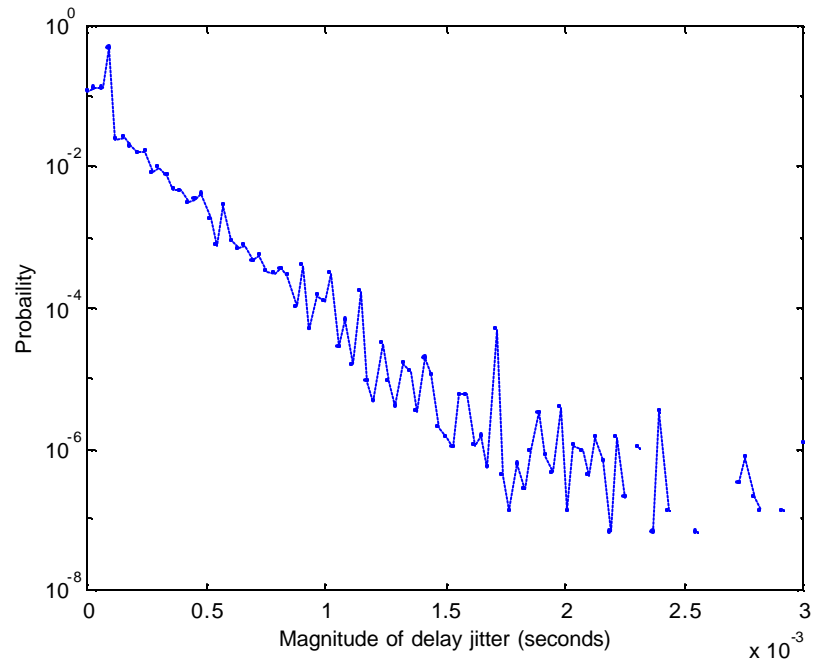


Figure D1. Distribution of the magnitude of packet delay jitter, linear scale (top) and log scale (bottom). FIFO/DropTail simulation with 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

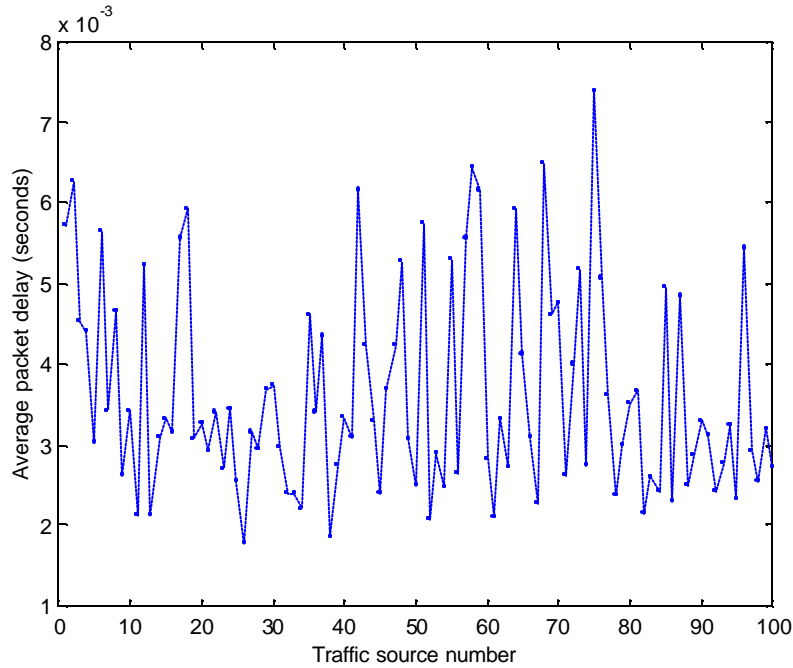


Figure D2. Average packet delay for packets from the same flow. FIFO/DropTail simulation with 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

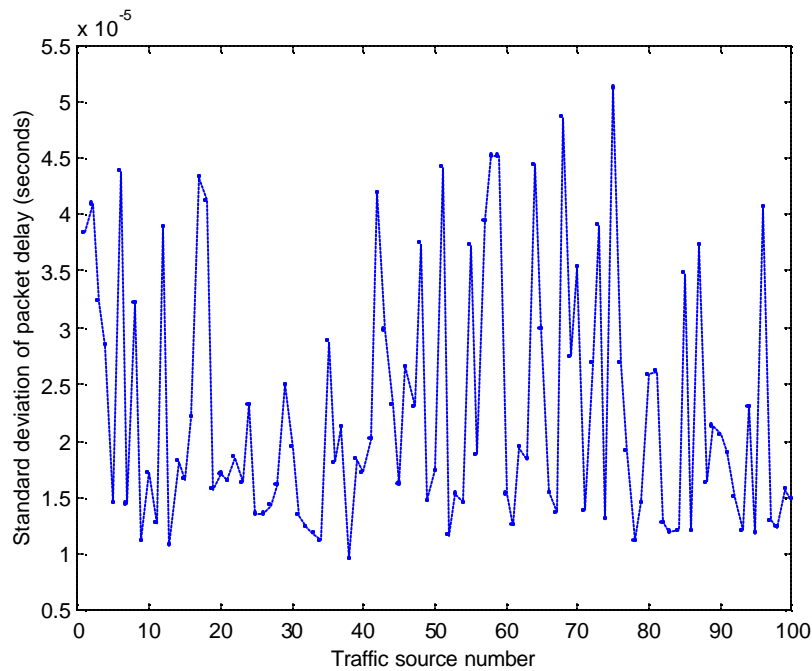


Figure D3. Standard deviation of packet delay for packets from the same flow. FIFO/DropTail simulation with 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

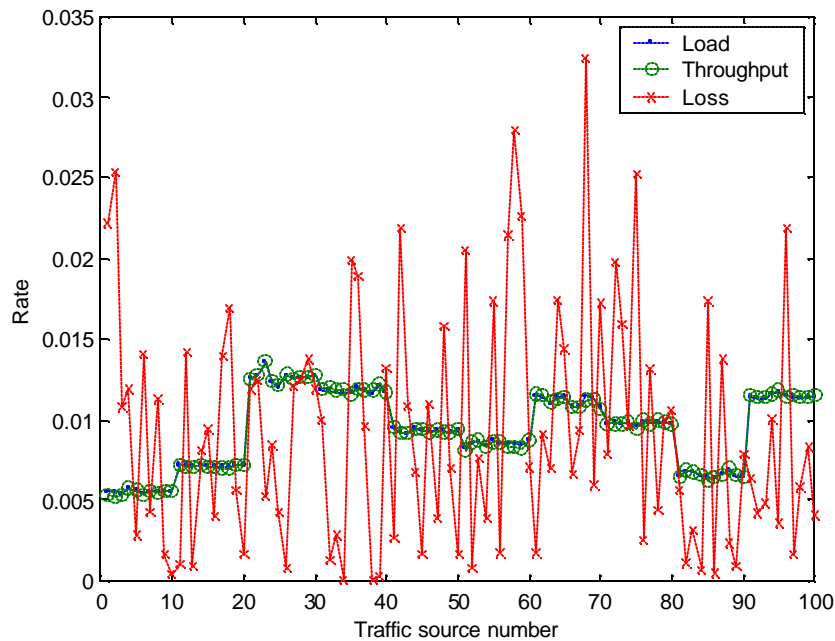


Figure D4. Per-flow load, throughput, and loss, calculated with respect to the total traffic load, throughput, and packet loss. FIFO/DropTail simulation with 100 traffic sources, using 10 different MPEG traces, buffer size = 46 packets. The total traffic load, throughput and packet loss are shown in Table 6.2.

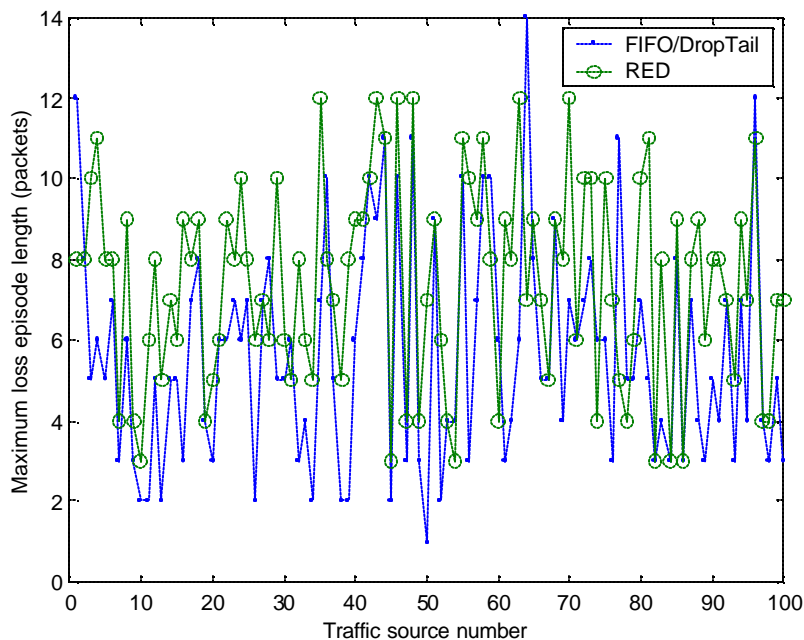


Figure D5. The length of the longest loss episode for each flow. Simulation with FIFO/DropTail and RED, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

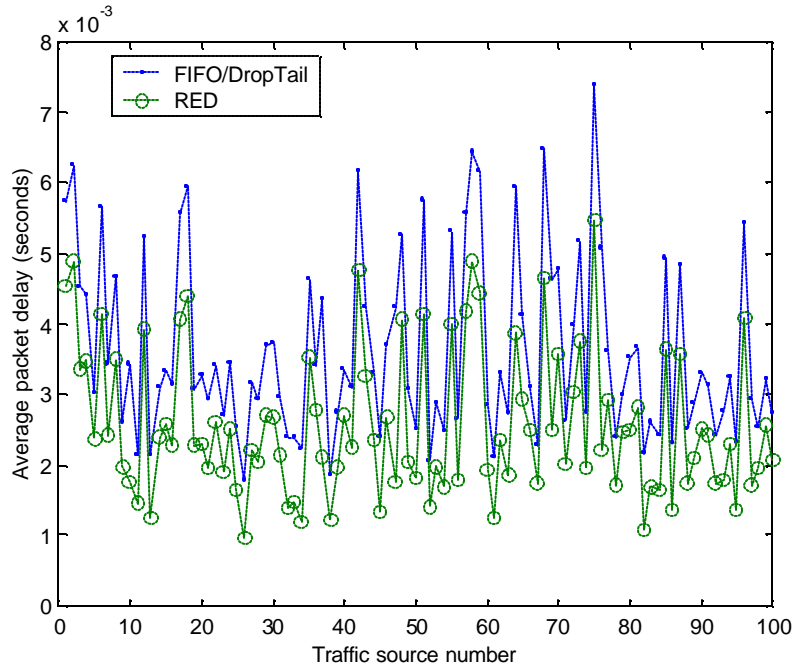


Figure D5. Average packet delay for packets from the same flow. Simulation with FIFO/DropTail and RED, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

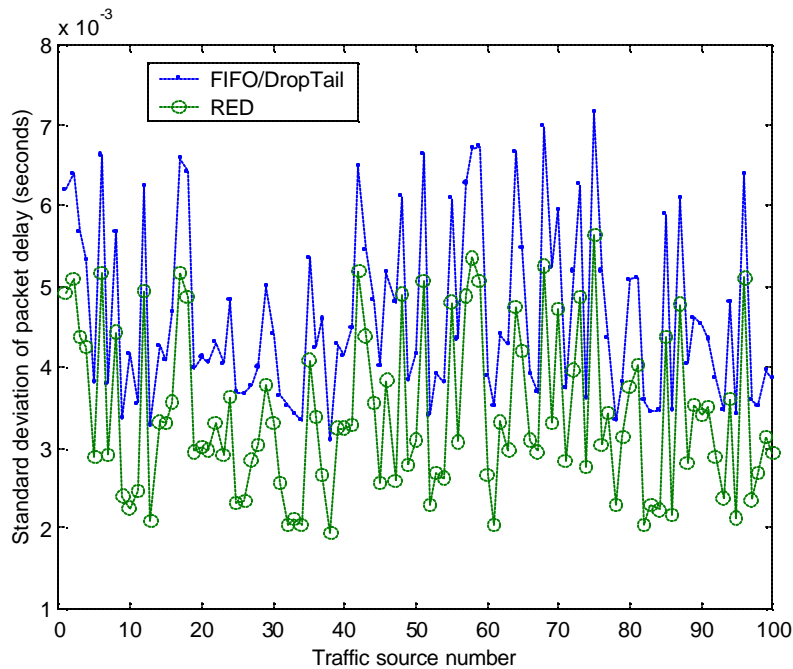


Figure D6. Standard deviation of packet delay for packets from the same flow. Simulation with FIFO/DropTail and RED, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

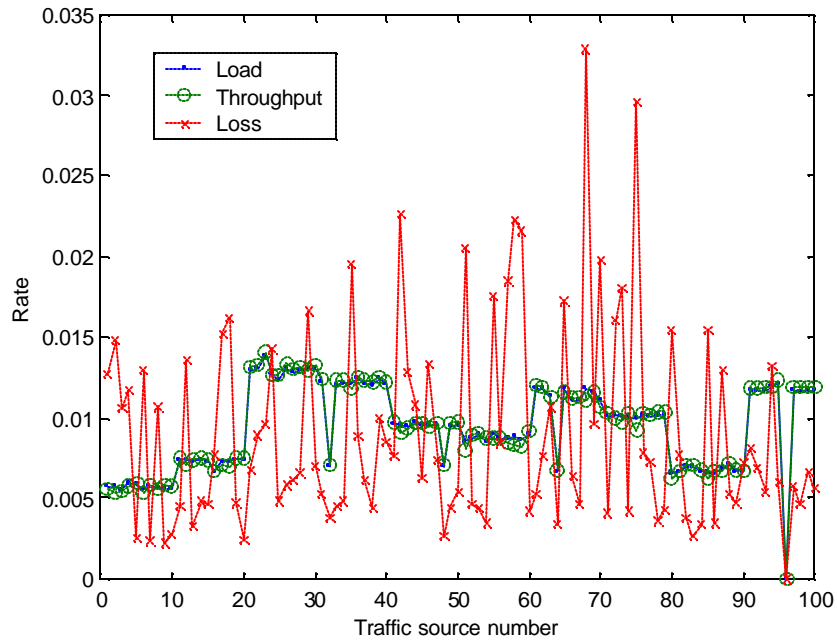


Figure D7. Per-flow load, throughput, and loss, calculated with respect to the total traffic load, throughput, and packet loss. Simulation with RED, 100 traffic sources, using 10 different MPEG traces, buffer size = 200 packets.

References

- [1] A guide to MPEG fundamentals and protocol analysis:
http://www.tektronix.com/Measurement/App_Notes/mpegfund [June 30, 2002].
- [2] G. Almes, S. Kalidindi, and M. Zekauskas, "A one-way delay metric for IPPM," IETF RFC 2679, Sept. 1999.
- [3] S. Battista, F. Casalino, and C. Lande, "MPEG-4: A multimedia standard for the third millennium, Part 1":
http://www.computer.org/multimedia/articles/MPEG4_2.htm [June 30, 2002].
- [4] Berkeley MPEG tools: <http://bmrc.berkeley.edu/research/mpeg> [June 30, 2002].
- [5] G. Bombelli, "Quantitative Comparison between DropTail and Random Early Detection Queue Management in a Simple Topology," Directed Studies Report, Engineering Science, Simon Fraser University, 2000
- [6] J. M. Boyce and R. D. Gaglianella, "Packet loss effects on MPEG video sent over the public Internet," *Proc. ACM Multimedia '98*, Bristol, UK, Sept. 1998, pp. 181-190.
- [7] L. Breslau, "Example traffic trace for ns":
<http://www.research.att.com/~breslau/vint/trace.html> [June 30, 2002].
- [8] K.C. Claffy, G. Miller, and K. Thompson, "The nature of the beast: recent traffic measurements from an Internet backbone," *Proc. INET'98*, Geneva, Switzerland, July 1998.
- [9] M. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," *IEEE/ACM Trans. Networking*, vol. no. 6 pp. 835-846, Dec. 1997.
- [10] A. Demers, S. Keshav, and S. Shanker, "Analysis and simulation of a fair queuing algorithm," *Proc. ACM SIGCOMM '89*, Sept. 1989, pp. 1-12.
- [11] Designing internetworks for multimedia:
<http://www.cisco.com/univercd/cc/td/doc/cisintwk/idg4/nd2013.htm>
[June 30, 2002].
- [12] A. Feldmann, A. C. Gilbert, P. Huang, and W. Willinger,. "Dynamics of IP traffic: A study of the role of variability and the impact of control," *Proc. of ACM SIGCOMM '99*, Cambridge, MA, USA, Aug. 1999, pp. 301-313.

- [13] A. Feldmann, A. C. Gilbert, and W. Willinger. "Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic," *Proc. ACM SIGCOMM '98*, Vancouver, BC, CANADA, Aug. 1998, pp. 42-55.
- [14] M. Fester, "Performance issues for high-end video over ATM": <http://www.cisco.com/warp/public/cc/sol/mkt/ent/atm/vidatwp.htm> [June 30, 2002].
- [15] F. Fitzek and M. Reisslein, "MPEG-4 and H.263 video traces for network performance evaluation," Technical Report TKN-00-06, Technical University of Berlin, Telecommunication Networks Group, Berlin, Germany, Oct. 2000.
- [16] S. Floyd and V. Jacobson, "Random early drop gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397-413, Aug. 1993.
- [17] M. W. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," *Proc. Sigcomm '94*, Sept. 1994, pp. 269-280.
- [18] D. Hoffman, G. Fernando, V. Goyal, and M. Civanlar, "RTP payload format for MPEG1/MPEG2 video," IETF RFC 2250, Jan. 1998.
- [19] M. Jiang, "Analysis of Wireless Data Network Traffic," M.A.Sc. Thesis, Engineering Science, Simon Fraser University, 2000.
- [20] W. Jiang and H. Schulzrinne, "QoS measurement of Internet real-time multimedia services," Technical Report CUCS-015-99, Department of Computer Science, Columbia University, December 1999.
- [21] S. Keshav, *An Engineering Approach to Computer Networking*, Reading MA: Addison Wesley, 1997.
- [22] S. M. Klivansky, A. Mukherjee, and C. Song, "On long-range dependence in NSFNET traffic," *Proc. of 7th IEEE LAN/MAN Workshop*, Marathon, FL, USA, Mar. 1995.
- [23] M. Krunz, R. Sass, and H. Hughes, "Statistical characteristics and multiplexing of MPEG streams," *Proc. of IEEE INFOCOM*, 1995, pp. 455-462.
- [24] S. Lavington, N. Dewhurst, and M. Ghanbari, "The performance of layered video over an IP network," *Proc. of 10th International Packet Video Workshop*, Cagliari Italy, May 2000.
- [25] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1-15, Feb. 1994.

- [26] V. Markovski, "Simulation and Analysis of Loss in IP Networks," M.A.Sc. Thesis, Engineering Science, Simon Fraser University, 2000.
- [27] V. Markovski and Lj. Trajkovic, "Analysis of loss periods for video transfer over UDP," *SPECTS '2K*, Vancouver, BC, Canada, July 2000, pp. 278-285.
- [28] P. McKenney, "Stochastic fairness queuing," *Internetworking: Research and Experience*, vol.2, pp. 113-131, Jan. 1991.
- [29] MPEG-4 and H.263 video traces for network performance evaluation:
<http://www-tnk.ee.tu-berlin.de/research/trace/trace.html> [June 30, 2002].
- [30] J. Nagle, "On packet switches with infinite storage," *IEEE Trans. on Comm.*, vol. 35, no. 4, pp. 435-438, April 1987.
- [31] ns-2 network simulator: <http://www.isi.edu/nsnam/ns> [June 30, 2002].
- [32] Quality of service networking:
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm [June 30, 2002].
- [33] Quality of service:
http://www.cisco.com/univercd/cc/td/doc/product/voice/ip_tele/network/dgqos.htm
[June 30, 2002].
- [34] K. Park and W. Willinger, "Self-similar network traffic: An overview," in *Self-Similar Network Traffic and Performance Evaluation*, Ed. K. Park and W. Willinger, New York: Wiley Interscience, 1999, pp. 1-38.
- [35] Reference guide: The benefits of using MPEG at low bit rates:
http://www.cisco.com/warp/public/cc/pd/mxsv/iptv3400/tech/mpeg_rg.htm
[June 30, 2002].
- [36] O. Rose. "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," Technical Report 101, University of Wuerzburg, Institute of Computer Science, Am Hubland, 97074 Wuerzburg, Germany, Feb. 1995.
- [37] N. Rowe, "Multimedia systems":
<http://www.cs.nps.navy.mil/people/faculty/rowe/multimediaover.htm>
[June 30, 2002].
- [38] Z. Sahinoglu and S. Tekinay, "On Multimedia Networks: Self Similar traffic and Network Performance," *IEEE Communications Magazine*, pp. 48-52, Jan. 1999.
- [39] H. Sawashima, Y. Hori, H. Sunahara, and Y. Oie, "Characteristics of UDP packet loss: Effect of TCP traffic," *Proc. INET'97*, Kuala Lumpur, Malaysia, June 1997.

- [40] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF RFC 1889, Jan. 1996.
- [41] J. Sedayao, "Comment on delay metrics and value when packets dropped": <http://www.advanced.org/IPPM/archive.2/0459.html> [June 30, 2002].
- [42] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round robin," *Proc. of ACM SIGCOMM '95*, Aug. 1995, pp. 231-242.
- [43] Star Wars trace in ASCII format: <ftp://ftp.telcordia.com/pub/vbr.video.trace> [June 30, 2002].
- [44] Star Wars trace in ns format: <http://www.research.att.com/breslau/vint/trace.html> [June 30, 2002].
- [45] K. Thompson, G. Miller, and R. Wilder, "Wide area internet traffic patterns and characteristics," *IEEE/ACM Trans. Networking*, pp. 10-23, Nov. 1997.
- [46] J. Maheswary, Traffic characterization of heterogeneous applications: http://csgrad.cs.vt.edu/~jiten/Final_QoS_files/frame.htm [June 30, 2002].
- [47] UDP network system documentation: <http://www.hesinki.fi/~pkamppur/udplibray/documentation.html> [June 30, 2002].
- [48] University of Wuerzburg archive of video traces: <http://nero.informatik.uni-wuerzburg.de/MPEG> [June 30, 2002].
- [49] VINT (Virtual InterNetwork Testbed): <http://www.isi.edu/nsnam/vint> [June 30, 2002].
- [50] J. Warland and P. Varaiya, *High Performance Communication Networks, 2nd. ed.*, San Fransico, CA: Morgan Kaufmann, 2000.
- [51] W. Willinger, D. V. Wilson, W. E. Leland, and M. Taqqu, "On traffic measurements that defy traffic models (and vice versa): Self-similar traffic modeling for high-speed networks," *ConneXion*, vol. 8, no. 11, pp. 14-24, 1994
- [52] F. Xue, V. Markovski, and Lj. Trajkovic, "Packet loss in video transfers over IP networks," *Proc. IEEE Int. Symp. Circuits and Systems*, Sydney, Australia, May 2001, vol. II, pp. 345-348.