Methods of Congestion Control for Adaptive Continuous Media

A dissertation submitted for partial fulfilment of
Doctor of Philosophy

by
Shalini Tater (B.Eng., AMIEE)

Oxford Brookes University
School of Computing and Mathematical Sciences
Gipsy Lane Campus
Oxford, OX3 0BP
UK

2002

Abstract

Since the first exchange of data between machines in different locations in early 1960s, computer networks have grown exponentially with millions of people now using the Internet. With this, there has also been a rapid increase in different kinds of services offered over the World Wide Web from simple e-mails to streaming video. It is generally accepted that the commonly used protocol suite TCP/IP alone is not adequate for a number of modern applications with high bandwidth and minimal delay requirements. Many technologies are emerging such as IPv6, Diffserv, Intserv etc, which aim to replace the one-size-fits-all approach of the current IPv4. There is a consensus that the networks will have to be capable of multi-service and will have to isolate different classes of traffic through bandwidth partitioning such that, for example, low priority best-effort traffic does not cause delay for high priority video traffic. However, this research identifies that even within a class there may be delays or losses due to congestion and the problem will require different solutions in different classes.

The focus of this research is on the requirements of the adaptive continuous media class. These are traffic flows that require a good Quality of Service but are also able to adapt to the network conditions by accepting some degradation in quality. It is potentially the most flexible traffic class and therefore, one of the most useful types for an increasing number of applications.

This thesis discusses the QoS requirements of adaptive continuous media and identifies an ideal feedback based control system that would be suitable for this class. A number of current methods of congestion control have been investigated and two methods that have been shown to be successful with data traffic have been evaluated to ascertain if

they could be adapted for adaptive continuous media. A novel method of control based on percentile monitoring of the queue occupancy is then proposed and developed. Simulation results demonstrate that the percentile monitoring based method is more appropriate to this type of flow. The problem of congestion control at aggregating nodes of the network hierarchy, where thousands of adaptive flows may be aggregated to a single flow, is then considered. A unique method of pricing mean and variance is developed such that each individual flow is charged fairly for its contribution to the congestion.

Acknowledgments

I started this research at Oxford Brookes University and finished a significant part before taking a year's placement at Nortel Networks Harlow Laboratories and I am currently working at Tandberg Television. I am grateful to Oxford Brookes University for giving me the opportunity of research with a generous scholarship and part-time employment, to Nortel Networks for providing me with a chance to gain industrial experience as well as some additional funding, and to Tandberg Television for employing me and supporting me in the last few months to complete this thesis.

Words of encouragement, advice, guidance, and support from all my friends and colleagues have all been extremely instrumental in accomplishment of this work and I am extremely grateful to them all.

Particularly, I would like to thank all members, past and present, of School of Computing and Mathematical Sciences of Oxford Brookes University for their help and support at numerous occasions. Special thanks are due to the Head of the School, John Nealon, and the School Administrator Sue Flint particularly for their support when I started my year at Nortel Networks. Thanks are also due to Nigel Crook for his helpful pointers and to Phyllis Callinan for getting me up to speed with OPNET™. At the university Research Centre, I would like to thank Abigail Fisher and Jill Organ for their efficient administrative support.

I would also like to thank everyone at Nortel Networks, including those who have since left the company, for making my time there extremely enjoyable and productive. In particular, I would like to thank John Winterbotham and Ashraf Khan for managerial help, Marino Calierno and Raviraj Rajkulasingam for their invaluable advice, and Sabesan

Subramaniyam, Martin Biddiscombe and Paula Fonseca for their suggestions at various times.

I am greatly indebted to my supervisor Dr Frank Ball for his excellent guidance and support throughout and to Dr Paul Kirkby for supervising my work at Nortel. I have been extremely fortunate that both my supervisors have contributed a lot of their time and I have benefited a lot from their insight.

I am thankful to Geoff Tagg, then the Head of Distributed Systems Research Group at Oxford Brookes, for responding to my queries regarding a PhD position and to Neil Hollingum for suggesting the university and putting me in touch with Geoff Tagg. It turned out to be just what I had wanted. I am also grateful to Rathe Hollingum for putting up with me and generally looking after me particularly in the last few months, which have been very busy.

Finally, I would like to thank my parents Jashkaran and Vijaya Tater and my brother Shalaj. My father was the first person to ever suggest doing research to me, when I was about 15 and the suggestion stayed with me. He also helped me great deal by proofreading the entire thesis. My parents have been extremely supportive of me and have consistently encouraged me to pursue my career even though it meant that I would be staying thousands of miles away from home. And though he may not realise it, Shalaj has been fantastic in his ability to cheer me up and keep me motivated.

Contents

List of Figures

List of Tables

1

Introduction

In recent years, we have experienced a revolution through the Internet. People around the world are sending e-mails, photographs, electronic greetings to their loved ones at the touch of a button whereas previously it would have taken many days, even weeks for their messages to get through the conventional postal service. The services available over the Internet are numerous and increasing. One can browse the web, shop online, make deals in the stock market, attend a virtual meeting with colleagues scattered around the globe, listen to music, watch movies, and so the list goes on. Most children who grow up using the Internet cannot believe that there ever was a world where it was not usual to "surf the net." Indeed, it is difficult for most of us to realise that this "revolutionary technology" had actually started in 1950s as the US Department of Defense set up Advanced Research Projects Agency (ARPA) for military purposes. As early as 1969, we had a network. There were only 4 hosts but it was the beginning of the colossal network we have today with over 100 million hosts.

A short poem by Leonard Kleinrock then celebrating 20 years of ARPAnet takes us back in time for some to reminisce the yesteryears and for others to marvel at how it all began [RFC 1121].

THE BIG BANG!
(or the birth of the ARPANET)
by
Leonard Kleinrock

It was back in '67 that the clan agreed to meet.
The gangsters and the planners were a breed damned hard to beat.
The goal we set was honest and the need was clear to all:
Connect those big old mainframes and the minis, lest they fall.

The spec was set quite rigid: it must work without a hitch.
It should stand a single failure with an unattended switch.
Files at hefty throughput 'cross the ARPANET must zip.
Send the interactive traffic on a quarter second trip.

The spec went out to bidders and t'was BBN that won.
They worked on soft and hardware and they all got paid for fun.
We decided that the first node would be we who are your hosts
And so today you're gathered here while UCLA boasts.

I suspect you might be asking "What means FIRST node on the net?"
Well frankly, it meant trouble, 'specially since no specs were set.
For you see the interface between the nascent IMP and HOST
Was a confidential secret from us folks on the West coast.

BBN had promised that the IMP was running late.
We welcomed any slippage in the deadly scheduled date.
But one day after Labor Day, it was plopped down at our gate!
Those dirty rotten scoundrels sent the damned thing out air freight!

As I recall that Tuesday, it makes me want to cry.
Everybody's brother came to blame the other guy!
Folks were there from ARPA, GTE and Honeywell.
UCLA and ATT and all were scared as hell.

We cautiously connected and the bits began to flow.
The pieces really functioned - just why I still don't know.
Messages were moving pretty well by Wednesday morn.
All the rest is history - packet switching had been born!

Since the time when the first ever packets were sent from University College of Los Angeles to Stanford Research Institute in 1969, we have seen a truly exponential growth in the number of users and it has led to a corresponding increase in applications and services. With the increase in services on offer over the Internet, particularly the multimedia applications, the expectations of users have risen sharply. Users want better services at lower cost and suppliers are constantly striving to develop new technology to meet these demands. Modems have become faster and better, or often replaced by higher bandwidth

alternatives such as ISDN or xDSL lines. Faster computers and faster networks are enabling us to carry out a number of activities over the Internet that were simply not possible some years ago. However, it is not surprising that after a number of improvements a technology reaches its limit beyond which further improvements and additions cease to be effective and a new approach is needed. We can see something of this nature with the Internet as well. The majority of the network uses the TCP/IP protocol suite, which although well suited for data transmission has severe limitations with delivery of multimedia. Data such as e-mail has a different requirement from the network than real time traffic such as interactive audio and video. For example, services such as Internet telephony and teleconferencing are highly sensitive to delay and variation in delay between packets. The widely used IP version 4 is rapidly running out of address space and has no mechanism to differentiate between the two traffic types. It is not feasible to continue using IPv4 if we are to offer a reliable service for continuous media traffic.

It is important to meet the Quality of Service (QoS) requirements of different traffic types and the range of heterogeneous applications makes this a difficult task. A number of new architectures are being developed that will enhance or replace the TCP/IP and other commonly used architectures such as X.25 and Frame Relay in order to provide multi-service. Most likely, the system will use bandwidth partitioning in order to differentiate between different classes of traffic. It is envisaged that there will be a "guaranteed" class for premium flows that require a given data rate, delay etc. Then, there will be a number of "adaptive" classes with different level of tolerance for flows that would prefer to receive a certain QoS from the network but may be tolerant to some level of degradation in one or more QoS parameters. Finally, there will be a "best-effort" class that transfers the data without any guarantees, in the same way as the current TCP/IP.

It then follows that if the bandwidth partitioning is done carefully, the different types of traffic should not interfere with each other. However, there is still a possibility of congestion within a class, which could be detrimental to services with constraints on acceptable delay or loss. Congestion occurs when packets arrive at a queue faster than they can be served. It may have a different meaning in different classes of traffic. For data traffic, congestion usually means that packets are lost due to the buffer overflow. In continuous media application, losses or severe delays can cause disruptions to the service. Congestion therefore, occurs when the length of time that a packet has to wait before being served exceeds a given limit.

In guaranteed classes, further bandwidth partitioning can be used in order to allocate a given bandwidth to each flow, thus eliminating the possibility of congestion but it would be impractical to treat all the flows in this way. Allocating peak rate bandwidth to bursty flows would be inefficient while allocating mean rate may not be sufficient. Such flows may be also be able to adapt to the network conditions. The adaptive flows, which can tolerate some level of degradation, will be more efficiently dealt with if a statistical allocation of bandwidth is made. In these classes, a method of congestion avoidance and control is necessary.

1.1 Research Objectives

This study was started by the author at Oxford Brookes University under the supervision of Dr Frank Ball to develop method of congestion control that are suitable for adaptive continuous media traffic in packet switched networks. A number of real-time applications require that the packet delay, loss and jitter stays within a range of acceptable values. For example, these flows would prefer to have none of the packets delayed for longer than a certain duration but can also accept the service if a small percentile of packets do suffer a

delay that exceeds this limit. These flows are referred to as adaptive flows or more precisely adaptive continuous media traffic.

We begin with the study of quality of service requirements for adaptive continuous media and consider the various methods of congestion control. The suitability of these methods to meet the requirements of adaptive continuous media is investigated. Then a novel method based on percentile monitoring is introduced and the algorithms are developed and tested against the performance of current methods. Simulations of network with each method of congestion control are carried out in OPNET™ Modeler to evaluate the performance.

The research mainly followed the pre-determined route of investigating the literature, evaluating the existing control methods and developing a novel scheme and indeed it was demonstrated that the novel technique based on high-percentile monitoring was more suitable for the adaptive continuous media applications. As is common in all research, there was also an element of evolution. The positive results of the novel monitoring technique led to a collaboration between the University and Nortel Networks. The author continued her research at Nortel Networks with Dr Paul Kirkby, concentrating now on congestion control issues at a higher level of the network hierarchy. This part of the research was based on the Dynamic Resource Control mechanism that had been developed at Nortel. The service requirements at core networks are different due to aggregates of flows being controlled and the need to keep the signalling to minimum. A congestion-price-based mechanism was developed that gives a unique way to optimally control delay sensitive traffic with a wide variety of QoS requirements and different degrees of burstiness. Simulations were used again to evaluate the performance.

1.2 Research Contributions

The major contributions of this research are in the area of congestion control schemes for adaptive continuous media. A brief discussion of the main points follows.

The research was started with a study of existing control schemes. In particular, Hysteresis and RED were investigated in detail to evaluate their suitability for adaptive continuous media applications. The algorithms were simulated in OPNET™ and tested to observe how effective they are in controlling the flows such that a high level of perceived quality of service is maintained. Perceived Quality is discussed in Section 2.4. The simulation results of Hysteresis and RED are presented in Chapter 6.

This followed on to development of a novel feedback control mechanism based on monitoring percentile queue occupancy. It was demonstrated through simulations that this method has clear advantages over the existing control methods, which are based on monitoring the average queue occupancy. The algorithm details and simulations are detailed in Chapter 7.

Then the congestion problem at the core networks was addressed. The study of the traffic behaviour highlighted that the control method must be able to deal with aggregates of flows while minimising signalling. A unique method of pricing mean and variance was developed, by combining the concepts of usage based charging and second moment measurements that have already been researched by others. This unique method uses price as a means of control and gives a solution to the problem of fairly controlling smooth and bursty flows. Fairness, as defined here, is that each flow should be charged for its contribution to congestion. The mean and variance pricing method and the simulation to verify its effectiveness is discussed in Chapter 10. Included in this chapter is a discussion of

fluid flow simulation that was extremely useful to prototype the algorithm. This also made a minor contribution to the simulation techniques.

1.3 Thesis Outline

This thesis comprises eleven chapters including the Introduction. The chapters are presented in a logical order of progress rather than strict chronological order of the work. The remaining chapters are organised as follows:

Chapter 2 discusses Quality of Service (QoS) in the context of Continuous media traffic. The commonly used QoS parameters such as bandwidth, end-to-end delay, jitter, and loss are described. These are the QoS parameters visible to the network and hence measurable. A user would have some expectations from the real-time service depending upon the media used by the application using the service, how the media is transmitted, and the purpose of the application. These requirements vary and it may not be possible to measure them directly. However, an appreciation of these requirements is necessary in order to design the service that will ultimately meet them. We shall discuss requirements of voice and video traffic with a brief introduction to MPEG encoding which is one the most popular encoding techniques. The chapter also looks at a number of different types of applications, which require real-time services but have different constraints. Finally, the chapter looks at the concept of perceived quality and why it is important to consider.

Congestion, as has been noted above, leads to degradation of quality and can have significant effects of the real-time service. In Chapter 3, methods of congestion control used in various networks, such as Asynchronous Transfer Mode (ATM) and Frame Relay networks, are considered. It is to be noted that ATM and Frame Relay are connection oriented networks where it is possible to do an end-to-end allocation of bandwidth. In

packet switched networks such as IP networks, the connectionless nature means that the congestion is dynamic and unpredictable. Some protocols, which have been developed in the recent years to carry real-time services over the IP networks, are discussed: mainly Real Time Protocol (RTP) and Resource reSerVation Protocol (RSVP). The discussion highlights the shortcomings of these protocols in providing a true real-time service. This leads to a discussion of emerging technologies, such as Intserv, Diffserv, and Queuing methods, which aim to enhance the TCP/IP architecture and provide multi-service.

It turns out that even with bandwidth partitioning, congestion may occur within a class. This is particularly a problem in adaptive classes of traffic where allocation is not made on a per-flow basis. Given that networks conditions change rapidly, the congestion control system must be very responsive. In Chapter 4, an analysis of a reactive congestion control scheme is presented. The process of detecting congestion, sending a feedback notification and adaptive measures taken by the receiving end nodes is considered in detail. This is followed by a proposal of an ideal behaviour that would be suitable for continuous media traffic such that the perceived quality is maintained. These four chapters present the research carried out to investigate the requirement for congestion control methods for adaptive continuous media. The author's hypothesis is that a responsive but graceful congestion control method is required for maintaining a high perceived quality of a real-time service and that the current methods of congestion control will fall short of this requirement.

A number of experiments, carried out in a simulated environment, are used to verify the hypothesis. However, before presenting the experimental work the description of the simulation environment and the framework of the network model is presented in Chapter 5.

The chapter also discusses the tests designed to evaluate the performance and suitability of the control algorithms.

In Chapter 6, the simulation descriptions of modelling Hysteresis and RED, two widely used methods for congestion control in data traffic, are presented followed by the respective results. The results show that performances of these methods, which are based on average buffer-fill, are not adequate in keeping the response graceful and losses to a minimum. A proposal is made for improvements through monitoring a high percentile of the queue occupancy.

This novel technique is discussed in detail in Chapter 7. The design of the algorithm is presented along with improvements made as it was tested and refined. Finally, the developed algorithm is simulated on the same network as the previous two methods and tested. The results demonstrate that this technique is more suitable for the adaptive continuous media due to the graceful degradation during congested conditions. A slow degradation is perceived to be better than a rapid fluctuation in the quality. It is shown that percentile monitoring outperforms the other two methods in terms of minimising losses as well.

Chapter 8 builds the bridge between the work done at Oxford Brookes University involving individual flows that were capable of changing their data rates to the scenario used at Nortel Networks where the flows may be an aggregate of other flows and may not have direct control over data rate regulation. A typical hierarchical network is presented and congestion problem at various points is considered along with the inherent issues for that part of network.

In Chapter 9 we enter the world of aggregate flows, where the congestion notification comes from a carrier network (a network dealing with aggregated flows) and

has to be used by, say ISPs, who may not have the right to control individual flows using their service. Also, it must be noted that with thousands of flows in the network, per-flow signalling like that used for feedback based on high-percentile monitoring is not desirable here due to significant overheads. The chapter starts with a description of the Dynamic Resource Control (DRC) mechanism, which was developed at Nortel and forms the basis of the author's work. The concept of congestion pricing and its suitability in carrier networks is introduced. The focus is again on adaptive flows and benefits of statistical multiplexing are explained. Finally, two main issues are highlighted: the problem of congestion control at the Ingress router for the carrier network; and the need for measuring the burstiness of the flow instead of just the mean.

Chapter 10 begins with a brief description of the terminology and the new concepts introduced with the pricing concept. The specifications of the pricing method, such as fairness for bursty and smooth flows, minimal signalling overheads etc are laid out. The details of algorithm development and improvements are then presented. First tests are carried out in a fluid-flow model until the algorithm is fully developed. Finally full simulation using packet generators and mean and variance metering is presented along with the results.

This thesis makes two main contributions to algorithms for adaptive flow congestion control. The percentile monitoring method provides a responsive feedback mechanism, which meets the requirements of continuous media and is shown to outperform the traditional methods used for data traffic. The mean and variance pricing method enables us to charge for burstiness of a flow as well as its mean ensuring that a flow is charged for its contribution to the load on the network. However, one achievement leads to more ideas that need to be investigated and problems that need to be solved. The constraints of time

meant that not all of them could be accomplished. The final chapter, Chapter 11 draws the conclusions of the research and gives some recommendations for further work.

2

QoS Requirements of Continuous Media

Quality of Service (QoS) requirements vary significantly from one application to another, and from one medium to another. Currently, all traffic on the Internet is treated in the same way and is mostly carried over networks geared towards supporting delay-insensitive traffic. Data traffic such as e-mail and web traffic are appropriately dealt with in this way. We all experience a delay of a few seconds or even minutes while downloading a web page or receiving an e-mail. The objective is that when it is received the integrity of the data must be maintained. Therefore, in the event of packet losses it is possible to correct the mistakes by sending the lost packets again, which increases the delay.

Continuous media traffic has different characteristics and, therefore, requires a different type of service. In order to develop a network that can fulfil the requirements of continuous media, it is important to be able to specify, measure and assess the QoS requirements both in terms of the network performance metrics and from the user's impression of performance. This chapter discusses the QoS requirements set by the type of media, application, network QoS parameters and the perceived quality, i.e., the quality expected by the user.

2.1 QoS Parameters

Most multimedia applications are a combination of, broadly speaking, voice, video, and computer data (e.g. text). TCP provides reliable transmission of data over an unreliable IP datagram network through use of congestion windows and retransmission of lost packets [RFC 2581] (see Section 3.1.3 for a discussion of TCP congestion control). Continuous media applications place different demands on the network and need to be treated differently from computer data. The requirements of continuous media applications in terms of bandwidth, delay and loss and comparisons with computer data are discussed in the following sub-sections. References will be made to TCP/IP in order to illustrate the differences between continuous media and delay insensitive data traffic and also because TCP/IP is a familiar and prevalent architecture.

2.1.1 Bandwidth

The precise bandwidth requirements of continuous media applications are governed by the type of application itself, as will be discussed in detail later. However, continuous media traffic has some fundamental differences from non real-time data traffic.

- Continuous media applications are "rate aware" and need sufficient bandwidth to transmit the data at an acceptable bit rate. The actual bit rate required may differ for each individual source depending upon the required resolution and temporal rate but it must be available for the duration of the connection[1]. For example, a video-conferencing session that requires an average of 10Mbit/s bandwidth for a duration of 20 minutes will use 10Mbit/s on average even if more bandwidth was available.

[1] "Connection" is used here in the generic sense of the word and implies an active session, which can be applicable to both connection oriented and connectionless paradigms.

- Computer data traffic, on the other hand, is not rate aware and will generally make use of all of the available bandwidth, e.g. a file transfer which would take 5 minutes if 1 Mbit/s is available would take approximately 1 minute if the bandwidth was 5Mbit/s. Congestion windows in TCP are used to regulate the amount of data transferred by changing the window size.

2.1.2 End to End Delay

The end-to-end delay encountered in transmission of a packet is calculated from the time of sampling at the source to the time that the end-user receives the sample. This includes the time taken in compression, the transmission time across the network and the time taken in processing, decompressing etc. With data traffic, latency is generally not a critical issue. All the packets do not have to arrive at the destination within a fixed period or in the correct order although correct and timely delivery is desirable. With real-time traffic, however, it is very important that the delay is bounded. For example, for a video service, the frames have to arrive at the destination in the correct order and with appropriate timeliness to ensure smooth playout. For interactive applications, the acceptable value of end-to-end delay (including the propagation delay) is in the range of 150-400ms [Fluckiger 95]. Generally, for conversational purposes, a delay of approximately up to 3 times the average interarrival time (the time between two subsequent frames) can be tolerated. A longer delay may cause annoyance to the users. The perceived quality and subjective assessment are discussed later in this chapter in Section 2.3.

2.1.3 Jitter

The variation in end-to-end delay is known as jitter. The end-to-end delay consists of the propagation delay (D_p) due to the physical distance, and delay in processing of data etc and

queuing delay (D_q) which is the delay encountered by a packet at each node while waiting to be served. While propagation delay is generally constant, queuing delay may change rapidly even in short time-scales as the network conditions change and the number of packets going through a particular node changes. This causes variation in the end-to-end delay. Consequently, successive packets or frames experience different delays in the network, see Figure 2.1. In this figure, D denotes the end-to-end delay. It is assumed that D_p remains constant and therefore the variation in D, shown in the figure, is solely due to the variations in D_q.

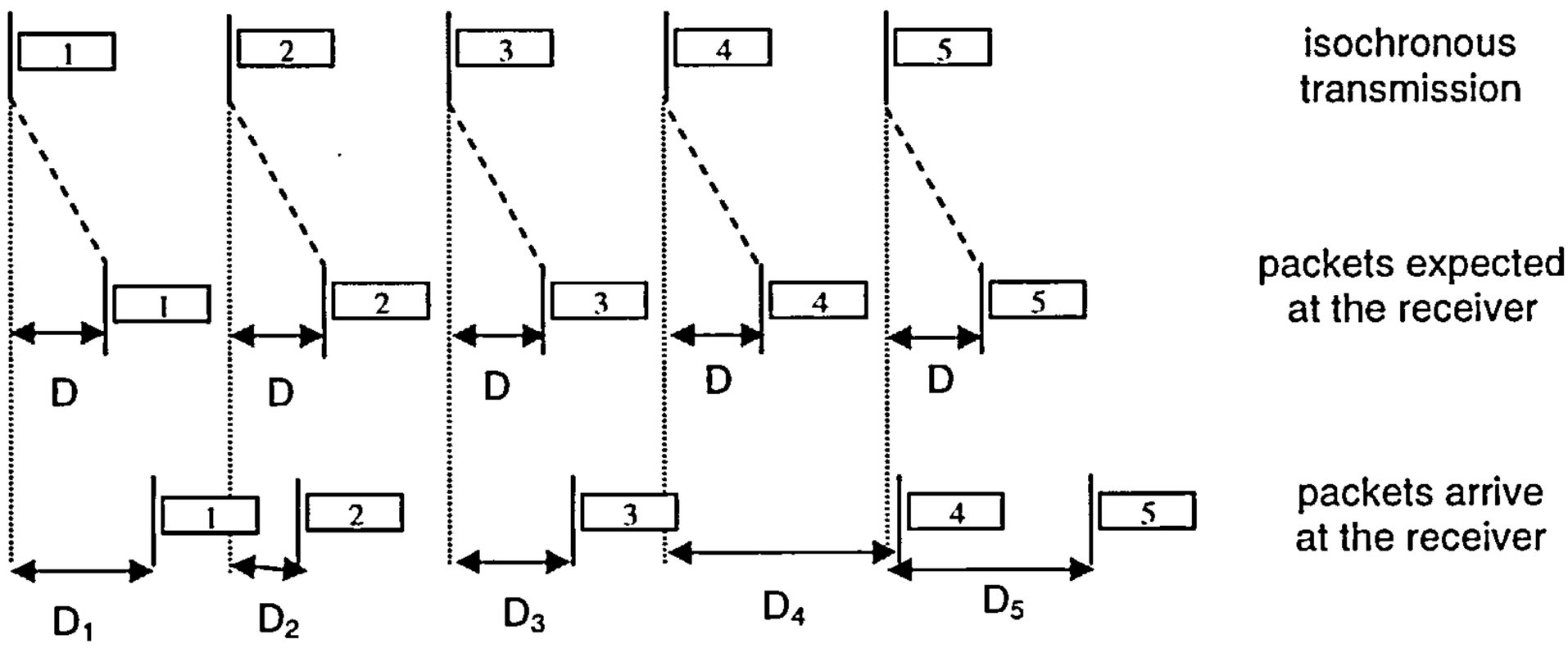

Figure 2.1: Illustration of jitter in the packets received

Figure 2.1 illustrates isochronous transmission of five packets. Assuming that the end-to-end delay in the network is D, all packets will be expected to arrive at the receiver after a delay D. However, the actual delay that is encountered by each packet by the time it arrives at the receiver, happens to be a changing value, D_1, D_2 ... D_5. We can see that while D_4 is much larger than D, D_2 is in fact smaller than D. This can cause a major problem at the receiver because while some frames are excessively delayed, others arrive before they are

expected. In a jitter-free network, the end-to-end delay (D) could be known a priori and could be compensated for. If all the packets were delayed by D, an isochronous reception would still be possible. The presence of jitter means that the delay is not known and frames have to be buffered for a longer period of time in order to cope with the worst possible delays.

To illustrate this, consider video media. With a constant playback rate and no smoothing, the receiver requires a frame every $1/f$ seconds where f is the frame rate; so if the frame rate is 25 frames/s, the receiver would expect to receive a frame every 40ms. If there are queuing delays in the network, some frames could arrive within 35ms of each other while at other times there may be a gap of up to 50ms between successive frames. This means that to restore the correct frame rate, the frames will have to be buffered for 20ms (twice the average jitter).

There is a general consensus that a jitter of up to 10ms (expected delay ± 5ms) is tolerable for video but larger variations can affect the decoding of the frames leading to jerky images, which may be found to be annoying to the user. Often a buffer is used at the receiver so that the inter-arrival times can be reinstated if the frames encounter variable delays. This requires an a priori bound on jitter for guaranteed services. However, this is not always applicable as using a buffer that is large enough may add to the end-to-end delay, which may exceed the acceptable delay for some interactive applications.

2.1.4 Loss and Error

In packet switched networks, losses are caused mainly by buffer overflow in the network and may have a significant impact on the perceived quality. Computer data traffic is highly sensitive to losses but generally not very sensitive to delay. TCP/IP guarantees a reliable delivery of data by ensuring that if the packets are lost or corrupted, they are re-transmitted.

At the receiving end, the packets are re-ordered as they may have arrived in an order different from how they were sent, especially if some of the packets had to be re-sent. This, however, is not usually an option with interactive video or live broadcasts that are time critical. Such traffic is highly sensitive to delay. Retransmission of lost or corrupted packets is not useful because the information may have become out of date by the time it is re-sent. In order to prevent losses, large buffers may be used but this adds to the delay. This trade-off between loss and delay is a very important issue because for all the real time services a packet that is delayed beyond an acceptable limit is just as useless as the one that has been lost. Nevertheless, some loss, although not desirable, is acceptable depending upon the nature of the application. This is because the human brain is highly efficient at interpreting the information presented to it. We often tend to complete or refine a picture if the resolution is low. The acceptable loss or error rate depends on several factors such as compression scheme, duration of loss interval, relative importance of lost data etc. It has been reported in the literature that the general loss requirement at the network level is expected to be in the range of 10^{-2}-10^{-6} bits.

2.2 Media Type Requirements

Within the broad class of continuous media, there are many different sets of requirements posed by video and audio. Traffic management becomes even more difficult when there is a mix of media qualities, for example a low resolution voice and high resolution music in the same application. In the following sections we look closely at the components of bandwidth requirements set by video and audio.

2.2.1 Video

Depending upon the chosen resolution and refresh/sampling rate, bandwidth requirements can vary considerably. For example with video transmission, there are four dimensions: height and width of the individual image, i.e., the spatial resolution, the frame rate that defines the temporal resolution, and colour depth. The spatial resolution of a picture may vary depending upon the application e.g. from 64x96 pixels to 512x768 pixels. The temporal resolution, or the frame rate, can vary from, say, 5 frames per second (for some applications) to full 25 or 30 frames per second[2]. The colour depth of a picture can be chosen from 1 bit/pixel (Black & White), 8 bit/pixel (Greyscale), 8 bit/pixel (Bit Mapped Colour) to 24 bit/pixel (Full Colour). As a result, the bandwidth requirement of a non-compressed image can range from 245,760 bits/s to 235,929,600 bits/s [Williams 92]. This bandwidth requirement is clearly much higher than most current networks can provide. For this reason, video data is almost always compressed before it is sent over a network. A number of standards are in use for compression of video with synchronised audio. The standards that are most widely used are those specified by the Moving Pictures Expert Group (MPEG). We shall return to the MPEG compression techniques later. Before that, we look at the requirements of audio.

2.2.2 Audio

Audio applications vary from telephone quality (or toll quality), which has a frequency bandwidth of 4KHz, to the full spectrum of 22KHz for broadcasting and music. Traditionally audio compression has been achieved through sampling and quantisation. Nyquist's criterion states that to avoid aliasing problems in the signal, the analogue signal

[2] The playback rate of video varies from one format to another but the two widely used standards are PAL (in Europe and NTSC (in North America). PAL requires 25 frames per second whereas the NTSC standard uses 30 frames per second.

has to be sampled at twice the maximum frequency. Hence the sampling rate ranges from 8KHz to 44KHz. Each sample then has to be converted into a digital value by using quantisation which can be 8-bit for toll quality or higher. 8-bit quantisation means dividing the peak sample value to an 8-bit binary value and using it as a reference so that the rest of the samples can be assigned a binary value. Sampling at 8KHz and using 8 bit quantisation is a technique known as Pulse Code Modulation (PCM) and gives a data rate of 64kbit/s, which is adequate for telephone quality. For higher quality audio 16 or even 24 bit quantisation can be used. Compression techniques implemented by Advanced Differential Pulse Code Modulation (ADPCM) are often used here, which use differential levels for quantisation and hence give better signal resolution. Also, audio can be transmitted in mono or stereo. Stereo transmission will require twice as much bandwidth as mono. A 16-bit full spectrum stereo audio signal can typically require a bandwidth of 1.4Mbit/s.

2.2.3 MPEG Encoding for Audio and Video

There are many standards available for encoding audio and video signals depending upon the application. The main standards recommended by ITU-T are H.261 and H.263 for video conferencing and low bit-rate communications respectively. However, the standards that are widespread and gaining much popularity are those defined by the Moving Pictures Experts Group (MPEG). MPEG standards have been shown to be more efficient through use of motion compensation. The development of standards is an evolving and active area of research with interests from all industry domains with a stake in digital audio, video and multimedia and that includes Internet services as well as the broadcasting industry. Detailed study of MPEG standards and their individual benefits is too vast to come under the scope of this dissertation but the author attempts to make a brief introduction and lead

on to application on packet switched networks. Among others, the book by Ghanbari [Ghanbari 99] gives a substantial grounding in the subject.

The MPEG standards are ISO/IEC standards developed by MPEG (Moving Pictures Experts Group). MPEG-1 and MPEG-2 made interactive video on CD-ROM and Digital Television possible. MPEG-1 was the first standard to be developed for coding of moving pictures and associated audio for digital storage media. It is currently available and supports products such as Video CD and MP3. The latter, in particular, has revolutionised the music industry. MPEG-2 defines generic coding of moving pictures and associated audio. Later on MPEG-4, MPEG-7 and MPEG-21 were added. MPEG-4 is for multimedia for the web and mobility and provides the standardized technological elements enabling the integration of production, distribution and content access for digital television and interactive graphics. MPEG-7 is the standard to describe the multimedia content data that will support some degree of interpretation of information, which can be passed on to, or accessed by a device or a computer software. A number of references and documents related to MPEG can be found on the Internet [MPEG 00b, MPEG 00a].

MPEG standards that are being currently developed are more likely to be used for the services over the future Internet, particularly MPEG-4, which will support different bit rates. However, a lot of research is going on in understanding the traffic patterns of MPEG-1 and MPEG-2 over the network [Sikora 97]. In this work, we shall focus on MPEG-1 as the traffic characteristics are more widely known and we believe that our methods will be applicable to other standards as well. We now consider the coding process of audio and video in MPEG-1.

The main feature of MPEG as a compression technique is that it uses human perception in order to decide what information is essential and what can be discarded. For

example, in order to reduce an audio stream instead of simply decreasing the quantisation from 16 to 8-bit, MPEG uses "perceptual coding." When we hear anything, the brain analyses sound and filters out irrelevant information. So, if there is a strong signal then the weaker signal behind it is not "heard", because the brain has filtered it out. The MPEG-1 audio codec uses this psycho-acoustic attribute through 'auditory masking', where parts of a signal are not audible due to the function of the human auditory system. For example, if there is a sound that consists mainly of one frequency, all other sounds that consist of a frequency close by but are much quieter will not be heard. The parts of the signal that are masked are commonly called "irrelevant", as opposed to parts of the signal that are removed by a lossless coding operation, which are termed "redundant". The codec contains a psycho-acoustic model that analyses the input audio signal into the spectral components and then applies human auditory masking model. Further references and details can be found in [Thom 99].

As in audio compression mentioned above, the MPEG-1 video coding scheme treats different pictures in a sequence in different ways. A video image is essentially a periodically updated picture from a sequence of still pictures. Some pictures are considered as being more important than the others, for example the ones after scene-cuts, and are coded with as much information as possible while others are coded with more distortion. At other times the bulk of the pictures can be coded coarsely with the differential information and provided that there is an adequate number of fully coded pictures at regular intervals the picture sequence can be decoded correctly. The pictures are classified into four categories: I, P, B and D, see Figure 2.2. I-pictures are the least compressed, contain all the data to reconstruct the picture and act as reference frames. P-pictures (predictive) are derived from the previous picture, and B-pictures (bi-directional) are interpolated from the

pictures both in front and behind them and are the most heavily compressed. D-pictures are intended for fast visible search mode and not included in the output bit stream. I-pictures act as synchronisation points for transmission so that, say, after every 10 P or B-pictures there should be an I-picture. Hence, the time between two successive I-pictures is usually in the order of 300 to 400ms. One reference picture, either an I or a P picture should occur ever 150ms [Fluckiger 95].

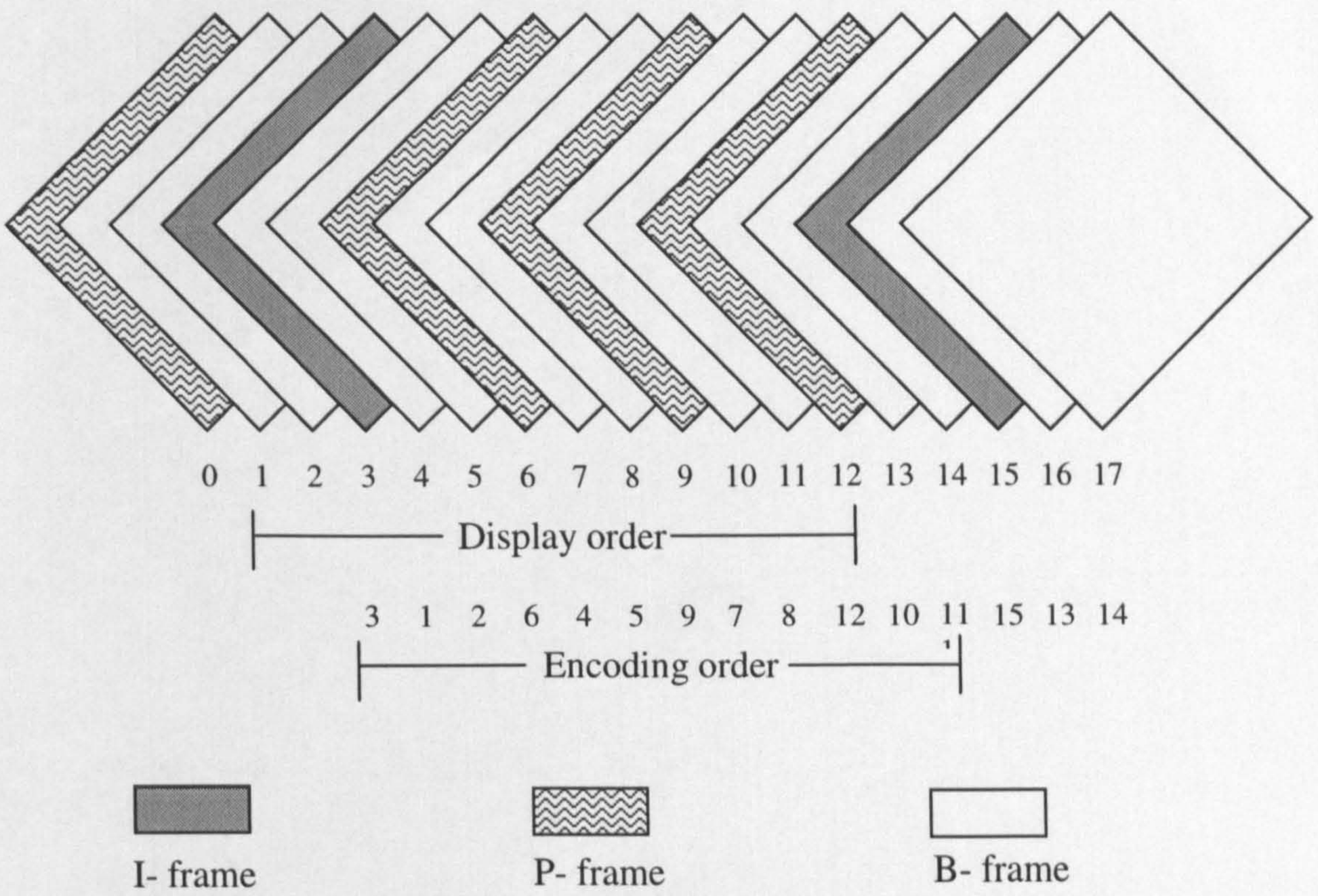

Figure 2.2: An Example of MPEG encoding [Ghanbari 99]

Because B-pictures are bi-directional, they have to be encoded last. Therefore, the pictures have to be buffered and reordered before encoding as shown in Figure 2.2. When the decoding end receives the bit stream, it will have to use buffers as well to ensure that it can use I and P-pictures to update the information.

The output from an MPEG encoder is a bit stream. The standards do not specify how the packetisation should be carried out, although in case of MPEG-2, the usual practice is to packetise them in 188-byte fixed length packets[3] (which includes a 4-byte header) so that each packet can be neatly divided into four 47 bytes segments to be accommodated in four ATM cell payloads. To the author's knowledge, packetising for variable length IP packets has not been implemented. Currently MPEG-1 and 2 streams are sent over the internet but they are used for store-and-play type applications. For example, the software decoder such as Real Player receives the bit stream into buffers, decodes it into pictures and re-orders the pictures to playout the image sequence. Much research is underway for transmission of MPEG over packet switched networks such as IP [Kanakia 93, Dagiuklas 96, Wrege 96, Willebeek-LeMair 97, Ramanujan 97, Tater 00a, Baldi 00].

Most researchers assume that one picture is contained in one packet. But, a Group of Pictures (GOP), typically 12 pictures or more may be encoded as 1 packet. This is particularly useful for multimedia storage devices but results in long packets typically of several 100K bytes. The traffic characteristics in these two cases (one picture per packet and one GOP per packet) are very different from each other and significant for rate based flow control [Lougher 92]. The actual packet rate that is transmitted will depend upon the codec implementation. In MPEG encoded video, an I-picture is the only one that can be generated independent of any other pictures. In Figure 2.2, the encoding order shows that pictures 1 and 2 are encoded after picture 3 has been encoded and similarly pictures 4 and 5 are encoded after picture 6 has been encoded. The characteristics of the resulting traffic will be different as described below.

[3] In digital TV broadcasting, often a 16-byte header is added to an MPEG packet, thus creating a 204-byte packet used in satellite and terrestrial transmissions.

- If the pictures are sent as soon as they are encoded, it may result into multiple pictures being sent together with a longer interval; e.g., picture 3, 1, 2 would be sent as soon as picture 2 has been coded followed by a delay until picture 6 is encoded and sent.

- Alternatively, if the encoded pictures are transmitted with appropriate interval re-instated between successive pictures such that after encoding pictures 1 and 2, picture 3 is sent when picture 4 is encoded, but pictures 1 and 2 are sent after successive delays while 5 and 6 are encoded.

It seems obvious to use the second method because for the price of buffering two pictures it will induce less variance in the output traffic. In the experiments that will follow, the traffic generated by this method will be used.

Finally, a brief look at the demands on the bandwidth requirements made by these standards. All standards have their own constraints and requirements. For example, MPEG-1 supports Constant Bit Rate whereas MPEG-2 and MPEG-4 support Variable Bit Rate. That is how a 2-hour movie encoded in MPEG-2 that requires 9Mbit/s can still fit on a DVD taking 675MB instead of 8.1GB. Table 2.1 summarizes the bandwidth requirements of some audio and video standards.

Table 2.1: Typical Bandwidth Requirements

Channel	Bandwidth
Standard Video	25 Mbit/s
MPEG-1	1.5 Mbit/s
MPEG-2	9 Mbit/s
MPEG-4 low	64 kbit/s
MPEG-4 intermediate	64 – 384 kbit/s
MPEG-4 high	0.384 – 4 Mbit/s
Voice Audio	64 kbit/s
Hi-Fi Audio	2Mbit/s

2.3 Application Requirements

Multimedia services range from simple multimedia mail to high demand services such as real time video, interactive learning etc. All multimedia applications do not have the same set of requirements because it depends on the objectives of the application. A high-resolution signal requires higher bandwidth, which can be expensive and often difficult to provide. Then there are the problems of jitter, delay and losses. A larger buffer would minimise jitter and losses but would increase the end-to-end delay. The priorities regarding whether to keep the delay minimised or to ensure loss free service depends upon the application rather than the media being used. Consider the requirements of a video player and an editor. The main requirement of a video player is to accurately synchronise between voice and video. The emphasis is on minimising jitter so that the user can enjoy a smooth reproduction even at the expense of some minor losses. However, the video editor requires each picture to be seen without errors for editing purposes. A jerky playout, while it may be annoying, is less of an issue in this case. Requirements also differ depending upon the user's expectation from the service. Standard TV and High Definition Television (HDTV) are both entertainment services but the requirements of HDTV are higher because of better

resolution and quality. Typical values of picture rate, resolution and colour depth for different types of video services are given in Table 2.2.

Table 2.2: An Overview of different Video Application Requirements

Application	Temporal Resolution	Spatial Resolution	Colour Depth bits/pixel
Multimedia	15	320 x 240	16
Entertainment TV	25	640 x 480	16
HDTV	60	1920 x 1250	24
Video Telephony	10	320 x 240	12
Surveillance	5	640 x 480	12

In broad terms, however, most continuous media applications can be categorised into three main types viz. interactive, storage and retrieval, and surveillance. These are discussed in detail in the following sub-sections. Of course, with the explosion in multimedia services there will be a number of applications that do not fall clearly into any of these categories.

2.3.1 Interactive Applications

The demands of interactive applications can escalate sharply in terms of resource and guarantee requirements as the information being transmitted becomes more complex. A simple head and shoulder video is relatively less demanding than an interactive learning course where the lecturer is explaining a complex biological phenomenon or a detailed structure. While with head and shoulder images we can compromise through encoding the background coarsely and focussing on main gestures, with the latter it is important that the whole scene is accurately reproduced. It is also important that the audio and video are

synchronised and there is no perceptible lag. Hence, tight bounds on delay are extremely important.

Such applications are being increasingly popular as they help in academia with distance learning, in medicine in the form of telesurgery and at work through Computer Supported Cooperative Work (CSCW) to name a few. Usually this type of service is of interest to organisations, which may be prepared to pay the premium for a fast service in order to save cost of travel and time incurred by meetings and conferences. Often there is a need for sharing a whiteboard or presentation in real time. At present, if organisations require such service they use a dedicated link or otherwise they find it difficult to synchronise.

The problem is that it is not cheap to use a dedicated link and when it is used, it may be a waste of resources. The traffic being generated from such applications will be difficult to predict. The system may be able to guess at jitter from the interarrival time between packets. However, the statistics of the packet lengths will not be known. Often these traffic streams are sent using Variable bit rate or adaptive encoding may be used which all contributes to the problem of not knowing how to allocate bandwidth.

2.3.2 *Storage and Retrieval*

Services such as Video on Demand (VoD) are mainly for entertainment purposes and have to match the quality provided by a conventional video player. Since there is minimal interaction, the absolute latency is not important but the jitter has to be minimised. The frames to be displayed in future can be downloaded in the background during the playback. A large smoothing buffer is often used to overcome jitter. However, the resolution and picture size required are quite high as the user will expect full size TV or even HDTV resolution.

Sometimes, applications such as multimedia tutorials are run using stored information. Often there is interaction required as well, for example the user may have to type or say something. The latency has to be low and the information requested by the user should not take an excessively long time. The image quality for the diagrams can be lower than that for entertainment standard video and normally 8-bit colour images can be used. The instructor's voice could be low quality whereas music samples will have to be of high quality. The mix in media qualities can lead to difficulties in traffic management. It may be possible to provide the service by downloading the information and decompressing in non-real time and then playing it out in real time in the local environment. However, this may not be very practical as it would require large buffers to store the information and the user may have to wait while the downloaded information gets processed. It will also hamper the interactivity.

2.3.3 Surveillance

Applications such as Closed Circuit Television (CCTV) come under this category. They can use black and white images of low resolution as long as the area under surveillance is clearly visible. Also, the temporal rate can be quite low, say 5 frames/s (see Table 2.2). Although this would lead to a sequence of disjointed images, the objective with this type of application is to get information and not entertainment. The system could be designed so that when there is some activity, the frame rate and resolution increases for better surveillance.

2.4 Perceived Quality

As with any other service, it is the end user, who ultimately determines the success of a multimedia service offered on a network. This subjective attribute of the service is both

important since this is the key to success, and difficult to assess as it depends upon a number of different conditions including user's expectation, experience, and mood which are difficult to estimate. An analogy can be drawn to a customer who goes to fast food cafeteria for a meal and one who goes to an expensive restaurant. The former is after a cheap and fast meal and has accepted that the menu will be limited and the food might be of mediocre quality, whereas the latter is prepared to wait longer and pay more but also expects a full range of menu and a high standard of service. In the end they both want food but their expectations are entirely different. Likewise, continuous media applications come with a range of quality expectations associated with them depending upon the user and their purpose.

Transmission over packet switched networks mostly involves trade-offs and it is generally not possible to provide both high quality and minimum delay service because the network conditions are beyond control in a connectionless system. Research in user perception has suggested that from the end users' subjective point of view, consistently acceptable overall quality is more important than a good image quality with interruptions [Watson 97, Watson 98, Bouch 99]. As anyone who makes long distance international telephone calls will be very familiar, a perceptible delay is distracting and not conducive to a normal conversation. Neither is it easy to talk normally if significant chunks of words are lost in transit. In case of video, it has been found that the variation in delay between consecutive packets, due to jitter, is also extremely important.

There is a great deal of literature addressing QoS issues. However, the emphasis has been mainly on the network level such as bandwidth allocation, call admission etc whereas the end user's point of view has been somewhat overlooked. It is vital to carry out a subjective assessment of QoS. The opinion in the networking community is divided. Some

are confident that, eventually, improvements in methods such as bandwidth reservation, e.g., RSVP will resolve the QoS issues [Zhang 93, RFC 2205] while others believe that there will be a demand for lower quality at lower cost [Podolsky 98]. A number of ITU-T recommendations have been specified for speech [ITU 98b] and video [ITU 98a] which address the issues such as listening effort and conversation difficulty etc but it is felt by some that these scales are no longer adequate for assessing the wide range of new services currently on offer. User perception trials carried out by [Watson 96, Watson 98] highlight the QoS issues from the end user's subjective point of view. It is important that for a multimedia service to be successful the perceived quality is taken into account. However, because of the subjective nature of concepts like "good resolution", it is difficult to map the perceived quality to network parameters that can be configured. Research has shown that users are most likely to notice effects of jitter, which relates to the temporal characteristics of the continuous media flow and network [Ball 96b].

2.5 Summary

The fundamental differences between computer data and continuous media traffic have been highlighted. Most of the chapter has drawn examples relating to video but usually, and especially due to the popularity of using MPEG encoding for both audio and video, similar concepts can be applied to audio as well.

Continuous Media applications are inherently rate-aware which implies that their requirements are significantly different from those of computer data. Existing packet switched networks, such as IP networks, have been developed for non-rate aware data traffic. Increasingly more applications are providing a mix of media types and hence their requirements regarding bandwidth, jitter, delay and loss etc are governed by the objective of the application. A lot of research is underway on improving the network while another

direction is into the Perceived Quality or the end user's expectation from the service. Since the end user is the one who ultimately decides whether a service is good and whether it is worth paying for, it is important to account for it when developing methods of traffic management and control. Current networks are not equipped to meet all the demands posed by a plethora of multimedia services.

Irrespective of how well we tie up the perceived quality with network QoS parameters, it is certain that congestion in the network will affect the perceived quality in some way and hence control is important. It has been felt that a large majority of existing and emerging multimedia services require guarantees of timely delivery of packets, low jitter and minimal loss. While a high spatial resolution is definitely desirable, it has been found that it is important to maintain a consistent temporal resolution to ensure improvement in perceived quality. Fluctuations in temporal resolution are often due to jitter, which in turn is caused by variations in queuing delays.

Of course, if a guaranteed service can be provided by the means of bandwidth allocation, it can be ensured that the packets are transmitted in correct time regardless of the network conditions. However, for such deterministic guarantees it is essential to know delay and jitter a priori. This is difficult as the network conditions may change rapidly over time. Therefore, it is important to control the congestion so that even the services that do not employ rigid bandwidth reservation are of tolerable quality. In the next chapter, we will discuss congestion control schemes used in a number of existing networks with emphasis on IP networks. An analysis of the control methods as well as a possible scenario for future IP networks will also be presented.

3

Congestion Avoidance and Control

In current packet switched networks the changes in network conditions have a direct effect on the QoS of existing flows. As the amount of traffic on the network changes, packets may experience different delays leading to jitter at the receiving end. However, it is essential for good quality continuous media applications that delay and jitter are bounded. Delays and losses in the network are often caused by congestion that occurs when the traffic arriving at the node exceeds its service rate and the length of time that a packet has to wait before service exceeds a given limit. In a node capable of multi-service, such a situation may develop on a per-class basis. For example, a guaranteed class may not suffer from congestion while flows in an adaptive class experience delay. It is logical to assume that the multi-service architecture will protect guaranteed classes followed by adaptive classes and finally a number of data classes in that order. Therefore, a situation where best-effort traffic sees no congestion while adaptive video has to suffer deterioration should not arise.

Nevertheless, it is important to control the network congestion within each class in order to provide QoS. In this chapter, first we discuss how congestion control is exercised in the existing networks such as Asynchronous Transfer Mode (ATM), Frame Relay and IP Networks. We then analyse the different aspects of the congestion control process that will

be highlighted by the discussion of existing techniques. Obviously, most of these networks do not exercise classification of traffic or per class congestion control but the concepts will be similar.

3.1 Congestion Control in Different Types of Network

Although the research presented in this thesis is mainly concerned with IP networks, it is important to understand the technologies used in other existing networks and to identify if they are complementary or contrasting. Also, it may be possible to adapt a method proved to be successful in, say ATM networks, for use in future IP networks. The following sub-sections give a brief overview of existing techniques and ongoing research in these areas.

3.1.1 ATM Networks

Development of efficient transmission of continuous media over Asynchronous Transfer Mode (ATM) networks has been an active area of research for over 10 years and the necessary traffic management mechanisms have been defined well within the ATM standards [ATM Forum 01]. ATM networks offer 4 classes of service by partitioning the bandwidth: Constant Bit Rate (CBR), Variable Bit Rate (VBR), Available Bit Rate (ABR) and Unspecified Bit Rate (UBR). CBR and VBR service classes offer QoS guarantees in terms of maximum delay and minimum throughput, and can be used for continuous media applications. Video is usually transmitted on VBR. The VBR class may have a further sub-class for real time traffic, known as real-time VBR (r-VBR). The ABR service class deals with bursty data traffic and attempts to maximise statistical multiplexing gain. There have been suggestions of using this class for bursty compressed video but most researchers have found VBR to be more appropriate due to QoS guarantees [Ball 96c, Ibrahim 98, Kalyanaraman 98].

ATM networks offer service guarantees by setting up an end-to-end virtual connection and performing per session admission control. This technique is similar to the one used in circuit switched networks. The most obvious example of circuit switching is the Public Switched Telephone Network (PSTN). When a call is placed, the entire end-to-end route is established and maintained for the duration of the call ensuring good quality. Likewise, when a flow wants to join the ATM network, it would make a connection request with its destination address, traffic parameters, and required QoS specified. The network will work out a path from ingress to egress and use Call Admission Control (CAC) at each node traversed to determine if the flow can be accepted. By accepting the connection, a node must not violate the QoS already guaranteed to the existing connections. Once the connection has been established, all the traffic from the flow will follow the same path identified by Virtual Circuit Identifier/Virtual Path Identifier (VCI/VPI). A fundamental difference is that in PSTN, the network can support a finite number of calls and any more call requests are blocked, whereas in ATM networks CAC will allow a new flow to join as long as the QoS guarantees for the existing flows are not undermined.

CAC performs congestion avoidance in a given class but is only used for CBR and occasionally VBR classes. CAC is slightly conservative but even so, there is a possibility of congestion in VBR classes. In case of congestion in any service class, ATM switches can notify the end devices of the congestion status using the Explicit Forward Congestion Notification (EFCN) marker in the ATM cell [Zheng 99]. Alternatively, specific Resource Management cells might be sent from the congested switch directly to the sources.

In the discussions in the previous sections, we have assumed that the sources are co-operative and adaptive. In real life, the sources may be greedy and may ignore the congestion notifications. To maintain QoS guarantees it is important to ensure that traffic

flows conform to the parameters agreed at the negotiation time, i.e., the traffic contract. Normally, the nodes at either end perform policing and make sure that end users abide by their traffic contract. End users may also use shaping to ensure that their traffic does not fail a policing check. In ATM networks, policing is carried out by Usage Parameter Control (UPC) often with a leaky bucket algorithm [Tannenbaum 96]. Non-conforming packets are treated as a lower priority in ATM. Likewise, RSVP polices the arriving traffic against the Traffic Specifications (Tspec) and shapes the downstream traffic pattern to the original Tspec.

Since we are mainly interested in continuous media applications, all further references to ATM Networks in the rest of this thesis imply the VBR service classes unless specified otherwise. It is assumed that even in future IP networks, there will be no class equivalent to CBR. This is because ATM uses its connection oriented approach and constant cell size to provide CBR, neither of which is available in IP networks.

3.1.2 Congestion Control in Frame Relay Networks

Frame relay service is another connection-oriented technology that evolved from X.25 packet switching and uses variable length frames to transport user traffic across an interface. It employs explicit feedback signalling in order to notify the sources about the congestion in the network.

The ultimate source or destination of data flowing through a frame relay network is often referred to as a Data Terminal Equipment (DTE). As a source device, it sends data to an interface device for encapsulation in a frame relay frame. As a destination device, it receives de-encapsulated data (i.e., the frame relay frame is stripped off, leaving only the user's data) from the interface device.

The network sets a bit in the frame during congestion. The direction of notification may be forward using Forward Explicit Congestion Notification (FECN) or backward using Backward Explicit Congestion Notification (BECN) [Bahner 98]. In case of FECN, the bit indicates to a DTE that the receiving device should initiate congestion avoidance procedures. It could then, for example, send a notification to the sender. On the other hand if BECN is being used the congested node notifies the sender directly and instructs it to initiate congestion avoidance procedures.

3.1.3 Congestion Control in IP Networks

IP networks form the majority of today's Internet. They have traditionally offered a best-effort service with no distinction between traffic types and requirements and they do not provide any service guarantees other than not to delay a packet if it can be avoided. In the widely used TCP/IP architecture, the Transport Layer TCP is meant to provide some reliability of delivery and ensure the integrity of the received data. TCP flow control uses a variable length sliding window protocol. Dynamic flow control has been extensively researched and many flavours are now available [RFC 2581].

We consider the simple Slow Start Congestion Avoidance TCP algorithm. In this case, the TCP sender keeps a track of the congestion window. The sender starts with a congestion window size equivalent to one TCP segment, it sends the segment, and then waits for an acknowledgement. When it receives the acknowledgement through an ACK (usually piggybacked on a TCP data segment), it doubles the congestion window to two segments and sends the next two segments. When it has received the ACK for these two segments, it doubles the window size again to four segments. Hence, as the session starts the congestion window is increased geometrically, "slow-start phase." Now, consider that congestion occurs and the fifth segment sent by the sender is lost. The sender then detects a

loss because the waiting time for the corresponding ACK will expire. This segment is re-transmitted and TCP resets the congestion window size to one segment and the "slow-start" process repeats. The "slow-start" phase is limited to the threshold where the congestion window reaches one-half the size of the window reported by the receiver. Beyond this, the congestion window size is not doubled when an ACK is received; instead it is increased by one segment. This linear phase is called the "congestion-avoidance" phase. As before, if an ACK is not received in time, the congestion window is reset to one segment [McDysan 00].

Much research has been carried out for congestion control in TCP and its various flavours [Floyd 94, Romanow 95, Floyd 98, Bansal 01]. However, TCP is designed to react to packet losses and to ensure that packets arrive at the destination without errors even if they are delayed. This is not usually suitable for continuous media traffic, which requires timely delivery, and hence such traffic is transmitted using Real Time Protocol (RTP) over the IP networks. RTP does not offer any guarantees but attempts to deliver the packets with minimal overheads without constraints regarding transmission and receipt at the destination.

In addition to this, Internet Engineering Task Force (IETF) has proposed a Resource reSerVation Protocol (RSVP) that would overlay and not modify the fundamental connectionless paradigm of IP [RFC 2205]. In the following sub-sections, RTP and RSVP are discussed in more detail.

3.1.3.1 Real Time Protocol

Real Time Protocol (RTP) was designed for transmission of continuous media traffic over the Internet. RTP is a transport layer protocol that runs with User Datagram Protocol (UDP) on top of IP in the TCP/IP architecture. Together, RTP and UDP form the transport layer. However, unlike TCP, the transport layer for data traffic, RTP and UDP do not

provide reliability in terms of error-free delivery. The packets are simply forwarded as quickly as possible towards the destination. RTP does not ensure timely delivery or provide other quality-of-service guarantees, but relies on lower-layer services to do so. All that RTP provides is a sequencing number for the packets, which can be used at destination to re-instate the correct order. Obviously, as different packets may take different routes and experience different delays, they may arrive out of order at the receiving end. The receiver normally has to store the packets in a buffer for smoothing jitter and re-ordering. The sequence numbers included in the RTP header allow the receiver to reconstruct the sender's packet sequence. RTP is augmented by an RTP Control Protocol (RTCP), which monitors the session and conveys information about active participants. RTCP provides a means by which each participant can monitor the number of participants in a session and a "loose" form of session control where members join in and leave with minimal negotiations. It allows for synchronisation between a number of streams using different UDP ports. The specifications for RTCP and RTP are in [RFC 1889].

With respect to congestion control, the primary function of RTCP is to provide feedback on the quality[4] of the data distribution. This is an integral part of the RTP's role as a transport protocol and is related to the flow and congestion control functions of other transport protocols. Some suggestions have been made to send the feedback directly to the adaptive encoders [Busse 95]. Feedback from receivers is also conveyed through RTCP sender and receiver reports. This is useful in large-scale transmission so that whoever is experiencing problems can identify whether the problem is local or global. In addition to various time stamps and information about lost packets, RTCP reports contain an estimate of "inter-arrival jitter", defined as the mean deviation (smoothed absolute value) of the

[4] This is not related to the perceived quality.

difference in packet spacing at the receiver compared to the sender for a pair of packets. This is equivalent to the difference in the "relative transit time" for the two packets; the relative transit time is the difference between a packet's RTP timestamp and the receiver's clock at the time of arrival, measured in the same unit.

The inter-arrival jitter field provides a short-term measure of network congestion. Packet loss tracks persistent congestion while the jitter measure tracks transient congestion. The jitter measure may indicate the congestion before it leads to packet loss. Since the inter-arrival jitter field is only a snapshot of the jitter at the time of a report, it may be necessary to analyse a number of reports from one receiver over time or from multiple receivers, e.g., within a single network. Although the jitter measure could be used to trigger an explicit feedback, it has not been implemented in the network and the adaptation is left to the applications on the end machines.

The RTP/RTCP configuration is not suitable for providing an acceptable perceived quality in a dynamically changing network. The RTCP reports are not sent frequently enough to deal with the problems faced by continuous media traffic. At best, they are useful for day-to-day network management as the reports give an estimate of traffic loads at various times of the day. For multi-service, a much more responsive control system is required. The research into multi-service networks led to the evolution of resource reservation method applicable to IP networks. The Resource reSerVation Protocol (RSVP) was standardised by IETF and has been extensively researched.

3.1.3.2 RSVP

RSVP is a receiver-based model where each receiver is responsible for making reservations and keeping them alive. This is mainly because a source-initiated system cannot cope with heterogeneous receiver requirements in a multicast operation. Although it is a receiver

initiated reservation, the transmitting router has to inform its potential receiver(s) through Sender Template (ST) along with a description of traffic flow or Traffic specification (Tspec) and an optional Advertising specification (Adspec) for delay information using the Path message. ST effectively defines a filter specification so that the upstream routers can identify the flow and reserve resources for it. The intermediate routers use the Path message to establish the path for the following packets. Path message also keeps a record of traversing non-RSVP networks. The receiver responds to the Path message by using a Resv message, which contains flow specification (flowspec). Flowspec includes the service class of an application, Tspec derived from Path message, desired QoS or Reserve specification (Rspec), and an optional Resv Confirm (RC). At each intermediate node, the request is passed on to the node's admission control module. If it is accepted, the QoS parameters are set up in the packet classifier and the scheduler according to the flowspecs, or else if the request is rejected an error message is sent to the appropriate receiver(s). The node also forwards the reservation request upstream towards the corresponding sender. However, this may be different from the request received as the node is allowed to modify the flow spec on a hop-by-hop basis or merge the reservations from the receivers (in multicast systems) and forward the maximum flowspec.

It is possible for the clients to request an acknowledgement of the requested QoS and this request will propagate until it is rejected or it reaches a point where existing reservation is equal to or greater than that being requested. It will then not be forwarded further and sender will be sent a confirmation. This is clearly not a true confirmation and enhancements can be provided using Adspec.

The reservation process using the RSVP model is known as "soft-state" as opposed to "hard-state" reservation in the case of connection oriented networks such as ATM

[McDysan 00]. The hard-state approach ensures end-to-end bandwidth allocation and establishment of a connection whereas this is quite difficult in the soft-state approach as the reservation packets may arrive at each router along the path in different order, and particularly so in multicast. Since there is no reliable way of confirming the reservation, all the routers along the path may not be consistent in reserving the resources for a particular flow. Also, the network may deny refresh request or simply stop providing the reserved capacity midflow. This problem is inherent in a connectionless paradigm and RSVP may therefore not always be successful.

3.1.3.3 Flow Acceptance Control

Measurement Based Admission Control (MBAC) or Measurement-based Flow Acceptance Control (MFAC) is an active area of research [Casetti 96, Gibbens 97, Ball 99c, Qiu 01]. It is most suitable for guaranteed flows where it is essential that the flows receive their required data rate and bandwidth is allocated for the peak rate of each flow. However, the majority of flows cannot be given such treatment because it would be inefficient to allocate peak rate bandwidth to bursty flows. Also, the majority of the network is based on connectionless structure making it a difficult task to ensure end-to-end bandwidth allocation for every flow. The overheads that would incur mean that such services can be offered to a limited number of premium class users that are willing to pay more. In this work, we shall concentrate on improving the bandwidth provisioning for a range of flows that do not require such rigid guarantees.

There are concepts of QoS routing being developed by which, say, a guaranteed service flow is restricted to a set route and is not allowed to change, and in return the traffic on that class could be protected from the changes in the network.

3.2 Emerging Network Architectures

It has been recognised that IP networks as they stand are somewhat inadequate for the demands of heterogeneous multimedia services. The CAC function of the ATM congestion control scheme is a method of Preventive Congestion Control. By performing a CAC, the network attempts to avoid congestion altogether. RSVP is only a vehicle for requesting resources. It does not carry out resource allocation. It is obvious that some form of admission control is needed for IP networks but it has to be one that does not depend on routes and can adapt dynamically. Admission control techniques for packet switched networks are evolving, particularly for IPv6 [RFC 1883] which has a traffic class field and flow labels that can be assigned to different types of traffic flows. The traffic class field is in addition to the Type of Service (ToS) field, which exists in IPv4 but is often not used. When the IPv6 specifications were released in 1995 the flow label was experimental. The use of flow labels is still being discussed and finalised but it is assumed that with the development of architectures like Diffserv and MPLS, the flow labels will become useful in providing multi-service. Perhaps we will be able to derive an adequate method of congestion control for IP networks. However, we will still need a network capable of distinguishing between different types of data.

There have been several proposals for enhancements as well as new architectures such as Integrated Services (Intserv), Differentiated Services (Diffserv) and queuing algorithms such as Class Based Queuing (CBQ) and Weighted Fair Queuing (WFQ).

3.2.1 Intserv

The Integrated Services Working Group of IETF has undertaken the task of supporting some form of QoS guarantees through integration of IP and ATM networks known as Integrated Services or Intserv. In this approach, ATM is used as the transport layer for IP

traffic. The focus is mainly on meeting the requirements for high network performance, lowering costs and providing a well-defined network quality for the end user. Intserv networks based on IPv6 have been developed and implemented in a pan-European field trial for applications such as distance learning [Andersen 00]. The Intserv framework consists of three service levels (initially there were five): guaranteed service, controlled load service and conventional best-effort service of which the guaranteed service is best suited for continuous media. Each traffic flow is characterized at the network entry point by five parameters: token bucket size, token generation rate, peak rate, maximum datagram rate and minimum policed unit. The traffic flow provides these parameters and requests for bandwidth. The network allocates the bandwidth and guarantees a maximum packet delay (loss is zero). Resource allocation was suggested to be made using RSVP.

However, RSVP only provides a method of requesting the necessary bandwidth; it does not guarantee it. As explained in Section 3.1.3.2, RSVP requests are initiated by the receiver and this leads to problems with scalability. Following the RSVP's fall from grace due to these problems, IETF started another framework in contrast to Intserv known as Differentiated Services or Diffserv.

3.2.2 *Diffserv*

Unlike the Intserv model, Diffserv does not require a reservation protocol. Instead it relies on a mutual agreement among the Internet Service Provides (ISPs) and between an ISP and an end user. Traffic is classified while entering the network and attributed to different behaviour codes each of which is identified by a field in the IP header. This was the same as the ToS octet in IPv4. Within the Diffserv core network the packets are forwarded according to the per-hop behaviour [Bernet 00].

The Diffserv architecture is currently under development as the per-hop behaviours are being refined. Mainly there are 3 types: Expedited Forwarding, Assured Forwarding Group and Best Effort. Packets in EF class get the highest priority and the network must meet their quality requirements in terms of delay and jitter bounds as well as throughput. In the AF group, there can be a number of different levels of quality provided. The packets will mostly receive their expected quality but occasionally may not. The Best effort group is essentially similar to current IP network where there is no guarantee given. Diffserv, on its own, provides an infrastructure for transmitting different types of traffic. It requires some flavour of resource reservation mechanism to ensure that service guarantees can be met. Alternatively, it can work in conjunction with a responsive feedback system that can be used to regulate the traffic. Generally, Diffserv is more scalable than Intserv but it requires per flow signalling of the traffic class information along the path.

3.2.3 Queuing Methods

It is quite clear that future networks will have to support multiple classes of traffic through bandwidth partitioning. This would obviously lead to a number of queues, one for each class and each with different priorities. In such cases, the queuing behaviour has to be more sophisticated than First in First Out (FIFO). Class Based Queuing (CBQ) essentially separates traffic into different categories according to their service requirement or priority. The packets can then be forwarded with priority scheduling where the packets in the high priority class are always served first. For example, architectures like Diffserv can then apply EF or AF group behaviour for these classes. On the other hand, Weighted Fair Queuing (WFQ) chooses the packet with the shortest finish time from all those in the queue and serves it first. A study was carried out by [Callinan 00b] into comparison between the

two methods. The results have shown that for real time services CBQ performs better while data traffic is best served by WFQ.

3.3 Summary

This chapter started with a general discussion of congestion control techniques used in a number of modern networks. The objective was to learn how the problem of congestion is dealt with in the existing networks and whether they can be adapted or improved.

The preventive approach implemented in ATM networks using CAC along with policing through UPC works very well and also guarantees QoS through allocating the required bandwidth. But, ATM is connection oriented and maintaining the bandwidth allocation for the duration of the session is relatively straightforward as all the cells have the same fixed length and take the same route thus enabling precise bandwidth allocation at each node along the path.

It became obvious that IP networks, which were developed for data communications are very well suited for delay-insensitive traffic but not for continuous media and further developments are necessary. At present, the RTP - UDP combination is the most widely used for continuous media but it does not provide guaranteed QoS. IPv6 shows some promise by introducing traffic flows but essentially, it provides a method of classifying traffic flows, which is only a part of the solution. The problem of allocating adequate resources for each class in a dynamically changing network is still to be overcome. RSVP together with some policing and shaping mechanism could provide the required QoS. However, being in a connectionless paradigm, there are inherent problems such as when packets go through non-RSVP compliant routers, which form the majority of

the network today. Furthermore, even for the RSVP compliant routers, it is not mandatory to allocate the requested bandwidth.

Future IP networks will have to offer multiple classes of service bandwidth partitioning instead of the current "one size fits all" type service where no distinction is made regarding the QoS requirements of the flows. This may be enabled using a Diffserv or similar architecture with support for multiple queues such as CBQ, and it may even incorporate some form of admission control. However, as network conditions change, the traffic class may be affected and there will be a need for responsive congestion control that instructs the flows to adapt according to the changes in the network. Such congestion control will have to be provided within each class of traffic. In such a scenario, a feedback based reactive control scheme would probably be most appropriate. An analysis of reactive congestion control for dynamic changes in the bandwidth is presented in the next chapter.

4

Reactive Congestion Control

In the previous chapter, we discussed the methods of controlling and avoiding congestion that are used in different networks. It was highlighted that current IP networks are not adequate for different types of services that will be demanded in future. Future IP networks will have to distinguish between the QoS requirements of different types of traffic and offer multiple classes of services through bandwidth partitioning. Using a Diffserv or similar architecture with queuing methods such as CBQ or WFQ, and possibly including some form of admission control in some classes may provide this. Nevertheless, in order to optimise the use of bandwidth, pre-defined allocation is not desirable particularly for the classes that carry bursty flows. In that case, it is likely that with dynamically changing network conditions, the problems of congestion will persist and it will be important to control or avoid congestion within such classes.

In this chapter, the special requirements of adaptive continuous media traffic are the prime concern. Traffic, such as video, which is bursty but requires low delay transmission, puts stringent demands on the network. It is often not optimal to allocate the peak rate and allocating mean rate may not be adequate. Instead, the bandwidth allocation must be more fluid. The changes in network conditions will affect this class, as a strict per-flow

bandwidth partitioning is not feasible. Adaptive traffic has the additional capability of changing the data rate dynamically. This can be used to alleviate the congestion in the network while ensuring that the throughput is acceptable. For this, a responsive feedback system is required so that the sources can be notified in time without incurring delay or loss of packets. We will look at different aspects of feedback based reactive congestion control methods in greater detail and discuss how they should behave for the treatment of continuous media traffic.

4.1 Structure of Reactive Congestion Control

Reactive Congestion Control techniques have been widely used for a very long time for computer data transmission. The underlying concept is simple. When a node experiences congestion, this information is conveyed back to the sources, which would then reduce the data rate. Conversely, when a node has a very light load on it the feedback mechanism will convey this to the sources as well which will then have the option of increasing their data rate[5]. There are three main components of the control scheme: feedback notification, adaptive action taken by the sources and most importantly detection of congestion, which would trigger the feedback.

4.1.1 Feedback Notification

The information that a node has become congested may be conveyed to the sources through implicit or explicit feedback. An implicit notification can be in form of a packet loss. In protocols such as TCP if a packet is lost or delayed, the acknowledgement ("ACK" packet) will not be received. In that case, the TCP source will reduce the size of its congestion window thereby reducing the amount of traffic transmitted. The TCP congestion window

[5] We have assumed that the flows use adaptive encoding or similar methods to change their data rate.

mechanism was also explained in Section 3.1.3. Further details on TCP congestion window can be found in [Jacobson 88, Floyd 93]. In a simple implementation, packet loss may happen when the buffer in the router, for example, overflows, such as in a Tail Drop method.

Alternatively, feedback signal can be sent explicitly from the network to the source in the forward direction via the destination (Forward Congestion Indication) or in the backward direction, directly back to the source (Backward Congestion Notification) [Marson 97]. As discussed in Chapter 3, FCI has been used in ATM networks and either FCI or BCN may be used in Frame Relay networks. With FCI, the destination, upon getting a packet with the congestion bit set, may choose whether to inform the source through an acknowledgement packet or ignore the congestion indication. It can also control the frequency of signalling. In case of BCN, sources are directly informed about the situation and they can take corrective action without further delay. BCN is therefore quicker in response although there are more overheads since the system is required to keep a record of the back path or maintain a dedicated control link from the routers to the sources.

In the relevant literature, suggestion for how much information should be sent as feedback has ranged from single bit [Ramakrishnan 90] to explicit rate notification [Kanakia 93] or information about the service rate and queue size at the nodes [Kanakia 96]. A single bit feedback is simple to implement but it is left to the sources to figure out what action to take. Explicit information obviously involves calculation of estimated rates, and hence more overheads, but instructs the sources as to what needs to be done. A scheme for transmission of VBR compressed video for interactive applications using explicit-rate feedback control has been suggested in [Lakshman 99]. The paper also proposes rate-allocation mechanisms in the network based on a weighted min-max fairness scheme so

that connections with different demands are affected proportionately by changes in the resources. Explicit Congestion Notification (ECN) has also been proposed for IP networks [Floyd 94, RFC 2481]. This would be in the forward direction with a congestion bit that would be set in the IP packet header.

4.1.2 Adaptive Mechanisms at the End Nodes

A source that receives a feedback may be rate aware and non-rate aware and this affects the consequences of the feedback notification. A TCP source, for example is not rate-aware. The TCP congestion window reduces in size when congestion is detected (see Section 3.1.3). This mechanism limits the amount of data that can be transmitted before the acknowledgements are received and does not correspond to a specific rate. On the other hand, the rate aware sources, such as an adaptive encoder, would adjust the current rate according to some pre-agreed policy or assign a specific rate for transmission.

Continuous media applications sources, by their very nature, are rate aware. Hence, they will have to change their data rate as network conditions change. This can be done by changes in the temporal rate or in the spatial resolution. As perceived quality is affected more by inconsistent temporal resolution, (Section 2.4), it is better to change the spatial resolution. This ensures that the frames still arrive at the same rate but they may be more or less noisy. Adaptive codecs, which are capable of changing their quantisation levels in real time, thereby changing the spatial resolution, are being developed. There will, of course, be a need for some form of policing mechanism, which ensures that the sources do adapt their rates according to the feedback they receive and adhere to their traffic parameters.

4.1.3 Monitoring and Detection of Congestion

The control system has to monitor the network against some suitable paradigms and make the decision of reporting congestion of possibility thereof. Most of the existing mechanisms operate by monitoring the average buffer occupancy and they trigger feedback signals when a set threshold value is exceeded. There is an inherent problem. If the monitored queue occupancy is fluctuating around the threshold, the system will also fluctuate between states of congestion and non-congestion, leading to oscillations in the response.

An improvement to a single threshold is the hysteresis algorithm in which two thresholds are used, high and low. A feedback signal is triggered if the queue occupancy crosses the higher threshold. Once started the signal is stopped only when the queue occupancy reduces to the lower threshold. Random Early Detection (RED) is another threshold based monitoring technique that has been shown to work very well with TCP traffic. The packets are dropped (or marked)[6] randomly once the queue occupancy exceeds the lower threshold. The probability that a packet will be dropped increases as the queue occupancy approaches the higher threshold and once this threshold is crossed, all the packets are marked. The idea is that sources start reducing their rates as their packets are dropped more often and hence avoid reaching the higher threshold [Floyd 93]. The randomness in marking means that the sources are notified at different times and hence it alleviates the problem of global synchronisation commonly seen in TCP with implicit feedback. Global synchronisation occurs in TCP during periods of congestion when implicit feedback of packet loss is used for congestion notification. When the buffer overflows and packets are lost, all the sources or flows simultaneously reduce their congestion window and hence the traffic. The reduction in traffic volume decreases the

[6] With RED, it is possible to mark the packet rather than dropping it and thus a feedback can be generated. However, it is not widely implemented at present.

congestion problem at which point they all increase their traffic, thus leading to severe oscillations in the amount of traffic traversing the system.

Although these techniques have been proved highly suitable for TCP data traffic, they have not been investigated for performance with continuous media. In a constantly changing network, it is crucial to detect the onset of congestion in time. It is noted that the adaptive source must be able to process the feedback signal it receives and take the appropriate action quickly. The execution time of this process is not a problem, as the processors chips are getting faster everyday. However, the speed of congestion detection or the capability of anticipating congestion depends on the algorithm in use. For example, an algorithm that gets progressively more aggressive such as RED is likely to be more responsive than one that waits for a threshold to be crossed. If congestion could be detected early or predicted, the system would be more responsive as it will be possible for sources to take corrective action quicker and alleviate the effects of congestion.

4.2 The Ideal Control Behaviour

In an ideal world, there would be a control system that can detect congestion before its onset and notify the sources in such a way that smooth graceful changes made by the co-operative sources would eliminate the congestion.

We assume that the sources are co-operative and therefore, they will adapt to the feedback sent by the congested node. The frequency of changes made by the source in the data rate is significant in the context of continuous media traffic because it has been found that rapid changes in, for example, video quality is more annoying to the end user than a consistent but lower quality reception. Ideally, we would like to have graceful changes in

the quality, particularly in the quantisation of spatial resolution of audio or video streams, Figure 4.1.

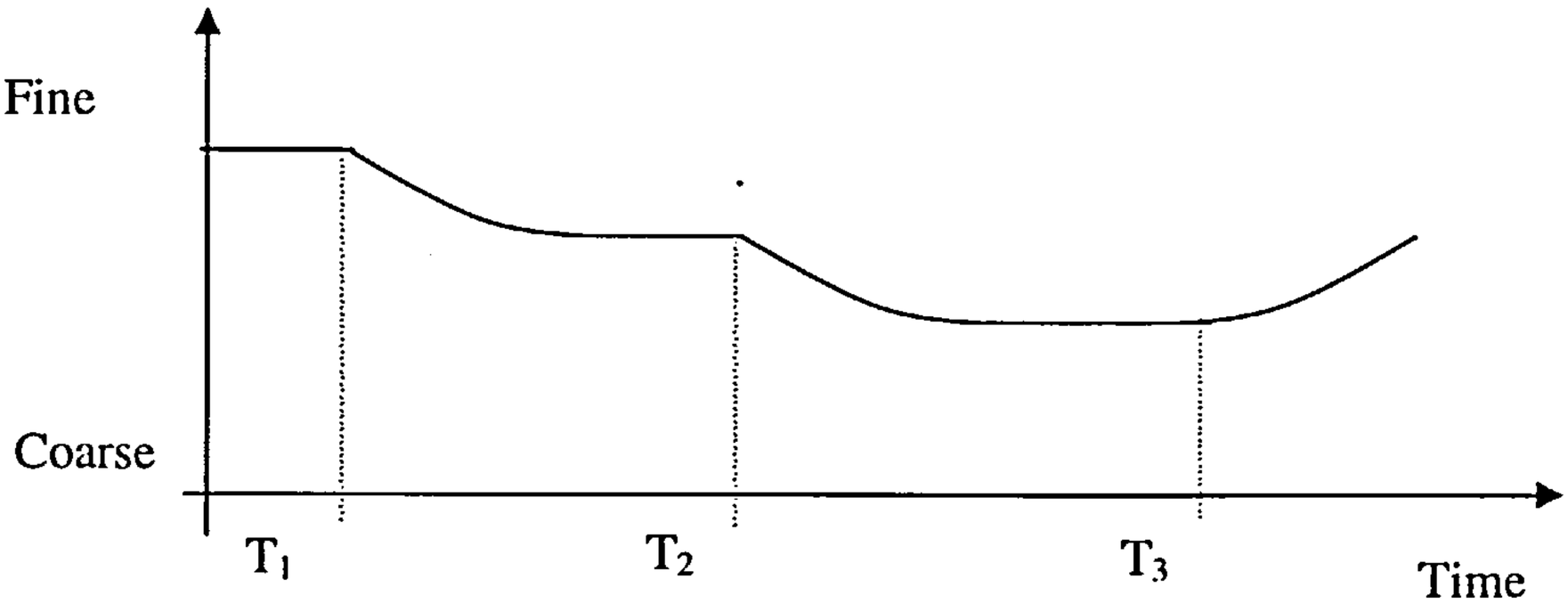

Figure 4.1: Graceful Changes with Changes in Network Condition

It is also important that the control system manages to regulate the traffic so that the majority, and not just the average, of packets experience a delay no greater than a given target. So, in an ideal control system, we would expect that a high percentile, e.g., 99%-ile, of the queue occupancy would remain below a target threshold, which maps on to the target delay.

4.3 Summary

In future, it is expected that IP networks will support a number of heterogeneous traffic classes through bandwidth partitioning and sophisticated queuing. However, it is believed that some control system will be required to minimise congestion and its consequences within a class. The simplest approach would be to develop a feedback based reactive congestion control scheme.

In this study, the focus is on adaptive continuous media traffic. This chapter presents an analysis of feedback based control methods with a view to developing a method suitable for adaptive traffic. We discussed that the amount and the type of feedback sent to

the source are significant; they govern how the sources would make the decision to adapt. The nature of adaptation has been assumed to be in spatial resolution through changes in quantisation levels. This is becoming possible in modern adaptive codecs and ensures that the frame rate stays consistent, as it is important for the perceived quality. We then identified perhaps the most significant component of the control system, which is congestion detection. The congestion detection process has to be fast and accurate in order to trigger a timely feedback signal. We introduced a couple of methods for ascertaining the level of congestion and generating feedback that have been successful with data traffic.

Finally, the chapter ends with a discussion of the ideal behaving control scheme. The ideal behaviour is based upon the study into perceived quality which explains that gradual changes are less offending to the users and sustained low quality is perceived to be better than frequent fluctuations between high and low resolutions.

During the discussion of congestion detection, some methods, which have been shown to work well with data traffic, were introduced. The author proposes that this is an area which should be investigated further. In particular, we need to see how Hysteresis and RED perform with continuous media traffic requirements. We will carry out the evaluation through simulating adapted versions of Hysteresis and RED. The details of simulations and the results will be presented in Chapter 6. Before that, the network model and aspects of simulations are presented in the following chapter.

5

Network Scenario and Modelling

So far, we have discussed various aspects of QoS and congestion control both in IP and other networks. We have identified the need for a dynamic bandwidth control in the context of IP networks so that flows can adapt to the changing network conditions. In the last chapter, we analysed the components of such a controller based on explicit feedback signalling. One of the most important objectives of any feedback based control for continuous media is the responsiveness of the algorithm, which is significantly dependent on the process of detecting congestion and making the decision to send feedback. With this, we move on to the experimental phase of the research.

In this chapter, the network scenario that was modelled in the OPNET™ Modeler environment for evaluating the performance of control algorithms is presented. The discussion includes the important design issues that were used and developed along with a justification for the choices made.

5.1 Network Scenario and Simplifications

The network scenario used for experiments was chosen to be as simple as possible so that the effect of the control algorithm could be clearly evaluated with minimum possibility of

the results being affected by other processes and parameters. The model is a simplified version of a situation that may occur between a number of users all of whom use one ISP (Internet Service Provider). The ISP router would accept different types of traffic from all the users and this would be capable of multi-service, that is, it would classify the incoming traffic into various classes according to their QoS and attempt to avoid congestion within the guaranteed and adaptive real time classes. Usually the ISP router will forward the traffic to one or more routers depending upon the configuration but for now we will assume that the backbone network routers do not induce any delay and that the congestion is likely only at the output of the ISP's router. Furthermore, since we use different classes of traffic, we assume that each class will be isolated from the others. For example, congestion in the adaptive class will not induce delay in the guaranteed service class. The underlying mechanism that provides the bandwidth partitioning and multi-service could be Diffserv, Intserv or any other, which is not relevant to this study.

Our focus is on the adaptive real time service class so we simplify the model for simulation as shown in Figure 5.1 and only include adaptive video traffic class in our model, thereby eliminating the need for traffic classification.

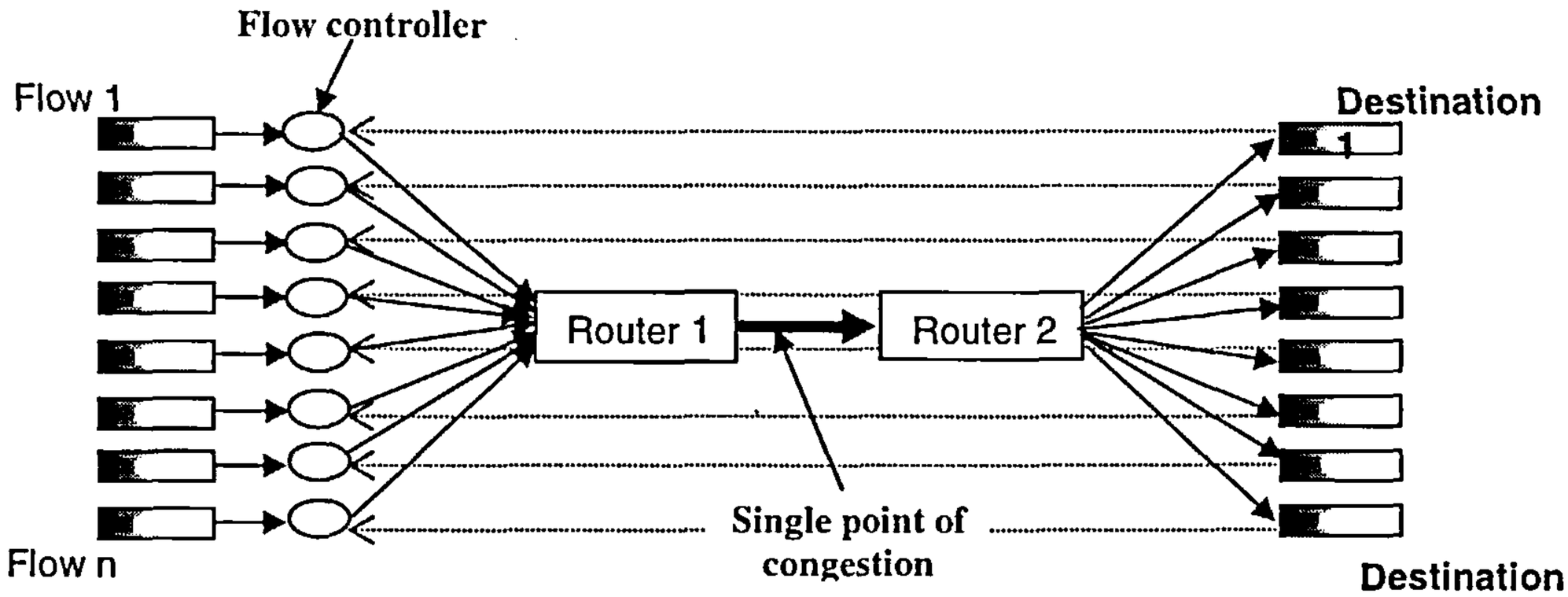

Figure 5.1: Network Scenario for Feedback Based Control Systems

The network topology consists of a number of flows that generate video traffic. The Flow Controller process carries out the policing of this traffic before sending it to Router 1. Router 1 implements the ISP router in the network scenario described above, while Router 2 models the delay-free router that receives traffic from the ISP router. The point of congestion in this network model is, therefore, at the output queue of Router 1 and this is where the congestion detection part of the algorithm takes place. The simplified logical diagram of the routers is shown in Figure 5.2.

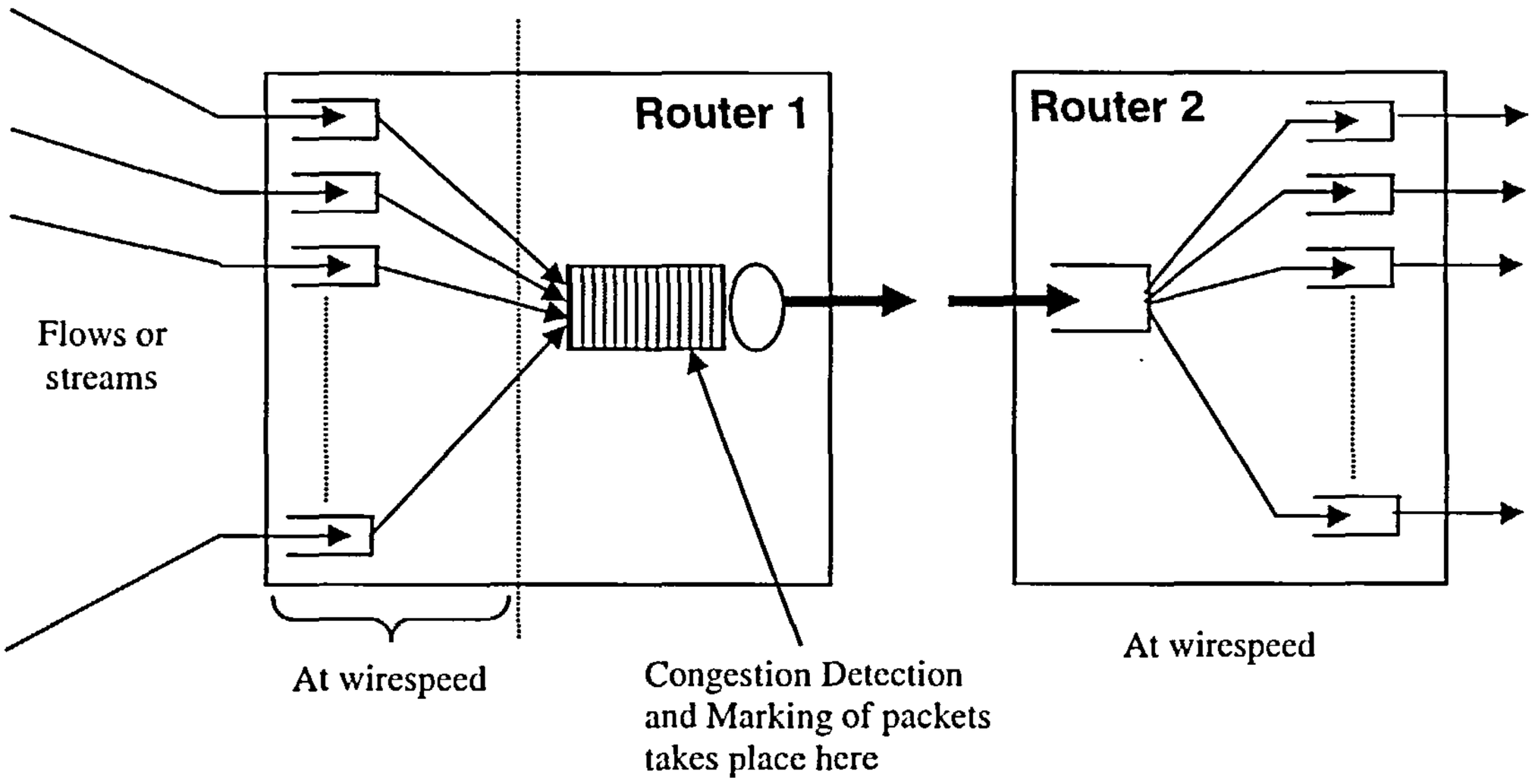

Figure 5.2: Simplified Logical Diagram of Routers 1 and 2

For simulation purposes, they can be further simplified and modelled as a single bottleneck node which performs congestion detection and stamps the packet and then carries out a look up routing that would be the function of Router 2 as shown in Figure 5.3.

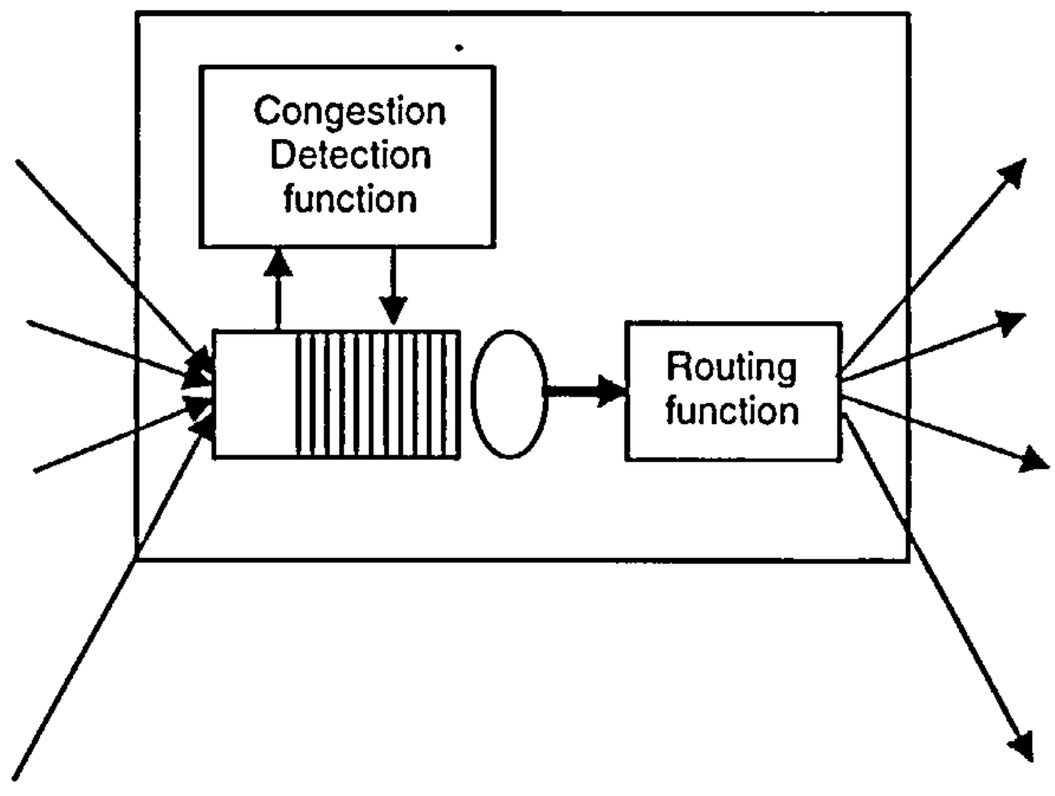

Figure 5.3: Bottleneck Node Model

The network model above, see Figure 5.1, shows a direct path between the destination and sources or flows. For simplicity, this path was modelled with no delay or queuing. In reality, the feedback packets in an FCI system like this will encounter some queuing themselves. It is also important to note that no delay other than queuing delay was modelled. This is because the propagation delay due to the physical distance between any two nodes in the network is not affected by the changes in the network conditions, as long as the routing does not change significantly. Including this delay would add to the complexity of routing behaviours and distract us from our purpose of evaluating the effects on queuing delays and congestion.

In the following subsections, the components of the network model are briefly described. More detailed descriptions can be found in Appendix II.

5.1.1 Packet Generator

The packet generator process obtains the packet lengths from an MPEG trace file, see Appendix I, and uses the specified attributes such as average inter-arrival time and squared coefficient of variance in the inter-arrival time to generate and send packets. In all the experiments, the behaviour of the Packet Generator is kept the same.

This process generates packets and transmits them while it is active, i.e., between a given start and stop time. It reads in the packet lengths for successive packets from the specified trace file. It then uses the generalised exponential function to derive the delay before the next packet can be sent using the algorithm shown in Figure 5.4.

```
if(u ≤ 1- τ)
        delay = 0

else
                    log((1−u)/τ)
        delay = ----------------
                        −σ
```

Figure 5.4: Generalised Exponential Algorithm

where,

$$\tau = \frac{2}{squared\ coefficient\ of\ variance + 1} \qquad \textbf{Eq 5.1}$$

$$\sigma = \frac{\tau}{average\ interarrival\ time} \qquad \textbf{Eq 5.2}$$

and $0.0 \leq u < 1$

This algorithm is based on the Poisson process with exponential distributions [Gross 98] and has been shown to work well for the values of squared coefficient of variance ≥ 1. With the squared coefficient of variance set at 1, this algorithm approximates a Poisson process. If the squared coefficient of variance was larger than 1, then τ would be ≤ 1 and hence the probability of delay between successive packets to be zero will be higher. This would then model bulk arrivals [Ball 96a].

5.1.2 Flow Controller

The function of the flow controller process is to receive the congestion information from the destination node and to perform policing on the flow from the source or packet generators. If a negative feedback is received, the data rate has to be reduced. This is executed by reducing the packet length because this will change the resolution whereas reducing temporal rate would induce jitter and therefore cause flickering images in the case of video. Reducing the packet lengths is possible using adaptive encoding techniques. Video coding such as MPEG uses a sophisticated method of decomposing a picture into blocks and refining each block further. The degree of refinement, and hence the number of bits required to carry the information, depends upon the quantisation level used. In the literature, adaptive encoders have been used to change the quantisation in order to adapt the frame length [Casetti 96, Ramanujan 97, Feng 98]. The usual consensus is that packets smaller than half the original length do not contain enough information for acceptable transmission quality.

Sophisticated encoding techniques are outside the scope of this study. Therefore, a simplified method using a stepped change is implemented. This is modelled in the simulation by decreasing the packet size by a factor depending on the number of successive feedbacks. This factor has been termed as the Packet Length Reduction Factor *(f)* and

ranges from 0.5 to 1.0 changing in steps of 0.1. The controller will multiply the packet length of each packet from the source by this factor before forwarding it on. The range of f ensures that the resulting packet length will be half the original size at minimum and equal to original size at maximum. The default value is 1.0.

Packet truncation is a simple method of changing the data rate but since the objective of this work was to evaluate the control process, it is adequate to use this. Of course, in future, more sophisticated adaptive encoding techniques can be used. It is worth noting here that truncating packets rather than using smoothing buffers reduces the spatial resolution of the signal while maintaining the temporal resolution. As we have discussed before, this is important because from the user's point of view it is easier to tolerate, say, degradation in the video image (due to reduction in spatial resolution) than a flicker in the images (due to reduction in temporal resolution).

The controller process has essentially the same behaviour in all the experiments but the method of implementation changes for each algorithm. For example in Hysteresis and RED, the lack of negative feedback for a certain duration of time is inferred as a positive feedback so that the data rate can be incremented back to full while in the methods with explicit feedback for both positive and negative case the implementation will be slightly different.

5.1.3 Bottleneck Link

The service rate of the queue, which modelled the bottleneck node, was 12Mbits/s and the higher and lower thresholds were 450000 bits and 360000 bits respectively. The threshold values were chosen as they related to queuing delays of 37.5 ms and 30ms respectively for the given service rate. The typical end-to-end delay that can be tolerated for video applications is in the region of 150 – 200ms but it includes propagation delay due to the

physical location of the nodes, which cannot be reduced. Hence the queuing delay at a node must be a small fraction (here less than $1/5^{th}$) of the total delay. The important parameter is the difference between the upper and lower limit of the target delay as this would control the jitter in the system. Typically, video applications can tolerate a jitter of 10ms, and here the difference was 7.5ms. In simulation, the buffer capacity was infinitely long but in reality, the buffers would be finite and perhaps not much longer than 50ms as the aim is to minimise delays encountered in queuing.

Further discussion on the queue behaviour is related to the method of congestion detection and hence is specific to the algorithm in use. We will therefore return to it while discussing the algorithms in the next chapter. However, there is an important aspect that was kept consistent and hence is worth discussing here.

In every experiment, the packets at the head of the queue were marked with congestion information. If the packets are marked as they arrive, it would take at least the service time of all the packets present in the queue before the congestion notification is forwarded to all the destinations. Although marking the packet at the head of the queue improves responsiveness, unfairness is possible. For example, if there are two sources in the queue and a third source joins in causing congestion, the packets from the first two will be marked until the first packet from the third source gets to the head of the queue. Consequently, the first two sources could be penalised even if they were behaving well because of a greedy third source. However, in the event of packets being marked as they come in, we can visualise a scenario where two sources are loading the queue and packets from a third source, even if it has a low data rate, are marked. So, it is fair to mark the packets at the head of the queue as this method is more responsive and eventually packets from all the sources will be marked if the problem persists.

5.1.4 Destination

The destination process is very simple but the behaviour changes according to the algorithm in use. This is because the FCI scheme has been used and the destination, therefore, has to act upon the congestion notification. In all cases, the process receives the packets from the sources coming via the queue and looks up the congestion status before destroying it. However, the congestion information is dealt with differently in each case. In Hysteresis, for example, a periodic feedback is sent to the controller process.

5.2 Performance Attributes

The ideal behaviour of a reactive control system for continuous media traffic was discussed in Section 4.2. Each of the algorithms will be tested to see how closely they behave compared to the ideal. The performance is evaluated in terms of the following aspects:

- The frequency in the changes that the sources would have to exercise
- The high percentile queue occupancy

5.2.1 Frequency in Changes at the Source

The sources are assumed co-operative and hence they will adapt to the load in the network according to the feedback signals they receive. However, it is important that the changes are graceful (see Section 4.2) or at least not rapid oscillations. The simulation model was designed such that this behaviour can be easily observed.

5.2.2 99-Percentile Queue Occupancy

The queue occupancy has to be monitored carefully to ensure that the losses are minimized. Our aim was to control the queue so that a high percentile, say 99-percentile, of the packets would not experience more than a certain delay. This can be monitored by working out a

threshold "queue occupancy for the given delay" (from delay-bandwidth product) and ensuring that 99-percentile of the queue occupancy (Q_{99}), remained below the desired threshold. As we know, it is important to bound the jitter of continuous media packets. A fairly simple method would be to ensure that a high percentile of packets suffer a delay not too much longer than the target delay. Hence we measure the 99-percentile queue occupancy and compare it against the threshold as well as with that observed in other methods.

In the following subsections we describe the model specifications including the specific features of Hysteresis and RED. In further experiments the common features will be kept consistent as far as possible.

5.3 Scenario Details

As has been mentioned before, the model consisted of one source that generated background traffic and all the other sources were switched on or off to create different loading patterns on the bottleneck link. The two scenarios that were of particular interest are shown in Table 5.1.

Table 5.1: Simulation Scenarios

Scenario Name	Time (s)	Number of Sources active	Approximate Load (assuming no control)	Queue Service Rate (Mbit/s)
	0 – 200	1	8	
Gradual	200 – 300	2	10	12
change	300 – 350	3	12	
	350 – 500	4	14	
	500 – 700	3	12	
	700 – 800	2	10	
	800 – 1000	1	8	
	0 – 300	1	8	
Sudden	300 – 500	4	14	12
Change	500 – 700	3	12	
	700 – 800	2	10	
	800 - 1000	1	8	

In the "gradual change" scenario, sources become active and inactive one at a time. The simulation starts with the background source and the 1st source active, which remain active for the entire duration of simulation. The 2nd source becomes active at 200s, the 3rd source at 300s, and the 4th source at 350s. The sources then start going off with the 4th source stopping at 500s, the 3rd source stopping at 700, and the 2nd source stopping at 800s. This causes gradual increase and decrease on the load at the bottleneck link. The approximate load column indicates the estimated rate of incoming traffic with the corresponding number of sources active, if no control was in place. The scenario includes the situations where incoming rate exceeds the service rate of the queue. Of course, with the control algorithm in action, it is expected that the system will recover from such high utilisation. As we shall see later, the recovery patterns vary in each case.

In the "sudden change" scenario, there is only the background source and the 1st source active during the first 300 seconds and then 3 sources join in simultaneously at 300s, causing a sudden increase in the load. The sources then stop one at a time at 500, 700 and

800s, leaving the background and 1 source to continue until the end of the simulation. We will be using these two scenarios in particular to evaluate the performance of the control methods changing load conditions.

5.4 Summary

The evaluation of a control algorithm has to be carried out through simulation mainly because a test on real networks will be extremely complicated and will require immense resources. Therefore, the powerful modelling environment of OPNET Modeler™ is used to construct the network model and implement algorithms. The model is derived from a real life situation but has been simplified so that it is possible to test the performance of congestion control schemes without the artefacts of the scenario complexity.

This chapter introduced the network scenario that was modelled for the experiments of feedback based control systems. An example of a situation that may occur commonly in real networks was presented and then the model was derived from it through simplifications and assumptions as stated along with justifications for the choices made. The constituents of the model, viz. packet generator, flow controller, bottleneck link and destination, were each discussed in brief with detailed specifications given in Appendix II.

The discussion of performance attributes and loading patterns that were considered most important is also presented. Building upon the concepts relevant to continuous media traffic throughout this thesis, the author has presented two main attributes that need to be observed. They are the frequency of changes required to be made by the source and 99-percentile queue occupancy at the bottleneck. These attributes have been chosen for their pertinence to continuous media traffic as discussed in Section 4.2. The loading patterns that

would be used in the simulation are designed to illustrate the control response during gradual and sudden increase in the load at the bottleneck.

In the next chapter, we shall use this model for testing the performance of Hysteresis and RED and present the results.

6

Hysteresis and RED

The network scenario that was used for modelling and testing the control algorithm behaviour has been described in detail in the last chapter. Hysteresis and RED are two of the most common and successful congestion avoidance and control schemes used in data traffic. In this chapter, we discuss the investigations carried out using Hysteresis and RED and present the results.

In the last chapter the implementation details for the queue behaviour at the bottleneck link were not included as they are specific to the algorithm. Therefore, we start this chapter with a description of the bottleneck link behaviour and congestion detection process for Hysteresis and RED before moving on to the results.

6.1 Modelling Hysteresis

When a packet is received at the bottleneck, its length is added to the queue size and a first order filter function is carried out to get the average queue occupancy. The filter equation is shown in Eq 6.1.

$$Q_{avg} = (1-\alpha)Q_{avg}' + \alpha Q \qquad \textbf{Eq 6.1}$$

where,

Q_{avg}' = last calculated value of the average queue occupancy and

α = filter coefficient specified at the simulation time

The average obtained is then compared against the specified thresholds. When the Q_{avg} exceeds the higher threshold, the node is said to be congested and the packet that is being served is marked (the congestion status field of the packet would be set to 1). All the successive packets continue to be marked as they leave the service until the Q_{avg} goes below the lower threshold. After this the congestion status field in following packets will be set to 0.

When the destination process receives a packet with congestion status set to 1, it generates a feedback packet with the congestion status copied to its header and sends it to the flow controller via the return path. Thereafter, it sends a feedback packet periodically every second until it receives a data packet with congestion status 0. As described earlier in Section 5.1.2, whenever the flow controller receives a feedback packet with congestion status 1, it decreases the value of Packet Length Reduction Factor f that is used to change the length of packets being sent to the queue. When it receives no feedback for a specified duration, it would increment the value of f (see Appendix II).

6.2 Modelling RED

As in Hysteresis, the average queue occupancy is calculated as shown in Eq 6.1 and the Q_{avg} is then compared against the thresholds. This time, however, when the Q_{avg} exceeds the lower threshold, the node will start marking the packets leaving the queue at random. The probability of marking the packets increases (up to the maximum specified) as the queue occupancy increases and when the queue occupancy exceeds the higher threshold, all the

packets are marked. When the queue occupancy starts to decrease the probability of marking decreases proportionally. This is the RED gateway algorithm given in [Floyd 93]. It must be noted that the RED algorithm allows the packets to be dropped instead of being marked, thus making the control more aggressive. However, as it is undesirable for continuous media applications to have packet losses, we have adapted the algorithm to mark the packets with congestion status information in the header.

The algorithm specifies that if the queue has been idle for some time and then a packet arrives at time *t*, the average should be calculated with an assumption that a number of small packets had arrived during the time when the queue was idle. The number of small packets, *m*, is estimated by Eq. 6.2 and the average is then calculated using Eq 6.3.

$$m = \frac{(t - q\_time)}{s} \qquad \textbf{Eq 6.2}$$

$$Q_{avg} = (1-\alpha)^m * Q_{avg} \qquad \textbf{Eq 6.3}$$

where,

m = estimated number of packets

t = current time

q_time = time when the queue became idle

s = service time for a typically small packet

The probability of marking depends on the queue occupancy and is calculated from the following equations, Eqs. 6.4 and 6.5.

$$p_b = \frac{\max_p (Q_{avg} - \min_{th})}{\max_{th} - \min_{th}} \qquad \textbf{Eq 6.4}$$

$$p_a = \frac{p_b}{(1 - count \bullet p_b)} \qquad \textbf{Eq 6.5}$$

where,

p_b = initial probability of marking

p_a = final probability of marking

max_{th} = maximum threshold (or higher threshold)

min_{th} = minimum threshold (or lower threshold)

max_p = maximum probability specified

count = counter to increase the probability

Initialisation:

Q_{avg} = 0

count = -1

for each packet arrival

Calculate Q_{avg} from Eq 6.1 or Eq 6.3

if $min_{th} \leq Q_{avg} \leq max_{th}$

increment *count*

calculate p_a from Eq 6.5

with probability p_a

mark the packet

set *count* to 0

else if $max_{th} < Q_{avg}$

Figure 6.1: Algorithm for RED [Floyd 93]

In the case of RED, the destination differs from the one in hysteresis as it sends a feedback whenever a marked packet is received in order to preserve the randomness of congestion notification. The controller process remains the same in both (see Appendix II).

6.3 Results

The results were collated with a view to investigate the system performance in terms of frequency of changes imposed on the source monitors and 99-percentile of the queue occupancy (Q_{99}), as described in Section 5.2.

The frequency of changes made by the source is shown by the fluctuations in the value of *f*. This is a qualitative result and presented in the graphs below for comparison between Hysteresis and RED.

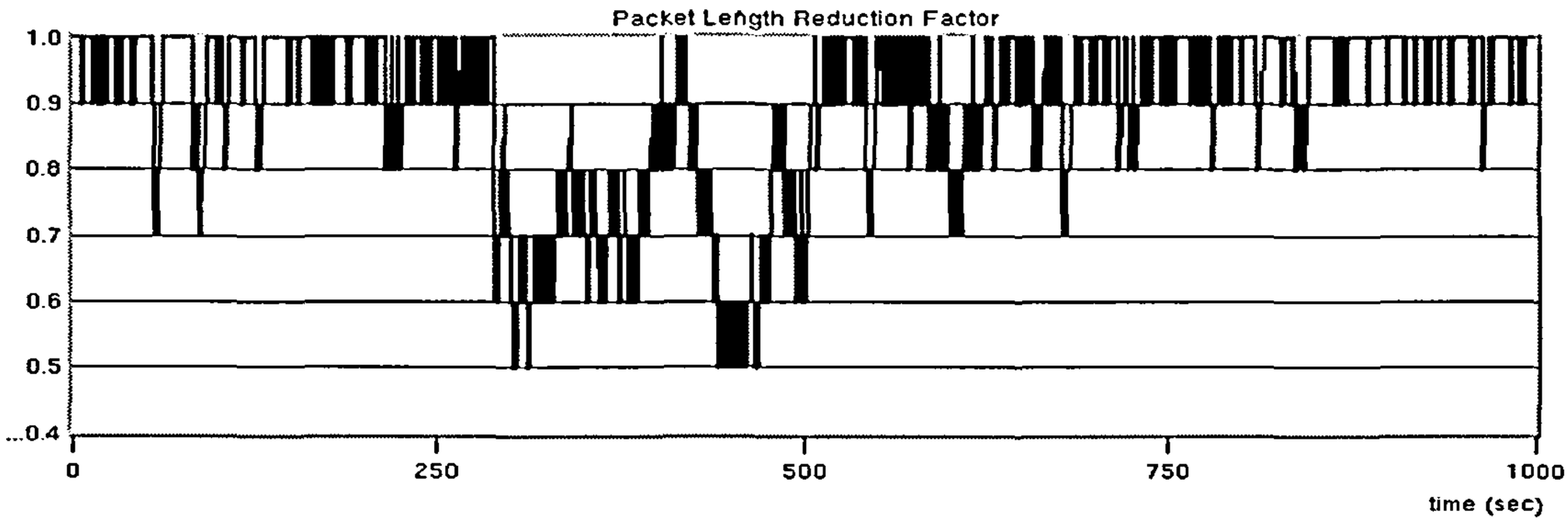

Figure 6.2: Fluctuations in *f* with Hysteresis (gradual change)

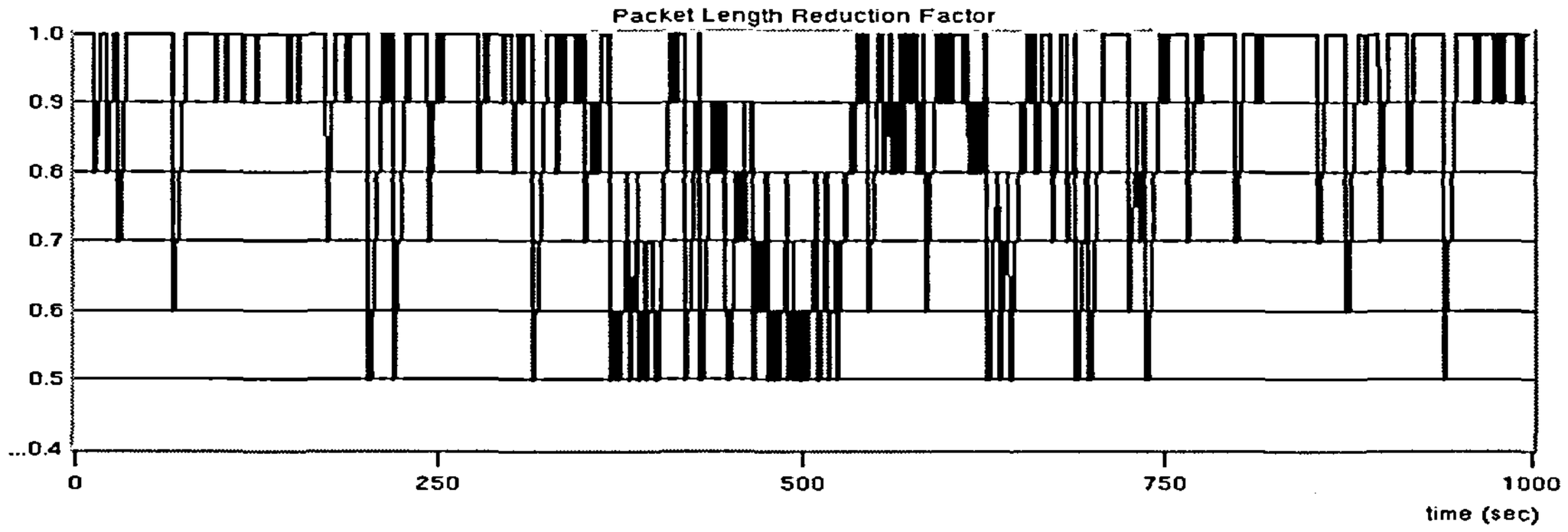

Figure 6.3: Fluctuations in *f* with RED (gradual change)

The graphs show that the value of *f* fluctuates very rapidly in both cases and the case of RED is worse than that of hysteresis. This is possibly because in hysteresis the negative feedback was being sent periodically whereas with RED the frequency of the feedback is determined by the function of probability. During congestion, this approach would generate proportionately more feedbacks. The traces for sudden changes are shown

in Figure 6.4 and Figure 6.5. As expected the fluctuations are severe around 300s when the load on the link suddenly increases.

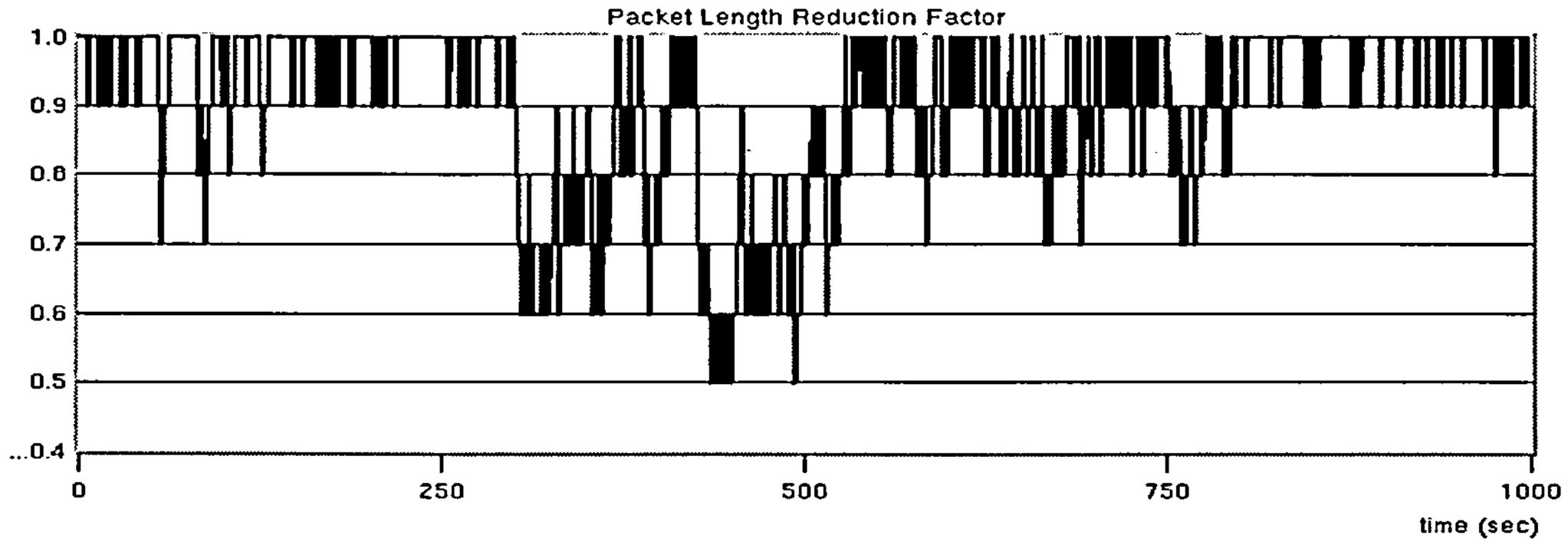

Figure 6.4: Fluctuations in *f* with Hysteresis (sudden change)

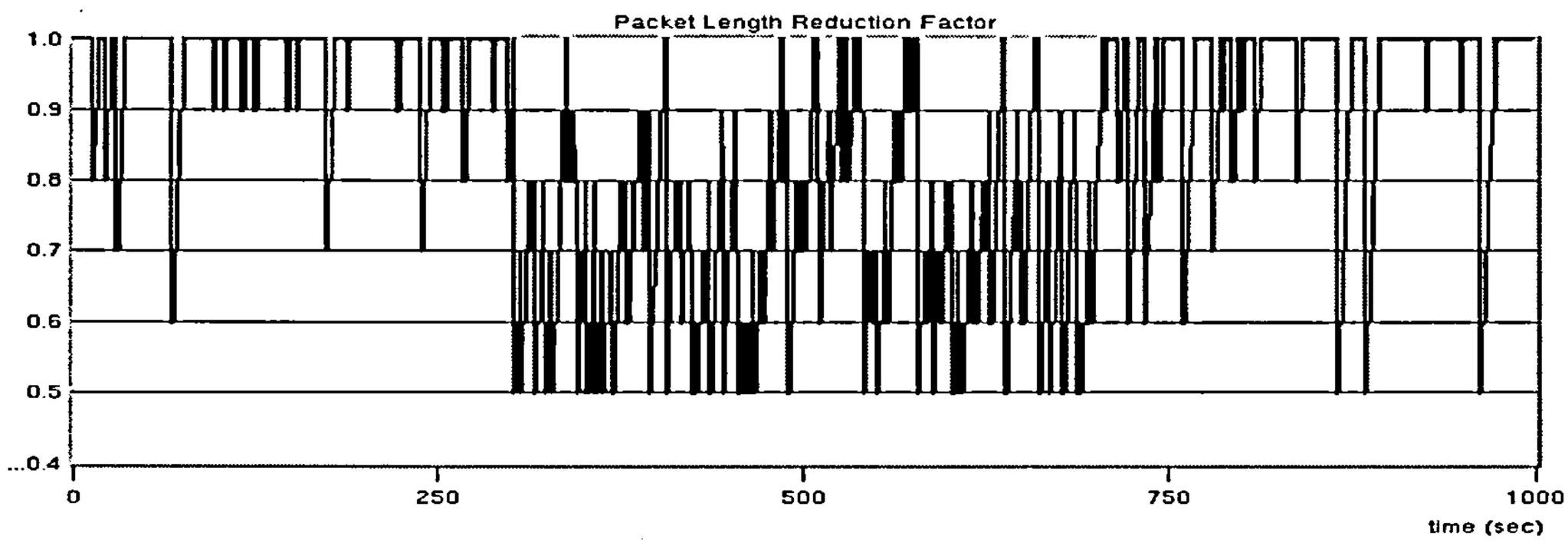

Figure 6.5: Fluctuations in *f* with RED (sudden change)

The 99-percentile queue occupancy *(Q_{99})* was measured for both gradual and sudden load patterns in each case; the values are given in Table 6.1. This measurement indicates the upper bound of the delay experienced by the majority of the packets. The simulations were repeated with different traffic patterns obtained by changing the seed value used to generate the random delay between successive packets. The table shows the average of Q_{99} in 12 runs. This is used to derive the 95% confidence region using Student's t distribution for small samples [Wonnacott 90]. The results are presented as a range of values, high and low, which are average plus confidence interval and average minus confidence interval respectively. For example, for the case of gradual changes in the load,

we are 95% confident that the 99-percentile queue occupancy was between 484126 and 503265 bits when Hysteresis was used as the controlling algorithm.

Table 6.1: Q_{99} (bits) results for with Hysteresis and RED

Algorithm	Gradual changes in load			Sudden changes in load		
	Average	High	Low	Average	High	Low
Hysteresis	493696	503265	484126	513803	528823	498782
RED	453597	458217	448977	451718	455468	447967

In all the cases of Hysteresis, the 99-percentile queue occupancy is entirely above the higher threshold mark (450000 bits). With RED, the average and the upper bound of Q_{99} exceeds the higher threshold mark while the lower bound is also quite near the higher threshold. Hence, it is safe to conclude that with both methods, the majority of packets would have encountered a delay longer than 37.5ms[7].

6.4 Proposed Improvements

It is quite clear that neither of these approaches is appropriate to maintain the control on the queues for interactive real time services. One of the main reasons is that these methods are based on monitoring the average queue occupancy against some threshold to ascertain congestion and to generate feedback. With real time traffic, it is important that the majority, and not simply the average number, of the packets do not suffer from excessive delays or loss. As shown in Figure 6.6, two completely different delay distributions may have the same average value. By bounding the average then, we cannot control the deviation in the delay and there may be a significant number of packets, which experience too long or too short delay and hence suffer from jitter.

[7] Queue length of 450000 bits corresponds to 37.5ms delay when the service rate is 12 Mbit/s

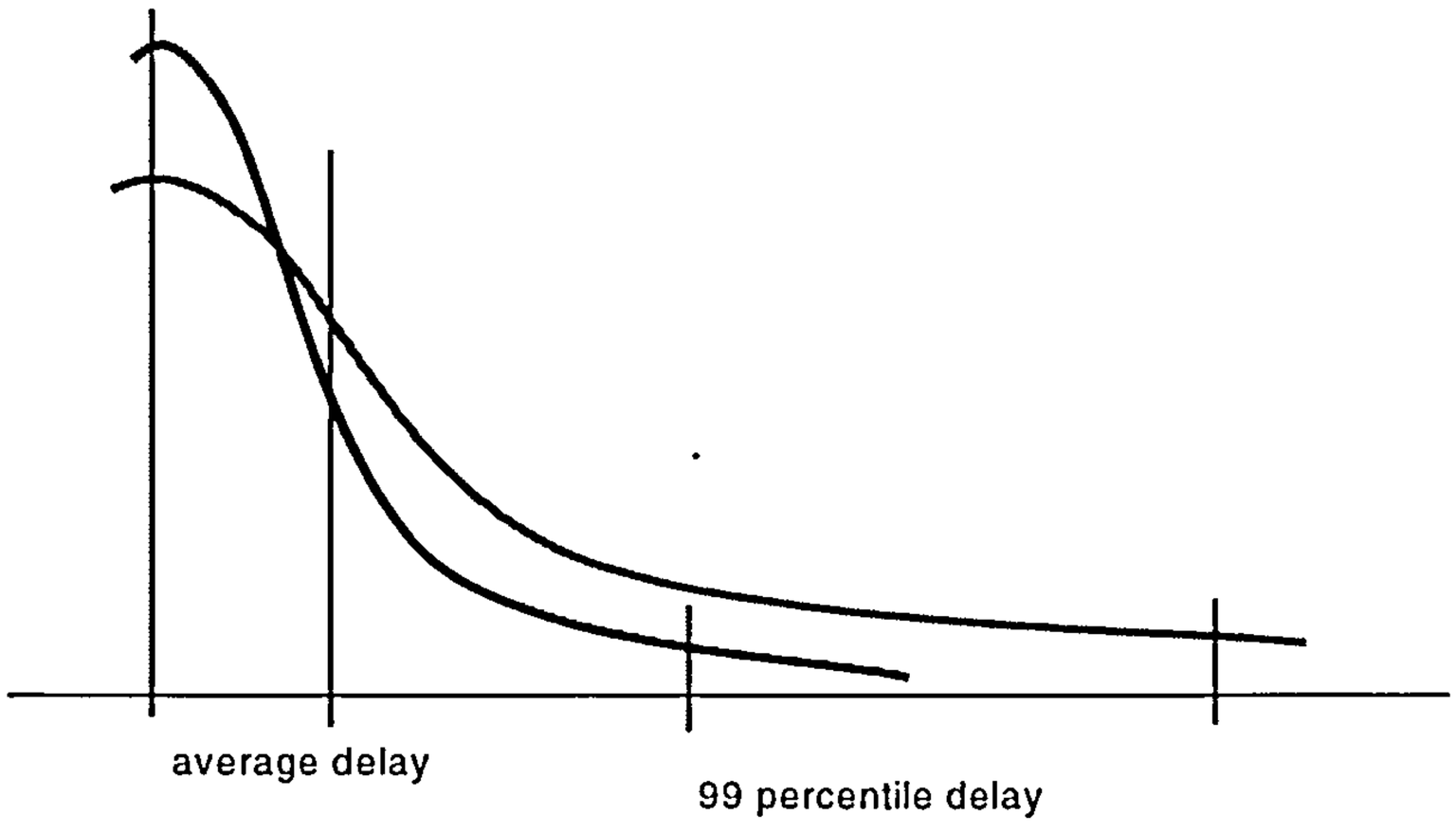

Figure 6.6: Different Delay Distributions with Same Average Value

It may not be possible to achieve the smooth and graceful degradation shown in the ideal behaviour, Figure 4.1, because all feedback systems have an inherent delay, which will always cause fluctuations, but it is important to develop methods of minimising them for controlling the queue occupancy.

6.5 High Percentile Monitoring

This is a new technique, which would enable the system to be controlled for the target delay. This type of control system will generate feedbacks and act upon them such that a high percentile of the packets experience a delay less than or equal to the target delay. If more packets start to experience delay longer than or equal to the target delay, the sources will be sent an instruction to reduce their data rate. Conversely, if fewer than the acceptable percentile of packets suffer the target delay, then the sources will be given an option to increase their data rate. This approach means that instead of trying to control the delay the emphasis is now on regulating the number of packets that experience the given delay. This is sensible because with a packet, switched network delay is inherent and there is no way of eliminating it. By monitoring high percentile queue occupancy it should be possible to

reduce jitter. The detailed model description and experimental results are presented in Chapter 7.

6.6 Summary

In this chapter, we have analysed the behaviour of some well-established feedback control systems and observed their performance in the context of continuous media traffic. The chapter presents the details of simulation model constructed to test the performance of Hysteresis and RED and to evaluate the effect on perceived quality. Through our experiments in the simulated environment, we saw that our speculations made in Chapter 4 have been found to be true as the performance of the methods tested here is far from ideal (Figure 4.1). The results show that the changes in the flow parameters (signified by fluctuations in *f)* are large and frequent. Also, the 99-percentile of queue occupancy is found to be above the threshold, which relates to an acceptable packet delay. The high percentile of queue occupancy is also higher than the thresholds. Although the queues used in the simulations were infinite in length, in reality, the buffers would be of finite length and this could lead to losses due to overflow. Long buffers would reduce losses but add to the delays.

Moving on, the author suggests that it may be more appropriate to use a feedback system that measures a high percentile of queue occupancy instead of the mean. In the next chapter, this novel control scheme is described along with the details of the development process and the results obtained through similar tests as those carried out for Hysteresis and RED.

7

Monitoring a High Percentile

The concept for monitoring a high percentile was envisaged in order to select an appropriate target delay and to monitor and control the queue occupancy so as not to allow a delay greater than the target delay. The queuing delay is a function of link rate (which was kept constant) and the queue occupancy (which varied according to the incoming traffic). Hence, the target delay could be translated to a threshold in the queue. When 99-percentile of queue occupancy (Q_{99}) appeared to be higher than the threshold, i.e. when more than the desired number of packets were experiencing a delay greater than the target delay, a negative feedback was sent to the sources, forcing them to reduce their bit rates. When Q_{99} appeared to be lower than the threshold, a positive feedback was sent, giving the sources an option to increase their data rates. While this is the basic concept, several refinements had to be made to the algorithm. These are discussed in the following sections. As in the previous experiments, the fluctuation in f and the value of Q_{99} were monitored.

7.1 Implementing 99-Percentile Queue Occupancy Monitoring

In this method, we have a tolerable target delay (D_q). The objective is to ensure that D_q represents 99-percentile of the queue occupancy, i.e. 99% of the packets experience a delay

no more than D_q or, in other words, 10 packets in 1000 experience a delay greater than D_q. Consider that D_q is 50ms. If D_q appears to represent a lower percentile, say 98.8, so that 12 in 1000 experience a delay greater than 50ms then the source output has to be decreased. On the other hand, if D_q represents a higher percentile, e.g. 99.2, when only 8 packets in 1000 are delayed by longer than 50 ms, then there is an option to increase the source output. Since the queuing delay is linked to the queue occupancy, D_q can be mapped to a corresponding 99-percentile queue occupancy (Q_{99}). Of course, it is a complex procedure to calculate the Q_{99} so a sampling method was implemented.

7.2 Estimating Q_{99} through Sampling Method

Monitoring a high percentile although desirable is not possible to do in true real time. Instead, some approximation has to be made using statistical methods. Consider a variable value being sampled and checked against a threshold. If in a set of 2000 samples, there were only 20 instances when the value was above the threshold, then this implies that the threshold coincides with the 99-percentile mark of the data. However, with all statistical approaches there are errors and also a degree of confidence with the results. For example, in this case we can be 95% sure that a value, **x**, is 99-percentile mark of a data of 2000 samples if the number of samples that exceed **x**, lies between 14 and 28. Further, we can assume that 99-percentile of the data is not greater than **x** if no more than 14 samples exceed the value. This has been termed as conservative here because it gives us a more strict measure. A liberal approach would allow up to 28 samples to exceed the threshold value. Table 7.1 presents the 95% confidence regions for set of 2000, 3000, 4000 and 5000 samples. The columns present the actual 99-percentile, conservative, and liberal values for a given sample.

Table 7.1: Range of Acceptable Values

Sample Size	Actual	Conservative	Liberal
2000	20	14	28
3000	30	23	39
4000	40	32	50
5000	50	41	61

We can now apply the above method for congestion detection. We sample the queue size every time a packet leaves the node and collect a set of 2000 samples. Using the network scenarios described in Section 5.3, the link service rate is 12 Mbit/s link and average packet length is 80000 bits. Unless the queue is emptied (which implies little congestion), the sample set of 2000 is equivalent to 13.33s, which is a reasonably short period for congestion monitoring. Furthermore, as will be explained later in this chapter, a congestion notification is made as soon as the congestion is detected without waiting for the end of sample set and therefore during persistent congestion, the feedbacks are a lot more frequent than every 13 seconds.

So, we record the queue size each time a packet leaves the node and also record the number of times the queue size exceeds a certain threshold, which is related to the target delay. If the number of times the queue size exceeds the threshold is greater than 14 in a set of 2000 samples, congestion will be reported. The algorithm also includes measures for reporting that the congestion has cleared and is designed to ensure that the delay encountered by the 99-percentile of the packets stays within limits of the confidence region. The complete algorithm will be developed and described in the following sections. It must be noted that the method can be adapted for use with higher numbers of samples in the set, say 4000, which may be more applicable for some applications. This should be investigated in further work on this method.

The experiments were first carried out using the conservative set of values to ensure that the system performed well under stringent conditions and then they were repeated with the liberal values. The number of samples observed is termed as *sample_size* while the number of samples that are allowed to exceed the threshold in a *sample_size* is called *exceed_limit.*

7.3 Algorithm Design Improvements

The algorithm was simulated using the OPNET™ environment. The initial results highlighted some design issues that were subsequently improved through a process of continual change and testing. The following subsections describe the main aspects of the algorithm that were addressed.

7.3.1 3-stage Congestion Notification

Initially, the algorithm was designed with one threshold with a view to ensuring that the 99-percentile of queue occupancy did not exceed the threshold more often than the *exceed_limit* in a given *sample_size*. It was found that the control becomes too strict and there is no effective method of restoring the data rate after the congestion has cleared. A lower threshold was then introduced. This was implemented in the model by generating a positive signal if the number of times that the queue went below the lower threshold exceeded the *sample_size* minus the *exceed_limit*. If the queue remains between the thresholds, it implies that the system is stable and the data rates should not be changed.

Consequently, the congestion notification uses two bits instead of one to indicate one of the three states depending upon whether the queue occupancy is:

- below the lower threshold -- 00
- above the higher threshold -- 01

- between the two thresholds – 10

This is considerably more informative than feedbacks in Hysteresis or RED methods. The receiving end would generate a negative feedback if the congestion bits are 01 and a positive feedback if they are 00. The sources would change the data rates according to the feedback received. If the congestion bits are set to 10, this implies that the queue occupancy is between the two thresholds and no feedback is sent.

7.3.2 Stability Improvement

Some initial experiments also showed that the sampling technique of the algorithm had a potential problem. If, for example, congestion is detected towards the end of a sample of 2000 packets, it is likely that the congestion will continue into the next sample. However, if the counters and congestion identifier parameters are allowed to reset at the end of the *sample_size*, the congestion will next be reported when the *exceed_limit* is violated again which would mean that the system has to wait for a few packets to arrive before the persisting congestion can be notified again. Until then, the system would report no congestion as the *exceed_limit* is not violated, even though in fact congestion persists. This could, therefore, lead to an unstable behaviour.

The problem was solved by "elongating the sample" during the periods of congestion. For example, initially sample size of 2000 samples would be chosen and congestion notification would be triggered if 14 (or 28 if liberal approach was used) instances of queue size exceeding the higher threshold were recorded. The sample would be then elongated and assuming that the congestion persists, recording a further 9 instances would generate another feedback. As the queue was sampled every time a packet left service and the average packet length was 80000 bits, for the queue service rate of 12

Mbit/s, this method generated a feedback every 60ms during congestion. This was adequate since the average time between successive packets was 40ms in these experiments. Higher service rate will generate more frequent feedbacks. Further details of the full algorithm and pseudocodes are given later in this chapter.

7.3.3 Frequency of Feedback Signals

The control system will have a huge overhead of signalling if a negative feedback is generated and sent to the source, for each packet received with congestion notification. This is the case in Hysteresis and RED used earlier in Chapter 6. Therefore, a method of generating feedback when the congestion status changes and regular feedbacks during persistent congestion was developed using a *sample_identifier*. The *sample_identifier* is a value that is incremented when a new sample is started or the existing one is elongated. This achieves the desired effect of minimal signalling during light load situations and increase in frequency during congested periods. Further details on the operation of *sample_identifier* are given in the next section.

7.4 Final algorithm

This is a complete control scheme consisting of algorithms at the bottleneck link, the destinations, and the flow monitors. A congestion detection algorithm sits at the bottleneck link and sets the appropriate bits in the packet header in order to indicate congestion. The destination process generates the feedback signals. The flow monitors change the data rate emanating from the sources according to the feedback received. For consistency, the flow monitors respond in similar way to the feedback signals in case of Hysteresis and RED except they now receive both positive and negative feedbacks, see Appendix II.

```
Initialisation (executes first time only):
sample_size = 2000 (specified by the user)
exceed_limit = 14 (specified by the user)
queue_exceeded_count = 0
queue_reduced_count = 0
sample_count = 0
congestion = 0

when a packet arrives:
increment sample_count
while sample_count< sample_size
keep a running count of the number of times that the queue
occupancy:
exceeds higher threshold → queue_exceeded_count
and the number of times it
stays below lower threshold → queue_reduced_count

if(queue_exceeded_count > exceed_limit)
congestion = 1
elongate_sample();
else if(queue_reduced_count > sample_size - exceed_limit)
congestion = 0;
```

Figure 7.1: Pseudocode for Congestion Detection

Figure 7.1 shows the process for congestion detection. Every time a packet leaves the node, this function is called, and the queue occupancy is checked against the two thresholds; the *congestion_value* is worked out and the packet count is checked against sample size to look whether the end of the sample has been reached. The congestion is evaluated as follows: The number of times that the queue exceeds the higher threshold or stays below the lower threshold are recorded in two separate counters, *queue_exceeded_count* and *queue_reduced_count* respectively. When the

queue_exceeded_count exceeds the *exceed_limit*, the *congestion_value* is set to 1 and the sample is elongated. When the next packet leaves the node, the queue size is again checked against the thresholds and counters are changed accordingly. The *queue_exceeded_count* is now compared with the new value of *exceed_limit.* If the *queue_exceeded_count* exceeds the *exceed_limit* before the end of the sample (the elongated length) then the sample is elongated again. All this time the *congestion_value* stays at 1.

If the *queue_reduced_count* exceeds *sample_size* minus *exceed_limit*, then the link is being under-utilised and hence congestion is set to 0. This would only occur if the end of the sample has been reached without the queue size violating the higher threshold more than an acceptable number of times. If neither of these cases are asserted then congestion value remains the same as it was the last time.

```
change the sample identifier  (toggle or increment)
increment sample size and exceed limit
   sample_size += 1000;
   exceed_limit += 9;

leave counters unchanged
```

Figure 7.2: Pseudocode for elongate_sample() function

The function *elongate_sample()* is called when *congestion_value* is set to 1, (Figure 7.2). It increments the value of *sample_identifier*. For practicality, this can be set to increment to a fixed value, say 100, and then loop back to 0. The number is used to distinguish between two consecutive samples. A simple toggle could be used but it would be confusing in a node serving many flows, as two successive packets from a flow may get

the same *sample_identifier* value even if they are several samples apart in the aggregate. The feedback generation process in the destination would use this number to decide whether or not to generate a signal. The function also increments an index which is used to increase the value of *sample_size* and correspondingly, the *exceed_limit*, see Table 7.1. So, using the conservative set of values, if the original *sample_size* was 2000, we increase it to 3000 and the *exceed_limit* correspondingly increases from 14 to 23. (Using the liberal set of values, the increase would be from 28 to 39.) As a general rule, it was found that if the *sample_size* is increased by 1000, the *exceed_limit* should be increased by 9 for conservative values and by 11 for liberal values. All the counters are left unchanged. When the next packet leaves the node the code in Figure 7.1 will get executed with the new values but the counters are left unchanged and congestion is still 1. If the queue continues to be above the higher threshold and exceeds the new *exceed_limit* as well, then the sample will again be elongated. Conversely, if the queue stays below the lower threshold then the *congestion_value* will be set to 0. If neither of the conditions are true at the end of the sample, then *congestion_value* will be set to 2 and sample will be reset using the *end_of_sample()* function, see Figure 7.3.

```
change the sample identifier  (toggle or increment)
if (queue_exceeded_count < sample_size &&
            queue_reduced_count < sample_size-exceed_limit)
   congestion = 2;

reset other parameters
   sample_size = 2000
   exceed_limit = 14
   queue_exceeded_count = 0
   queue_reduced_count = 0
   sample_count = 0
```

Figure 7.3: Pseudocode for end_of_sample() function

```
when packet arrives:
   get congestion value and sample id;
if (congestion != 2)
   if (last_congestion_value != congestion_value || last_sample_id !=
sample_id)
      create a feedback packet with congestion value and send it;
      last_congestion_value = congestion_value;
      last_sample_id = sample_id;
else do nothing
```

Figure 7.4: Pseudocode for Feedback Generation

The generation of feedback is based on the *sample_identifier* and the *congestion_value.* The code is shown in Figure 7.4. This part of the control system will probably be situated in the destination processes, which can send the feedback directly to the sources. The feedback signal is only generated if the *sample_identifier* or *congestion_value* has changed. This is to ensure that a repeat feedback signal is sent to a given source at the maximum rate of once per sample so that the possibility of over-

reaction to congestion is minimised. No feedback is sent if the *congestion_value* is 2, which is when the system is in control and the Q_{99} is between the thresholds.

The interpretation of feedback signals by the controller processes is simple. If it gets a negative feedback (congestion = 1), it should reduce its data rate (by lowering the spatial resolution). If it gets a positive feedback (congestion = 0), it can increase its data rate. If no feedback is received, no changes are required.

7.5 Modelling Percentile Monitoring

The design of the simulation model was based on the one described in Chapter 5. The packet generator worked in exactly the same way but used a different packet format, see Appendix II.

The controller process received an explicit feedback for both increasing and decreasing the data rate and is therefore implemented slightly differently than before. As before, the process uses the Packet Length Reduction Factor (f) to regulate the data rate. This method was explained in Section 5.1.2. If a negative feedback is received, f is decremented by 0.1 unless $f = 0.5$. If it is positive, f is incremented by 0.1 unless $f = 1.0$.

The congested node process has significant changes made to it. The general structure remains the same as the one discussed in Chapter 5 but the behaviour is different. There is no need to obtain an average of queue occupancy and it implements the congestion detection function described above, see Figure 7.1. The values such as *sample_size*, *exceed_limit* etc were specified at the simulation time for flexibility in testing. Further specification details are given in Appendix II.

The destination process was changed in the way it treated the packets and generated feedback as described above (see Figure 7.4).

7.6 Results

With this approach of sampling and 99-percentile monitoring, the results have been much more promising. The fluctuations in f were found to be a lot less in comparison to the Hysteresis and RED investigated in Chapter 6. Figure 7.5 shows the trace of f for the case of gradually changing load (Table 5.1) with the conservative set of values. Figure 7.6 shows the results for the same with the liberal set of values. Refer to Table 7.1 for the conservative and liberal values. Both graphs exhibit the expected decrease in f as the load increases but the rapid fluctuations seen earlier are no longer present.

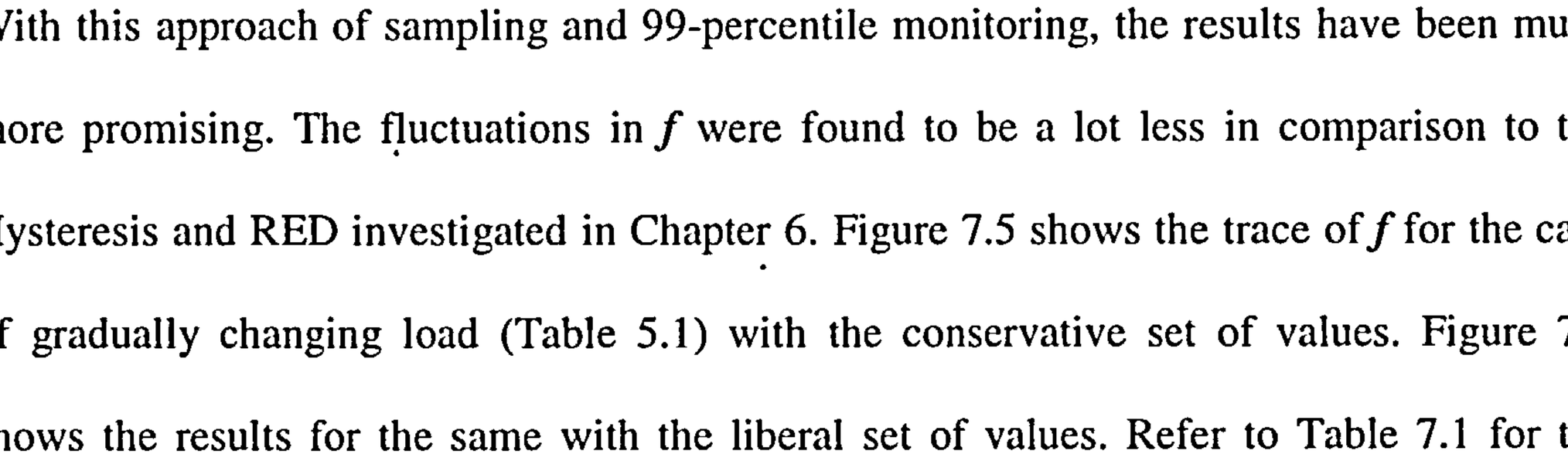

Figure 7.5: Fluctuation in f with 99-Percentile Monitoring conservative (gradual change)

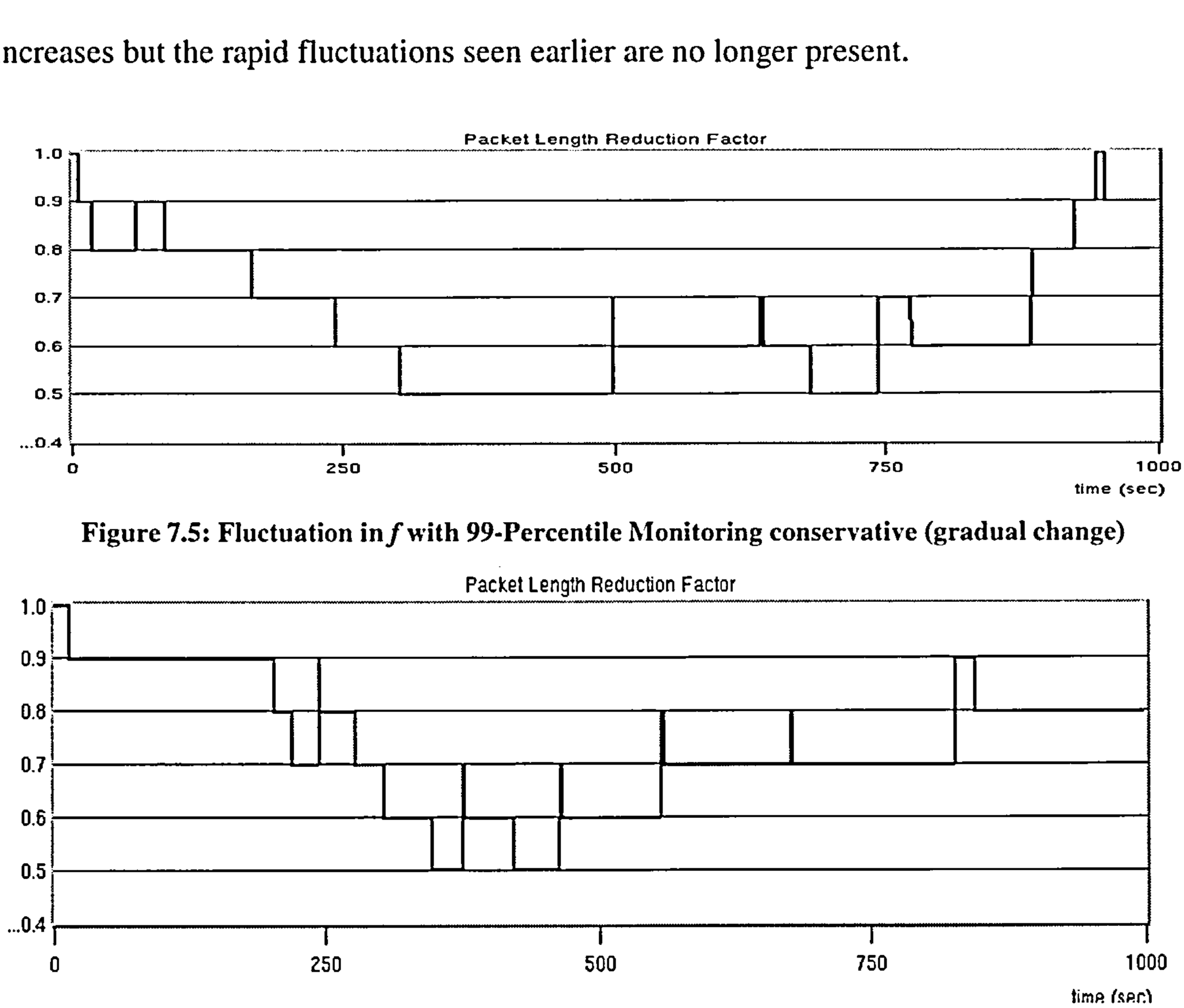

Figure 7.6: Fluctuation in f with 99-Percentile Monitoring liberal (gradual change)

There are, of course, some fluctuations because with any feedback-based system there is some inevitable delay between the feedback generation and responsive action. However, the fluctuations are much more controlled. In the case of the sudden change in

load scenario as well, the fluctuations are remarkably fewer, as seen in Figure 7.7 for conservative set of data and in Figure 7.8 for the liberal set.

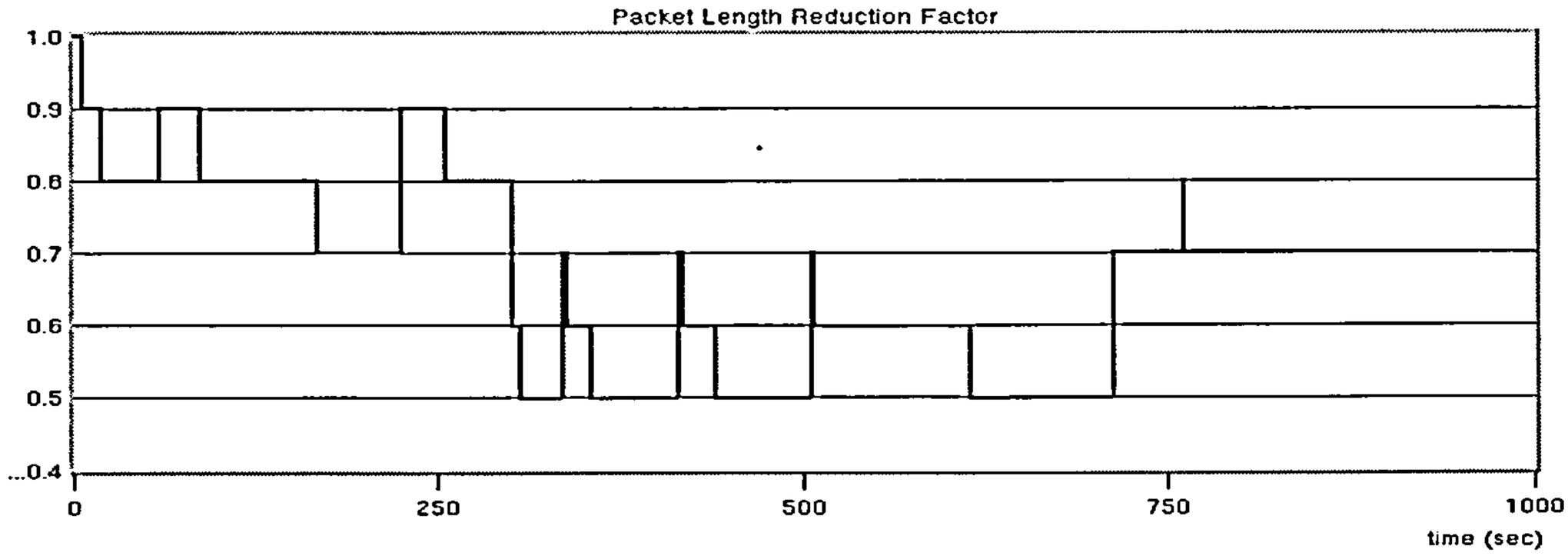

Figure 7.7: Fluctuation in f with 99-Percentile Monitoring conservative (sudden change)

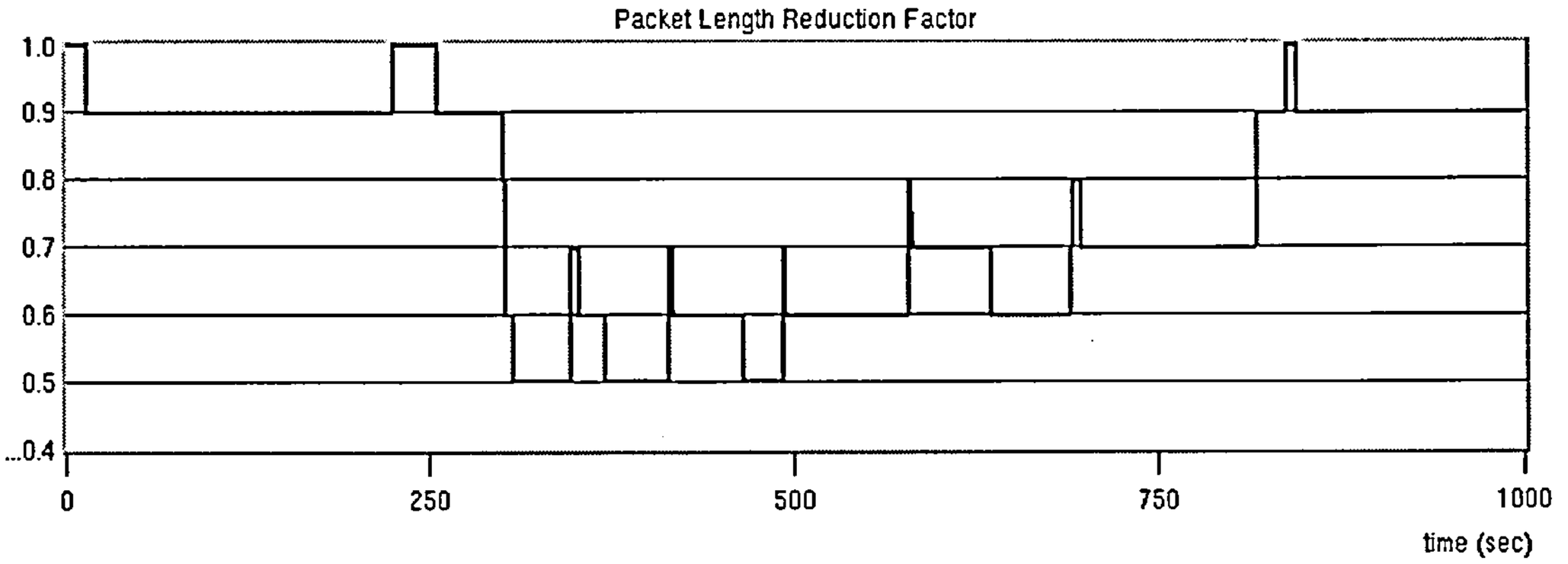

Figure 7.8: Fluctuation in f with 99-Percentile Monitoring liberal (sudden change)

The 99-percentile queue occupancy is also more controlled in the case of percentile monitoring. Table 7.2 presents the results for percentile monitoring based algorithm along with Hysteresis and RED. The results for Hysteresis and RED have been copied from Table 6.1. The results for percentile monitoring were collated using the method used for previous two algorithms. The high and low columns in Table 7.2 indicate the confidence regions for the results. The method is described in Section 6.3 in the explanation for Table 6.1.

With Hysteresis and RED, the on average Q_{99} is above the threshold. However, in all the cases, percentile monitoring outperforms the other two by maintaining the 99-percentile queue occupancy well below the higher threshold both with the conservative and the liberal sets of values. By controlling the high percentile of queue occupancy, percentile monitoring is able to minimise losses and as we have seen earlier in Figure 7.5-7.8, the changes made by the sources are more controlled.

Table 7.2: Q_{99} results for Hysteresis, RED and Percentile Monitoring

Algorithm	Gradual changes in load			Sudden changes in load		
	Average	High	Low	Average	High	Low
Hysteresis	493696.08	503265.39	484126.78	513803	528823.36	498782.64
RED	453597.25	458217.14	448977.36	451718	455468.29	447967.71
Percentile (conservative)	323977.25	330014.5	317940.00	324076.58	329837.32	318315.84
Percentile (liberal)	359407.33	364983.24	353831.43	360414.83	365432.4	355397.27

With any feedback method, it is likely that at times of sudden large increase in load, there can be some loss if the buffer size is finite, but our experiments have shown that Percentile Monitoring recovers from the sudden overload whereas the other two methods tested here would continue to overflow the buffer. The scenario presented here shows the extreme case where the sudden increase in load has been so large that the utilisation is greater than 1. In reality, the load conditions would be changing continuously but the utilisation would be, on average, between 0.6 and 0.8. In such cases, the percentile-monitoring method provides a graceful and responsive control.

Importantly, percentile-monitoring method is able to exercise tighter control on the 99-percentile queue occupancy by using significantly fewer feedbacks. This gives a major advantage on overheads.

7.7 Summary

The majority of existing bandwidth control schemes suggest monitoring mean buffer occupancy. While being simple, this is not a good measure particularly for real time traffic flows, which are bursty and delay sensitive. The performance of methods such as Hysteresis and RED when applied to continuous media traffic has been shown in Chapter 6. In particular, the investigations carried out so far show that both these methods can lead to frequent changes in the resolution while a majority of packets experience significant delay. The obvious mathematical alternative to monitoring mean queue occupancy would be to monitor a high percentile of the buffer fill. This was first introduced in Section 6.5.

In this chapter, we have described how the percentile-monitoring technique works and have demonstrated through simulations that the method provides better control on the buffer occupancy than when Hysteresis or RED (adapted for continuous media) are used. The frequency of rate adaptations by the sources is also lower, which implies that this system needs much less signalling to achieve better control on the congested node. The use of percentile monitoring is a novel approach that, to our knowledge, has not been used for flow control. The results have shown clearly that the performance of the percentile monitoring method is more appropriate for adaptive continuous media traffic than conventional methods based on measuring mean queue occupancy.

8

Control in a Hierarchical Network

In this chapter, we consider networks that are organized in a hierarchical manner and discuss how congestion control will need to be exercised at various levels in the hierarchy so that the entire system is capable of supporting continuous media traffic.

Firstly, we will illustrate a generic router model and show how the work presented in the preceding chapters fits into the context. Following on from that, we will discuss the structure of a hierarchical network and the direction of the second phase of work.

8.1 Generic Link Sharing Model for a Router

The structure of a generic link share model that can be used in a router for future networks suggested by [Ball 99a] is shown in Figure 8.1. This model is based on an amalgam of Class Based Queuing (CBQ) and Weighted Fair Queuing (WFQ) and has been shown to work well with a number of different types of traffic. It is assumed that the incoming flows would be classified into a number of classes according to their type, QoS requirements etc. In this example, there are 4 delay-sensitive classes: guaranteed audio, guaranteed video, adaptive audio, adaptive video, and a number of data classes.

Guaranteed services will require some form of admission control, for example, Measurement Based Flow Acceptance Control (MFAC). The flows that require guaranteed service would be served through a class that operates in isolation from the other service classes [Ball 99c]. Real time flows that are adaptive do not require strict guarantees but still prefer a high quality of service. These flows will be served in an adaptive class that again operates in isolation from other classes. There may be a number of such adaptive classes each with a different level of tolerance. The isolation is necessary so that the congestion in lower value classes or data traffic does not affect the performance of guaranteed and adaptive classes. Methods of congestion control that would be suitable for adaptive classes have been explored previously in this thesis. In particular, this work has demonstrated that a novel feedback system based on percentile monitoring is well suited to this class of traffic.

In a mutli-service capable router such as the one shown in Figure 8.1, there will be separate classes for each traffic type discussed above. In addition to these, there will be a number of delay insensitive data traffic streams, which must be dealt with separately. Callinan's work shows that CBQ is most appropriate for real time services whereas data traffic is optimally served by WFQ and this implies that the basic structure of the router should be divided into two main queues as shown in Figure 8.1. For a more detailed discussion see [Callinan 00a].

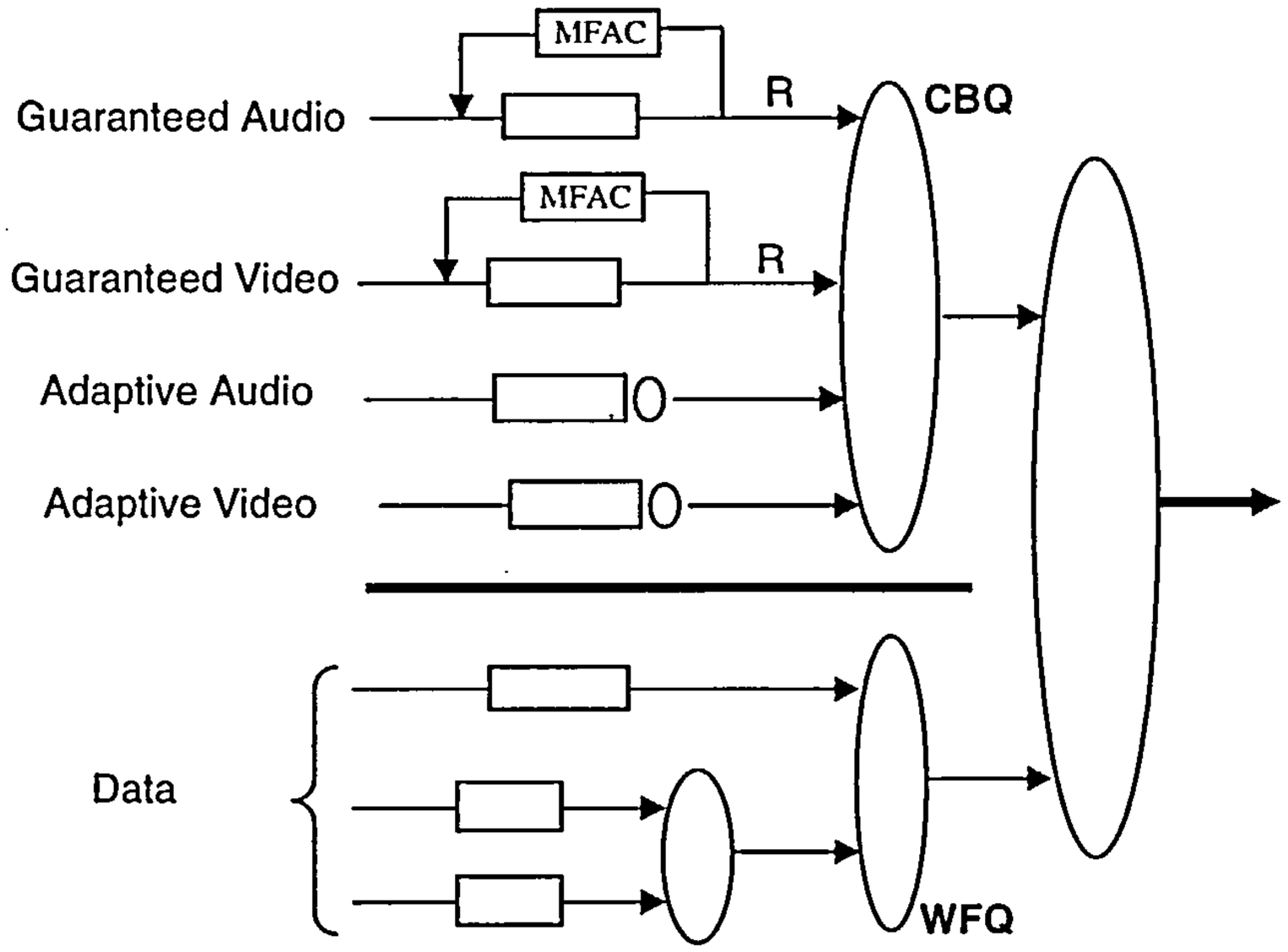

Figure 8.1: Hybrid CBQ/WFQ Link Share Model [Callinan 00b]

Any number of classes will be available in the link share model shown in Figure 8.1. Guaranteed and adaptive real time classes are treated with Class Based Queuing (CBQ) and data classes are served using Weighted Fair Queuing (WFQ). One of the scenarios where this type of router would be ideal, is in the network close to the end users such as an Internet Service Provider (ISP) which serves a number of home and small business users. The majority of the traffic will comprise e-mail and web surfing data but there will be some demand for real time traffic such as for interactive TV, video on demand, and networked games that require guaranteed or adaptive service.

8.2 Hierarchies in the Network

The Internet is organised in many levels of hierarchy. In a large company, individual flows from each computer on a LAN are aggregated on to one or more corporate routers depending on the size of the network and traffic demands. Flows from home users aggregate at the ISP's routers. The aggregated flows from a corporate or ISP router will be

carried by a carrier network[8], which may in turn have further levels of aggregation for trunk inter-continental routing handled by larger carrier networks.

The traffic at the aggregate level has to be handled in a different way mainly because both the characteristics and requirements are different from those of individual flows. Consider an example of several thousand video flows classified as adaptive traffic aggregating onto a single trunk route. It turns out that when several bursty flows are multiplexed together the resultant aggregate flow is smoother. This happens because the peaks and troughs in the individual flows are evened out by aggregation, thus leading to statistical multiplexing gain,[9] (discussed further in Section 9.5). Therefore, it is not straightforward to detect the effect of bursty flow on congestion at aggregate level.

Large carrier networks are generally designed to impose as little delay as possible. The routers inside the carrier network operate at wirespeed and are designed to forward the incoming traffic to the output ports without delay. This, however, invariably leads to congestion at the output ports. Ideally, the carrier network operator would like to provide a delay free service. A possible way to do this is to control the amount of traffic entering the network at the ingress router. The ingress router would typically receive an aggregate of flows. While the aggregate flows will be classified into a number of different classes, it is quite possible that within a class there may be sub-classes to provide finer classification, in order to serve the different quality of service requirements and delay constraints between the flows. Therefore, it is not appropriate, for example, to buffer all the flows in an aggregate in order to solve the congestion problem at the output ports. Some form of control may be implemented at the input ports of the ingress router or perhaps a better

[8] A carrier network in this context means a high-speed network capable of carrying aggregates of flows.
[9] It should be noted that certain types of data traffic, but not video and audio traffic, have been found to be self-similar in nature and they may not show this property.

solution would be put the control at the edge of the network, so that the volume of traffic entering the network can be regulated and feedback can be sent to the source of the flows.

8.3 Integration of Control Systems in Client and Carrier Networks

Figure 8.2 shows a schematic diagram of a network with two levels of hierarchy. In this diagram, a number of heterogeneous flows are served by the ISP router, which would be capable of multi-service and would differentiate the incoming traffic streams into classes according to their QoS requirements. The diagram shows the video traffic being forwarded on to the carrier network while the other traffic is sent elsewhere. In reality, the carrier network will be capable of multi-service and will have classes available for all kinds of traffic. The data traffic will be sent to a different class in the same ingress router. The author has chosen to present the network model in this way for simplicity. Because different classes operate in isolation from each other, it is not necessary to show other classes as only the adaptive video class is being discussed here.

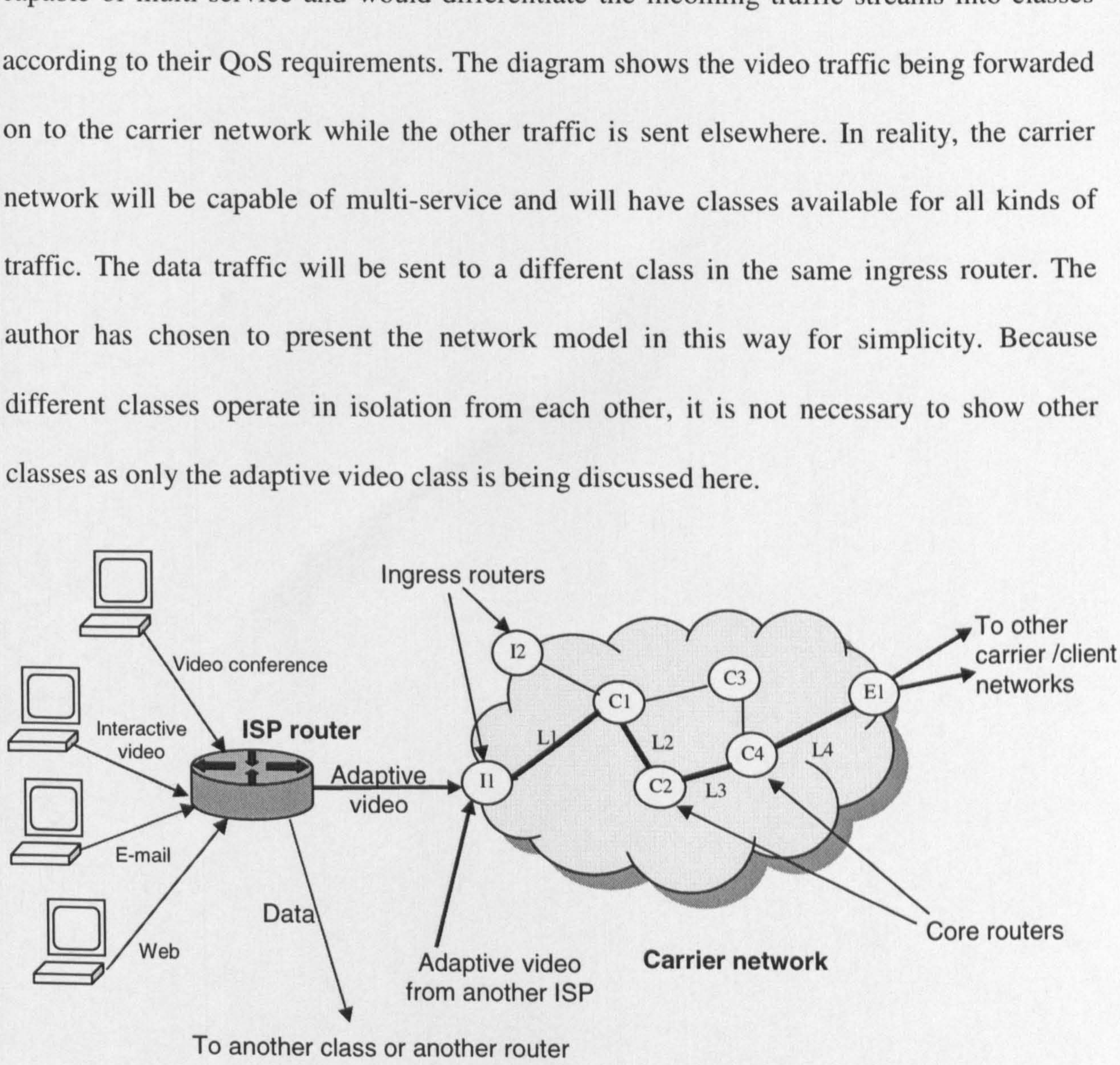

Figure 8.2: Hierarchical Structure of a Network

The ISP router will require some form of control system in order to regulate the video flows such that the output link between the ISP router and the carrier network is not congested. This is where the percentile monitoring method presented in Chapter 7 could be used. This method would monitor the traffic arriving at the ISP router and enable it to send feedback to the sources that can take corrective actions.

At the aggregate level, at the ingress to the carrier network, the adaptive flows from one or more ISPs may aggregate at the Ingress router I1. The flows arriving at the Ingress router I1 would most likely be aggregates of micro-flows. The aggregate flows would arrive at a number of input ports of the router I1 and in this example they are all assumed to be routed to a single output port which links the ingress router I1 to a core router C1, via link L1. All of the traffic entering the carrier network through I1 is destined to egress router, E1. The ingress router will usually set up a path (say, using MPLS protocol) and in this case, a path is configured via core routers C1, C2, C4 to E1 with the links labelled as L1, L2, L3, and L4.

Methods like percentile monitoring are not as effective here because although a feedback can be generated and sent back to the ISP, the flow from the ISP to the carrier network is an aggregate of video flows coming into the ISP. If the ISP decides to police the aggregated flow, it would delay all the individual flows equally, which would be unfair as they may have different contributions to the congestion. At this stage, we need a feedback system that takes into account the amount of "pain" caused to the network by each flow. This is where the pricing concept comes in. If we can fairly price[10] the bandwidth of link L1 between the ingress router I1 and core router C1, a feedback can be sent in terms of increasing price as the congestion increases on the link. The ISP can then use this price in a

number of ways. It can send the prices back to the sources so that the software in the source of the flow can apply the correct flow control policy and then use the prices for policing the flows. Alternatively, the ISP may have chosen to provide guarantees of high quality to the clients in which case it will pay the increasing price and hopefully make a profit through charging the clients at a premium rate. It may have been serving a range of adaptive flows, and can use a different type of feedback mechanism, such as that based on percentile monitoring, to notify the adaptive sources.

There are two main issues here, which are very important and interesting for network management. Firstly, at any time, there may be a number of different paths possible between ingress and egress router of the carrier network. It is essential that routing be maintained such that load is evenly distributed across all the links and no congestion "hot spots" are created. Secondly, the high-speed routers in a carrier network are designed to induce minimal delay. The network relies on the ingress routers to optimise the volume of traffic entering the network, ensuring both that the resources are not wasted and the delays are kept to a minimum.

8.4 Summary

This chapter acts as a bridge between the two parts of this research. In the first part, the emphasis of the research had been upon the analysis of existing control systems based on signalling and development of a more efficient technique. Extensive simulation was used to identify the problems with the control methods such as Hysteresis and RED which, although successfully used for data traffic regulation, were found to be inappropriate for

[10] Price is not necessarily related to real money. We use it in this context as a feedback term although it could be designed to have a relationship with monetary terms.

adaptive continuous media. A novel monitoring and signalling technique was developed and tested against the same paradigms and was shown to perform better.

The improved performance of the percentile-monitoring scheme led to a reflection on how it would fit within the network. It was noted that the congestion problem exists at different levels of hierarchy within a network. It may not be suitable to deal with each of them in the same way as the traffic characteristics and requirements may change. The delay at a single queue is controlled well by the percentile monitoring for the purpose of continuous media transmission. However, as we move deeper into the network, a different approach is required. The main issues pertinent to the aggregate flows in a carrier network have been highlighted. In the next chapter, we will discuss these issues in further detail.

9

Control in the Carrier Network

The situation within a high-speed carrier network is quite different from individual flows. Routers are designed to be fast and, increasingly, capable of multi-service such that they induce minimal delays to premium class of traffic. Certain classes will be given expedited service, that is, they will experience no delay and their packets will get served as soon as they arrive. Of course, this means that there may be other classes, which are throttled back and experience queueing delays. Also, the routers in the carrier network usually deal with aggregates of thousands of flows and often paths would be configured between the ingress and egress nodes using MPLS or similar protocol, thereby reducing short-term changes in routing. There are two prime considerations from the network management point of view: the traffic entering the network must be optimised for maximum revenue within a given delay constraint; and the load must be distributed across the network such that it does not lead to a few links being overloaded while others are underused. This chapter starts with a description of the Dynamic Resource Control (DRC) mechanism that was developed in an internal project at Nortel Networks to address these issues.

9.1 Dynamic Resource Control (DRC)

The Dynamic Resource Control (DRC) scheme developed at Nortel Networks addresses the dynamic bandwidth allocation problem in a high-speed carrier network. A DRC network has a multi-layer approach and would typically consist of a hierarchy of IP/MPLS paths operating on top of an optical lambda layer. As the traffic pattern changes, the demand on the bandwidth changes in the IP layer. For example, the demand on the IP layer may change several times during a day as the activity on the Internet usually peaks in the morning and afternoons and becomes light in the evening. In longer time scales, the demand on the high-bandwidth optical layer also changes. These time scales are likely to be several weeks or months long. For example, the demand on a corporate network may increase around the period when auditing is carried out, or when quarterly forecasts are made. The bandwidth allocation has to be resolved both for IP/MPLS layer and the lambda layer. In the following sections, the DRC scheme is described in relation to the IP layer. The problem of the lambda layer is orthogonal to this and research on this issue was also ongoing at Nortel [Michalareas 01].

The network has a number of edge routers, which perform the ingress control on the aggregates of incoming IP flows. The aggregates of flows are transmitted to another edge router through a number of MPLS pipes. Figure 9.1 illustrates a part of the network where the flow aggregate enters the network through an edge router (ER1) and is transmitted to another edge router (ER2) via a path through the mesh of core routers. The thick arrows in the figure show the path of the aggregate of micro-flows[11] from the client network.

[11] The term micro-flow is used to describe a flow from an individual user.

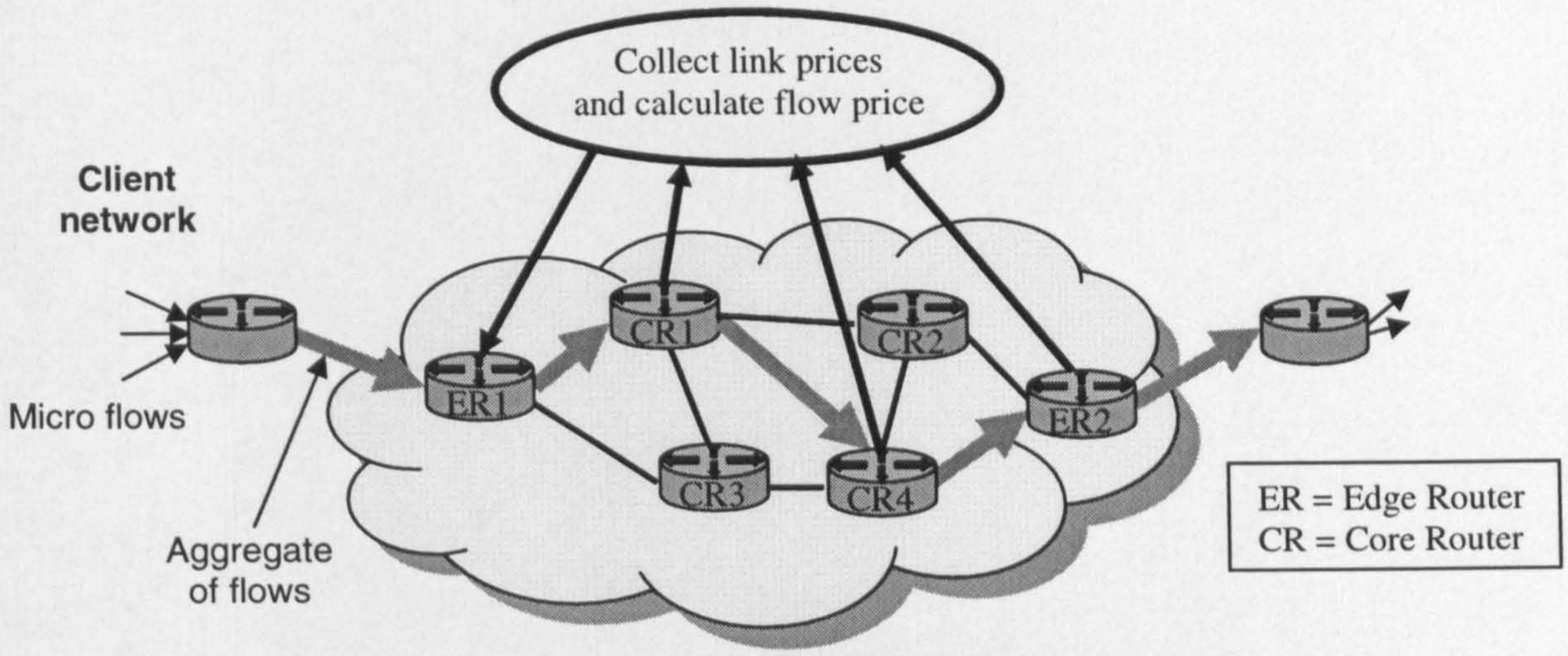

Figure 9.1: DRC Network Scenario

It is assumed that the protocols such as MPLS would be used to establish paths between the routers in the carrier network and hence the routing is static for the duration of the session. A number of things may still lead to congestion. For example, the volume of the traffic entering the network through all the ingress routers may change dynamically. Depending upon the topology of the network, some links may be common to a number of edge-to-edge router paths. This implies that the links between the routers must be capable of dealing with such 'hot spots'. This is discussed further in Section 9.4, DRC performance.

DRC uses a price-based feedback method. This mechanism employs a central or distributed monitoring function that polls each of the routers along the path to obtain a "link price" and then calculates a total "flow price" for using that path. This price is then fed back to the edge router (ER1) handling the aggregate flows from client networks. The ingress controller at the edge router can then police the traffic according to its "utility function." The term "utility function" of a flow denotes its value from the network operator's point of view. As the load in any of the links increases, the price will also increase and only the flows with high utility function will eventually remain in the system.

Hence, the network can be optimised for the high value customers who would probably generate more revenue.

9.2 Congestion Pricing

The concept of congestion pricing is based on economic models. The share of bandwidth is priced in the same way as goods are priced in the market. The case of congestion can be understood as a situation of low supply, which leads to a rise in the price. Research on congestion pricing is an active area with much interest [Courcoubetis 97, Gibbens 99, Kirkby 99a, Courcoubetis 00].

In DRC, the links within the carrier network are priced and the total path price is fed back to the ingress router to allow it to choose the least costly path and to enable it to regulate the traffic arriving at its input. We now take the pricing idea a bit further and propose to develop a feedback based on prices that the ingress router could send to the flows (aggregates of micro-flows), which are using its resources. Its importance is evident from the requirement that the ingress router must regulate the amount of traffic entering the network but often is not allowed to change the traffic behaviour directly because the arriving flows may comprise different classes and types of micro-flows. This was discussed earlier in Section 8.3.

Using "congestion price" as a feedback term has been suggested by [Kelly 97] so that traffic with higher utility function, and hence higher revenue for the network operator, is maximised in the network. This idea of usage based charging has been explored by many researchers [Kirkby 99b, Carroll 99, Key 99, Gibbens 99, Kirkby 99a, Kunniyur 00, DaSilva 00]. In [Biddiscombe 00], extensive research has been carried out to develop methods of bandwidth allocation based on price and "Willingness to Pay" (WtP).

The term "congestion price" is a variable value that relates to a flow or a source the degree of load it is putting on the link. It provides an effective method of policing and shaping the traffic flowing into the bottleneck node. The flows either pay the price for the service or must be throttled back at the source. If the congestion persists, prices increase further and the flows that cannot pay enough suffer further delays. Eventually, only the flows that can match the price are allowed to remain and receive the high quality of service. Pricing also eliminates the need for a per-flow signalling because the prices can be broadcasted regularly and a new flow can receive this information without any signalling. In carrier networks with aggregates of thousands of flows, it is particularly important that per-flow signalling is avoided.

It must be noted that the "price" does not necessarily have anything to do with real money although obviously there may be a separate tariff agreement by which the users can be charged for their use of a particular class of service over, say, a month.

9.3 Inelastic and Elastic Flows

The work carried out for the DRC scheme focussed on inelastic and elastic flows. The notion of flow elasticity is similar to price elasticity in economic models. An inelastic flow is one that has a rigid QoS requirement and once it has been accepted, it will continue to pay irrespective of how high the price becomes, to maintain its bandwidth for the duration of the session. However, it may terminate if the bandwidth requirement cannot be met. Such flows will require a guaranteed service and the session will be set up only if the network can provide the guarantee. Also, an inelastic flow has an associated rate and will not start using more bandwidth if it becomes available. An elastic flow on the other hand, will make use of any amount of bandwidth made available to it, but it has a limit on the cost it is willing to afford. As the congestion increases, the price/unit bandwidth will

increase and the flow will get progressively less bandwidth at the same cost. This behaviour is better described as uni-elastic or unitary elastic [Case 02]. Figure 9.2 illustrates the properties of inelastic and unitary elastic flows.

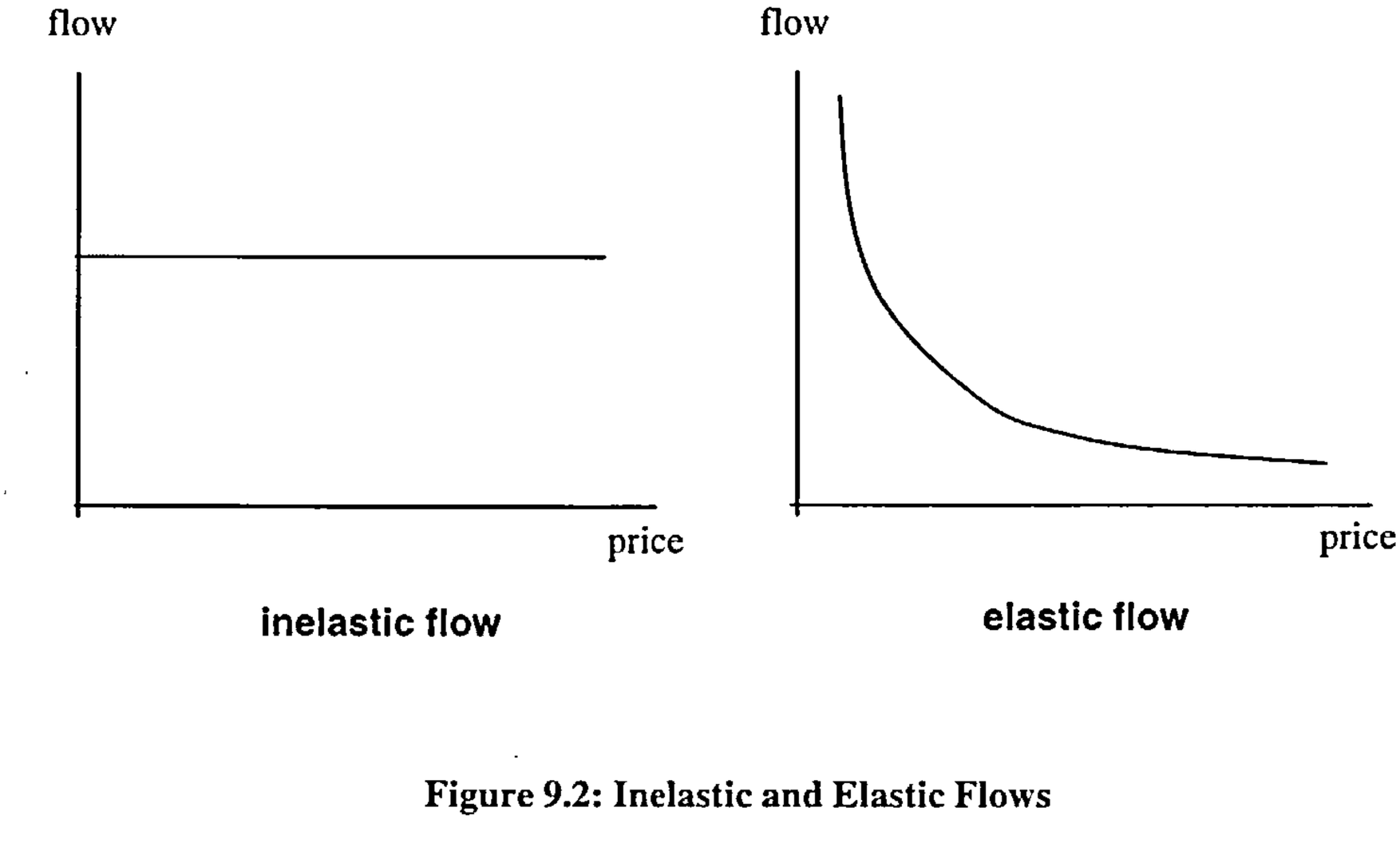

Figure 9.2: Inelastic and Elastic Flows

9.4 Path Selection using DRC

For stability, DRC network has to impose an upper limit on the amount of inelastic traffic each edge router can allow into the network. Firstly, a study of DRC and Diffserv networks was carried out to compare the performance in terms of volume of traffic carried by each network. The Diffserv network referred to here is one without any form of resource reservation, hence termed Vanilla Diffserv. A study was carried out using a small model of a partially meshed network to compare the volume of inelastic traffic that can be supported by both networks. The model consisted of 14 edge routers and 5 core routers with a static routing table for two-way traffic between all edge routers to all the other edge routers. The paths were chosen as the most direct routes and all links had equal capacity, and hence the utilisation was high (almost 80%) on some links between the core routers as they were common to more than one edge-to-edge path. A number of scenarios were considered:

traffic flow between 3 or 4 routers chosen randomly with all the traffic from any one router going to a single router; a distributed traffic flow such that traffic from any one router was forwarded to more than one edge router; and also an extreme situation where all the traffic from 11 routers was destined to a single edge router. The results suggested that Vanilla Diffserv has to limit the amount of inelastic traffic to approximately 10% of its total capacity at each edge router while DRC network can on average allow up to 4 times as much traffic [Tater 00c, Tater 00d]. Vanilla Diffserv has to place a rigid limit on the traffic that has to be served with minimal delay, because it is not aware of the congestion situations and has to be configured to cope with the worst case. It was apparent that as the network size increases the Vanilla Diffserv would have to employ a more stringent control on the amount of inelastic traffic entering the network. DRC performed better with this problem using the pricing mechanism because the increasing prices provide a feedback to the ingress router regarding the congestion within the core of the network.

An ingress router may have a number of output ports, each of which may be configured to transmit data along a different path through the carrier network. It is important to ensure that the load is evenly distributed across the network to avoid the possibility of causing delays along one path while another remains underused. The pricing mechanism of DRC scheme provided a solution to this. As the load on a link increases, the price of that link increases thereby increasing the price of the path that contains the link. The ingress router is informed of all the possible paths to reach a certain egress router and prices on each of them, which can then choose the path with lowest price.

9.5 Statistical Multiplexing Gain

It is well known that aggregates of bursty traffic generally require less bandwidth than the sum of their peak rates, the gain in capacity usage due to this is known as the statistical

multiplexing gain. As has already been mentioned in Section 8.2 some types of data traffic, such as web, exhibit self-similarity [Leland 94, Tuan 98, Sahinoglu 99] where the statistical multiplexing gain may not be true. However, here we are concerned with real-time video and audio traffic, which have not been found to show self-similarity.

When bursty flows are aggregated, the mean (m_A) of the aggregate is a linear sum of the mean of the flows (m_i). However, the peak rate of the aggregate is less than the sum of peak rates, as they do not add up linearly. Instead, it is found that the variance of the flows (v_i), which is the square of standard deviation (σ_i), adds up linearly to give the aggregate variance (v_A).

Expressed formally,

$$m_A = \sum_{i=1}^{n} m_i \qquad \textbf{Eq 9.1}$$

$$v_A = \sum_{i=1}^{n} v_i = \sum_{i=1}^{n} \sigma_i^2 \qquad \textbf{Eq 9.2}$$

It is known that for Poisson distributions the majority of the values lie within $m_j \pm 2\sigma_j$. We generalise this by replacing 2 by k. For each flow i, a good estimate of the peak rate is, then given by:

$$peak_i = m_i + k\sigma_i \qquad \textbf{Eq 9.3}$$

The peak of the aggregate is similar to peak of a flow, therefore, from Eq 9.3

$$Aggregate\ peak = m_A + k\sqrt{v_A} \qquad \textbf{Eq 9.4}$$

Substituting Eq 9.1 and 9.2:

$$Aggregate\ peak = \sum_{i=1}^{n} m_i + k\sqrt{\sum_{i=1}^{n}(\sigma_i{}^2)} \qquad \textbf{Eq 9.5}$$

The linear sum of peak rate would have been equal to:

$$\sum_{i=1}^{n} peak_i = \sum_{i=1}^{n}(m_i + k\sigma_i) = \sum_{i=1}^{n} m_i + k\sum_{i=1}^{n}\sigma_i \qquad \textbf{Eq 9.6}$$

By inspection, we can see that the peak rate of the aggregate will be equal to the linear sum of peaks of all the flows only if the sum of the standard deviations is equal to the square root of the sum of the squares of standard deviations. But this is true only when $n = 1$. The ratio of the linear sum of peak rates to the aggregate peak gives the statistical multiplexing gain.

We carried out a test to see how many bursty flows a 1000 Mbits/s channel could carry using the aggregation shown above. We fixed the peak to mean ratio of the flows to 5. It was found that the channel could carry more than 9600 flows all with mean rate of 0.1 Mbits/s and peak rate of 0.5 Mbits/s giving a statistical multiplexing gain of 4.8 whereas if allocation had been made according to the peak rate, the channel capacity would have had to be over 4800 Mbits/s. Of course, it is unrealistic to assume that all flows have exactly the same mean and peak rates. So, the tests were repeated with flows of different magnitude but each with a peak to mean ratio of 5. This time only 1800 flows could be carried but importantly the gain was still as high as 4.5. The reason the number of flows was less was that peak rates of some flows were up to 5 Mbits/s.

9.6 Adaptive Flow

So far, the DRC project had focussed on inelastic and elastic flows. In line with the research into the adaptive continuous media traffic carried out already, the author decided

to investigate the aggregate behaviour of adaptive traffic and how the pricing concepts could be developed for such traffic.

Until now, an adaptive flow had simply meant a co-operative flow in this work. In conjunction with pricing concepts, we now redefine the adaptive flows in the context of its price elasticity. In practice, most of the real time traffic flows are likely to behave in a hybrid of inelastic and elastic flows, which was termed as an adaptive flow. Such a flow will have a range of bandwidth that delivers acceptable QoS from a minimum to a desired maximum bandwidth and it will also have a limit on the cost it is willing to afford. When the network is not congested and prices are low, the flow will get its preferred QoS. The flow behaves in inelastic manner (i.e., constant bandwidth) until the price/unit bandwidth increases to the same level, as it is willing to afford. Further increase in price will lead the flow to assume a unitary elastic behaviour. When the price has become so high that the flow cannot "buy" the bandwidth for its minimum QoS requirements, it may choose to drop out (non-persistent flow) or it may choose to pay the increasing price (and afford the cost beyond its limit) to maintain the minimum QoS (persistent flow).

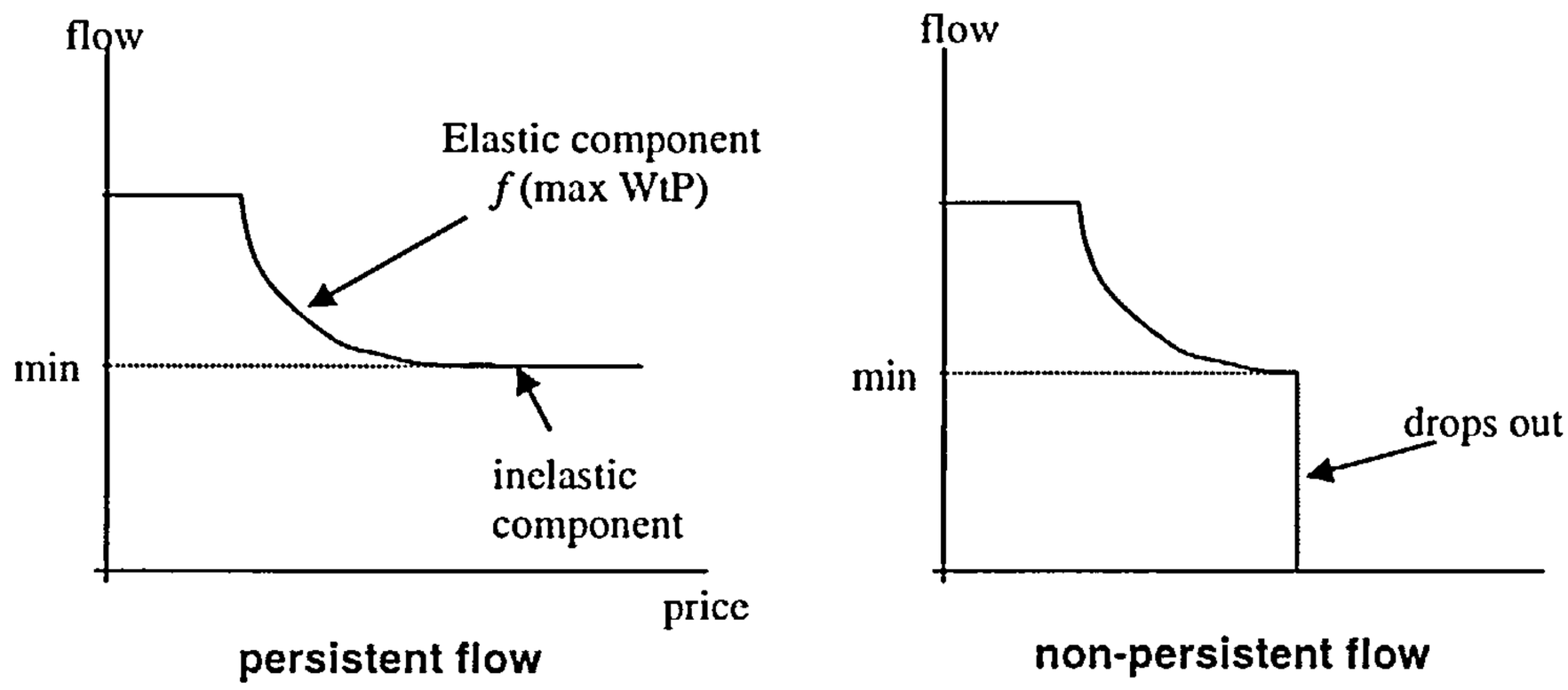

Figure 9.3: Persistent and Non-persistent Adaptive Flows

As we can see, an adaptive flow combines the properties of unitary elastic and inelastic flows. Inelastic flows can be considered as persistent adaptive flows where the minimum acceptable values for QoS parameters are equal to the desired values. Conversely, elastic flows are adaptive flows where the preferred values are very high and minimum acceptable is almost zero. Therefore, we shall only consider the performance of adaptive flows in the following chapters, unless otherwise stated.

9.7 Traffic Control at the Ingress Router

The study of DRC project highlighted a related issue that needed to be investigated further. While the main objectives of the project were to optimise the volume of high value traffic through the carrier network and thereby increase revenue, it was found that a congestion control problem exists at the edge of the network. The network typically consists of high-speed routers with trunks or paths configured from end to end and it is important to ensure that queuing delays within the network are avoided. The real time traffic will most likely be treated with expedited service. For the guaranteed class, the ingress router will perform some form of call admission control to ensure that the resources are allocated appropriately. For the adaptive class, rigid call admission control is not ideal but it is essential to keep the delays to a minimum level while optimising the number of flows that generate revenue. In Chapter 7, percentile monitoring method was shown to provide a feedback-based solution suitable for adaptive continuous media traffic. However, in carrier networks such traffic will require expedited service so significant delay is unacceptable. Since build-up of queues in the routers has to be avoided, monitoring the 99-percentile of the queue occupancy will not yield useful measurements. A unique solution is required that measures the traffic characteristics without inducing significant packet delay.

As discussed earlier in Section 8.3, to guarantee high QoS it is necessary to provide some form of admission control at the edge of the carrier network. For adaptive services, a simple yes/no admission control mechanism is inadequate. We suggest a congestion pricing method that calculates the price for the output queue of the ingress router on a second by second basis and send the price to rate control software that regulates flow rate at the source. The source of flow in this context means the point in the network where the decision to change the flow parameters can be made. This could be at the router of the service provider with appropriate agreements with its clients or right back where the flow started. Alternatively, the traffic regulation could be carried out at the input queues of the ingress router in which case the rate control software will operate at the inputs. In some cases, this configuration may be appropriate. However, here we shall consider the rate control at the source of the flows.

To demonstrate the feasibility of a scalable price based control system, the simplest scenario is shown in Figure 9.4. It can be related to the network scenario of Figure 8.2 as follows. The flow sources shown here are the sources as defined above, i.e., the nodes that control the traffic parameters. The flow controller is a process that receives prices from the network and uses them to regulate the amount of traffic sent at the source. We assume that all the traffic arrives at router 1 which is same as the Ingress Router I1 in Figure 8.2 and is routed to a single output port. This output port is linked to Router 2, which is the Core Router C1 in Figure 8.2 and hence the link between routers 1 and 2 is same as the Link L1 in Figure 8.2. We assume that this link is the single point of congestion. Prices are calculated for this link and will vary as the load on the link changes. Prices are signalled back to flow controllers.

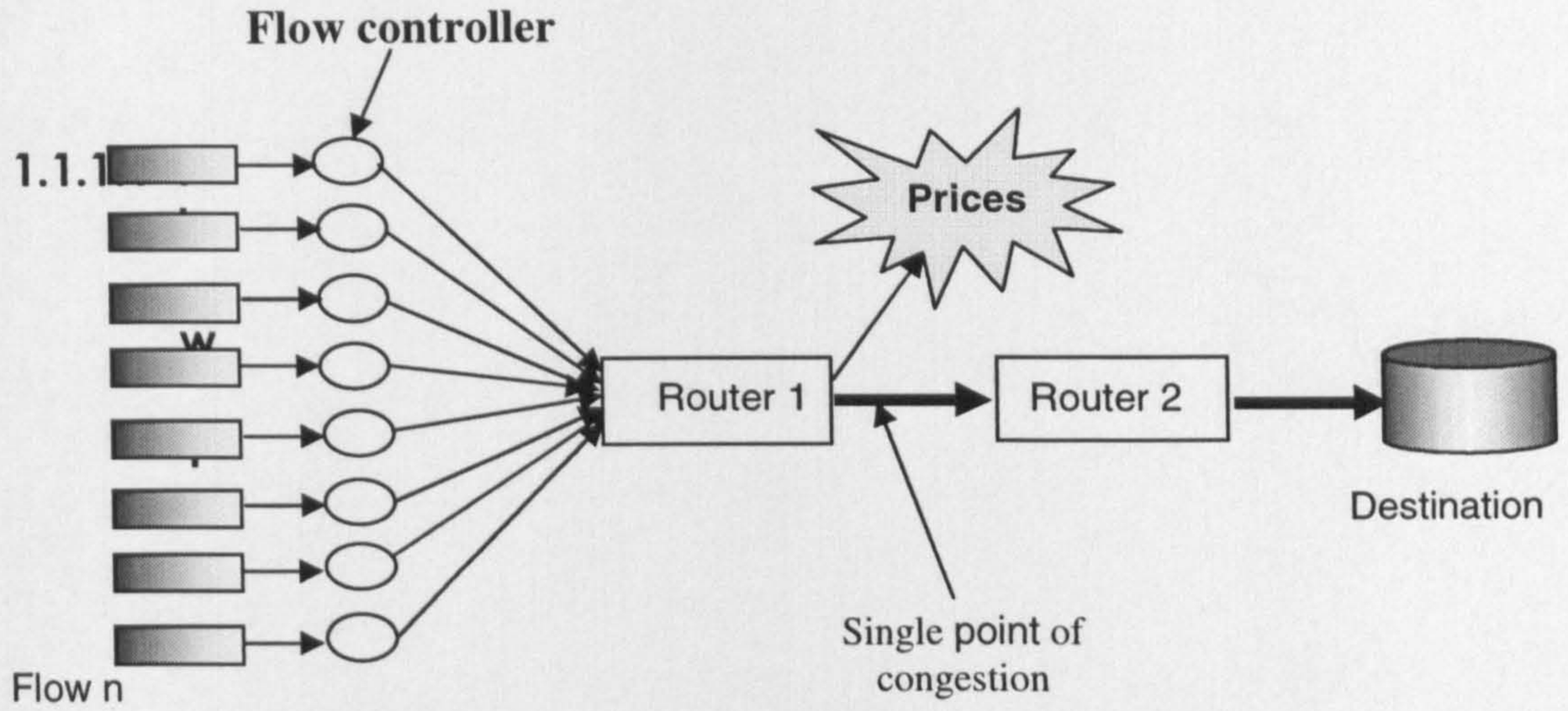

Figure 9.4: Network Scenario for Price Based Control System

Since we are only interested in the feedback process, a single destination is adequate for modelling purposes in so far as the feedback process is concerned the data packets are not used once they leave the output link of Router 1 and the destination can simply destroy the packets. Of course, in reality, the packets would be forwarded on to one or more destinations.

9.7.1 Second Moment Measurements

It is a well-established practice to use mean burst rate and peak rate for bandwidth provisioning of bursty flows using leaky bucket and token bucket algorithms. However, for continuous media traffic it will be more appropriate to use the variance, or the second moment, of the flow for bandwidth allocation as it gives a clearer representation the bandwidth usage. In recent research, [Knightly 97, Knightly 99] has suggested a measurement using rate-variance envelopes where both mean and variance of incoming flows are monitored.

9.7.2 *Pricing the Aggregate at Ingress Router*

In Section 9.5, we showed the huge advantage to be gained if the ingress control system dealing with bursty traffic used a more sophisticated scheme of bandwidth allocation than allocating peak rates to each flow. We noted earlier in this chapter that although high percentile monitoring has been shown to be suitable for continuous media it is not an adequate option in the carrier network scenario. We have suggested using a price-like feedback, but we need to design a method such that prices are fair and therefore, charge the flows for their contribution to congestion. We propose a marriage of first and second moment measurements and pricing.

In this way, prices will be calculated separately for mean and variance of the aggregate flow and sent to the respective sources of all the flows that make up the aggregate. Each flow will thus be charged according to its contribution to the load on the ingress router. Using separate prices for the mean and variance of aggregate flows will give flexibility to the flow controller to choose between making the flow smooth with good mean throughput or keeping it bursty with low mean rate. The policies can be made in a number of different ways and they are a usual part of network management.

There are some important issues related to the pricing method. Aggregates of flows must be priced so that it is fair for the individual flows, which may have different degree of burstiness. It is also essential that the method does not require per-flow signalling because the overheads to signalling to thousands of flows will render it unscalable. These will be discussed further in the next chapter.

9.8 Summary

This chapter describes the author's study of the DRC project and the surrounding issues. The DRC mechanism was used to analyse the traffic characteristics in the carrier network and to estimate the benefit due to statistical multiplexing gain. The investigations highlighted that a carrier network ingress controller is better at optimising the volume of high value traffic if it has knowledge of the congestion status within the network. The DRC solutions provide a mechanism of collecting prices along the path and feeding this back to the ingress router. The link prices give an indication of load along the links, giving the ingress router an opportunity to choose the cheapest path. Some crucial pieces of the jigsaw were missing:

- So far, DRC was used with inelastic and elastic flows based on economic market theory of elasticity. The author has incorporated adaptive flows with the economic models as they are potentially the most useful flows for real-time traffic.

- The ingress router can select a path for the arriving aggregate flows to route them through the carrier network but has no method of indicating the prices to the adaptive flows.

Considering that an aggregate flow may contain flows with some variations in characteristics, it would not be appropriate to simply send the path price to the flows. It was proposed that a method that combines measurement of mean and variance of arrival processes and generates separate prices would give a desirable solution as each flow would be charged for its contribution to the congestion in the carrier network. This leads to the main task of designing a pricing function that is fair and scalable and which enables the ingress router to effectively control the traffic entering the network. The development work, simulations, and results are presented in the following chapter.

We also found that the benefits of statistical multiplexing gain are huge in bursty flows. We shall show how our pricing method incorporates the bandwidth allocation that benefits from statistical multiplexing gain.

10

Pricing Mean and Variance

We have seen that the ingress router to a high-speed carrier network has to accept the incoming traffic such that the value of the traffic is optimised while ensuring that the delays are kept to the minimum possible. In the last chapter, we suggested that a congestion pricing mechanism, which generates separate prices for mean and variance, should be used at the output queue of the ingress router to the carrier network. The method must be able to support different types of flows through dynamic bandwidth allocation as the network conditions change. For example, there might be flows, such as interactive video, which can tolerate overall reduction in resolution, but would prefer to maintain the burstiness, while there might be other flows, such as video on demand, that would prefer to smooth out their data rate through buffering in order to keep a high resolution. If the system sends two separate prices for mean and variance, then these flows can be regulated according to their individual requirements, and receive the service that they pay for.

In Section 9.7, we suggested generating mean and variance prices on a periodic (for example, second by second) basis with a rate control mechanism that regulates the flow rate at the source of the flow. We now describe the method of measuring the mean and

variance of the aggregated flow arriving at a node and deriving prices such that the constituent flows can be charged in proportion to their contribution to the load.

10.1 Explanation of Terminology

Before we begin the description of the algorithms, we need to reinforce the meaning of the terminology used. The constraints of language often lead us to using the same word in two different contexts, which can cause ambiguity. Hence, it is important to define the key terms as used here, mainly mean, variance, willingness to pay and charge.

10.1.1 Mean and Variance

Mean is one of the most widely used tools of statistics. It is relatively simple to monitor the mean of a constantly changing attribute by using a moving average algorithm. An exponential weighted moving average was used in Chapter 6 to calculate the average queue occupancy, see Eq 6.1. In this scenario, we require the mean arrival rate and we estimate this by metering the number of bits that arrive at the node in a unit time, say 1 second. This provides us with a very simple method of measuring the mean arrival rate, but, as we noted before in the discussion of percentile monitoring, mean alone may not represent the traffic behaviour accurately.

The variation in instantaneous arrival rate from the average depends upon the burstiness of the traffic arriving at the queue. We need the second moment of measurement, the variance. Variance is the square of standard deviation. Standard deviation, σ, is defined as:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (Q_i - Q_{avg})^2}{n}} \qquad \textbf{Eq 10.1}$$

where Q_i = instantaneous measure of number of bits that arrive at the node

and Q_{avg} = average of instantaneous values of number of bits arriving.

Therefore, variance is given by:

$$\therefore Variance = \sigma^2 = \frac{\sum_{i=1}^{n} (Q_i - Q_{avg})^2}{n} \qquad \textbf{Eq 10.2}$$

Standard numerical methods exist to measure the variance of a set of data through iteration [Press 93]. However, in this case we need to measure the variance in real-time in order to generate the price feedback. We adapted the standard method to develop a calculation method based on a periodic count of bits arriving at the node over two timescales. In addition to measuring the number of bits over 1s period (for mean rate estimation), we also count the bits arrival every 5ms. From these 5ms samples, we calculate the variance at the end of the second. We can then use the mean and variance values to derive the peak rate estimations, as we shall see later. The double counting has to be taken into account for this estimation and is explained in Appendix IV.

10.1.2 Willingness to Pay

Willingness to Pay (WtP) is measured in tokens and can "buy" flow attributes, which in this case, are mean and variance[12]. As a given link starts to become congested, a feedback (in terms of increasing price) would be sent to the flows. The flows may have different priorities for delay and throughput, and each may have a different WtP. At any time, only the flows that have WtP to match the current price can continue to exist. Referring back to the types of flows discussed in Chapter 9, an inelastic flow effectively has infinite WtP. It

will therefore continue to pay for the quality as the prices increase. Of course, in practice, a flow cannot be truly inelastic as it will have some limit albeit high on its WtP. When such a flow can no longer afford the price, it will cease transmission. An adaptive flow will match the price until it exceeds its WtP, after which it will tolerate a degradation of quality of service until it degrades to the accepted minimum. Beyond this, a non-persistent adaptive flow will cease transmission while a persistent adaptive flow will behave inelastically to maintain the minimum quality. Finally, as used in this thesis, an elastic flow has very low or no WtP and only transmits using the available bandwidth, thus behaving as a best-effort flow. The network operator can use various configurations for optimisations. For example, it may impose a maximum limit on the WtP that a flow may have. It may use a control capacity for price calculation that is actually less than its service rate. The effect of these policies will become clearer later in this chapter.

10.1.3 Charge

The actual number of tokens being used to pay for mean or variance is defined here as charge. A flow may allocate more charge for mean than for variance or vice versa, but at any time the total charge in use must not exceed its WtP. In this context, charge can be thought of as a payment made by the flows.

10.2 Objectives of Pricing Algorithm

Several important issues need to be considered to design a scalable and usable method of calculating prices. The main considerations are Scalability and Fairness.

[12] WtP tokens can pay for both mean and variance although they have different units, bits/s and bits/s^2.

10.2.1 Scalability

The price calculation process must be independent from the flow activity. The ingress node must be able to calculate the prices based solely on the parameters it can measure at the aggregate level such as aggregate mean and variance, spare capacity etc.

10.2.2 Fairness

The distribution of charge tokens between mean and variance within a flow and the distribution of charge tokens among the flows should not affect the aggregate peak rate and spare capacity as long as the grand total of charges from all the flows remains constant. So for example, if the price increases, a flow may choose to decrease the tokens being used for variance and increase the tokens for mean by the same number. The resulting increase in mean and reduction in variance should be such that the aggregate peak remains the same.

10.3 Bandwidth Allocation

In order to make use of the statistical multiplexing gain shown in Section 9.5, we need to devise a suitable method of bandwidth allocation.

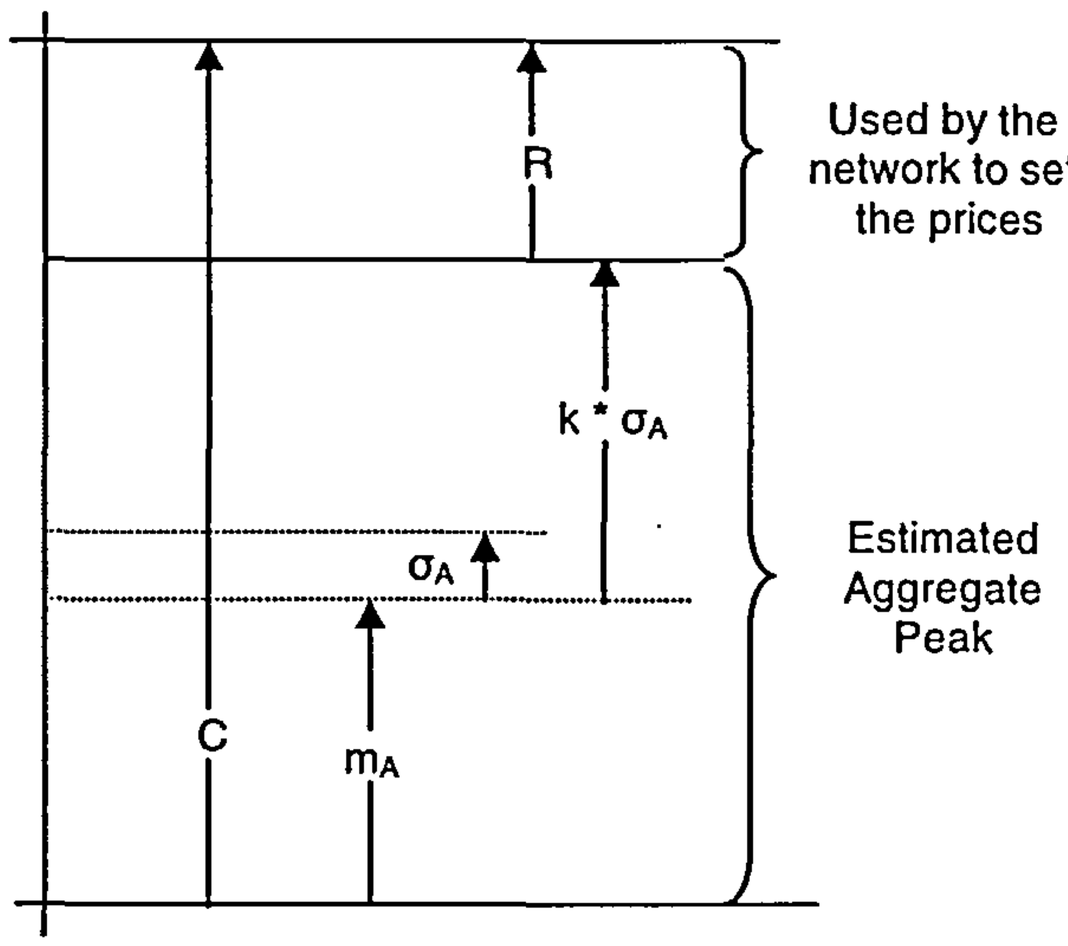

Figure 10.1: Bandwidth Allocation at the Aggregating Link

We use the model shown in Figure 10.1. The total capacity of the link is shown by C. Consider that n flows each indexed by i are aggregating at this link. We can obtain aggregate mean (m_A) and aggregate variance (v_A) by metering the arrival processes at the ingress router output port, see Figure 9.4. As the aggregate mean is the sum of mean values of each flow and the aggregate variance is the sum of flow variances, there is a direct relationship between the number of flows and their mean and variance and the mean and variance of the aggregate traffic. This is indeed the basis of our pricing method. We then estimate the aggregate peak using Eq 9.4, rewritten here as:

$$Estimated\ Aggregate\ Peak = m_A + k\sqrt{v_A} \qquad \textbf{Eq 10.3}$$

Rewriting Eq 9.5,

$$Aggregate\ peak = \sum_{i=1}^{n} m_i + k\sqrt{\sum_{i=1}^{n} v_i} \qquad \textbf{Eq 10.4}$$

From Eqs. 10.3 and 10.4, we can deduce that the change in mean or variance of a single flow leads to a related change in the aggregate mean and variance respectively and hence, by pricing mean and variance, we can fairly charge each flow for its contribution to the load or congestion.

10.4 Price Calculation

We found that a satisfactory way to set a meaningful price at the aggregation point was best understood as allowing the network operator to bid for the spare bandwidth at the link. Assume a network operator has a fixed number of tokens called "network bid" (WtP_{net}) that will be used to bid for the spare capacity and thus set the prices. The spare capacity (R) of the link is given by:

$$R = C - (m_A + k * \sigma_A) \qquad \textbf{Eq 10.5}$$

As more flows become active, the aggregate peak increases and R will decrease leading to increase in prices. It is as though the network operator itself has a WtP for all available bandwidth (similar to an elastic flow). .

10.4.1 Mean Price (P_m)

It is relatively straightforward to visualise how the mean price should be calculated. If the aggregate mean (m_A) increases, the spare capacity, R, will decrease proportionally and hence the price should decrease in the same proportion.

$$\frac{dR}{dm_A} = -1 \qquad \textbf{Eq 10.6}$$

$$\therefore P_m = \frac{WtP_{net}}{R} \qquad \textbf{Eq 10.7}$$

10.4.2 Variance Price (P_v)

Calculating the variance price is more difficult. We know that an increase in aggregate standard deviation (σ_A) will decrease the spare capacity linearly. We use this to derive the relationship between aggregate variance and spare capacity.

$$\frac{dR}{d\sigma_A} = -k \Rightarrow \frac{dR}{dv_A} = -\frac{k}{2\sigma_A} \qquad \textbf{Eq 10.8}$$

$$\therefore P_v = \frac{WtP_{net}}{R} * \frac{k}{2\sigma_A} \qquad \textbf{Eq 10.9}$$

The variance price equation has a σ_A term in the denominator, which means that if the aggregate variance is zero, the variance price will shoot to infinity. This is undesirable as the aggregate variance may often become zero or very small when, for example, there

are few or no flows active, or when the system starts and the metering process has not generated a result. Due to this, we need an initial variance that the system can assume to have in order to set the prices when no variance is metered.

A numerical test was carried out to validate that the pricing algorithm was fair as specified in our objectives (Section 10.2). The details can be found in Appendix V.

10.5 Adaptive Flow and Pricing

For the purpose of this study, we will focus on only the non-persistent adaptive flows and the term "flow" will mean this type of flow from now on unless otherwise stated. The iterative feedback control loop of pricing and corresponding changes made by the flow is shown in Figure 10.2.

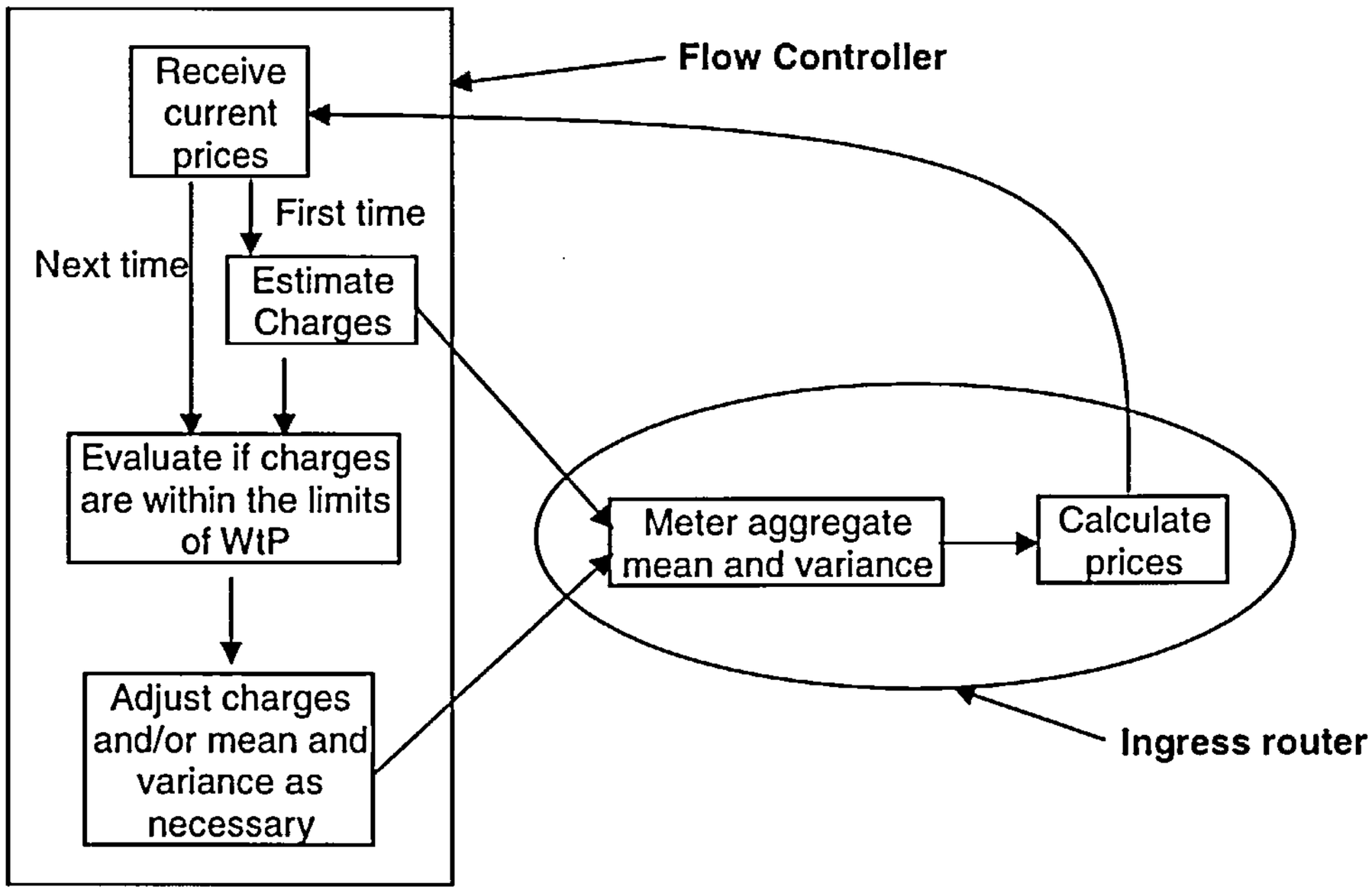

Figure 10.2: Price Based Feedback Control System

When the i^{th} flow wants to join in, its controller receives the current prices. Using its own requirements of mean (m_i) and variance (v_i), it works out the required charges for mean (Ch_{mi}) and variance (Ch_{vi}) using the following equations:

$$Ch_{mi} = m_i * P_m \qquad \textbf{Eq 10.10}$$

$$Ch_{vi} = v_i * P_v \qquad \textbf{Eq 10.11}$$

Each flow controller periodically receives a pair of prices and has the option of re-allocating its distribution of tokens between mean and variance according to its priority[13]. For example, consider that the priority is given to the mean. When the prices are updated, the flow controller recalculates the charge required. If the total charge exceeds the maximum WtP, then it will have to reduce the charge allocated for the variance until one of two things happen: the total charge becomes equal to the WtP, or the variance charge reaches a value that can only "buy" the minimum acceptable variance. In the latter case, the mean charge is reduced until the total charge becomes equal to the WtP or it reaches a value that can "buy" the minimum acceptable mean. If the total charge is still higher than the WtP, the flow will have to cease[14], but otherwise, the flow will transmit according to the new mean and variance.

10.6 Fluid Flow Simulation

The pricing algorithm had to be first tested for the fluid flow system that does not have the complexity caused by packet transmission. This was followed by a full packet level simulation. In the fluid flow simulation, it was assumed that the flow monitors are capable

[13] It is possible for the operator to assign a priority to the mean or variance for each flow that it handles. Some flows may be tolerant to smoothing (reduction in variance) wheras others may perform better if the bit rate was reduced while maintaining the burstiness (reduction in mean).
[14] We have assumed that the flows are non-persistent. Persistent flows will behave inelastically to preserve their minimum mean and variance.

of changing the bit rate of the flow such that it has the exact mean and variance as instructed by the control system. Another assumption was that propagation delays across the network were negligible because while it is possible to reduce queuing delays, one cannot mitigate the effect of the propagation delays that occur due to physical distances. The objective of this exercise was to ascertain that assuming that all the components co-operate, the pricing method is fair, scalable and capable of minimising queuing delays. The network model that we used is shown in Section 9.7. It was adapted for a fluid flow simulation. The description follows in the next sub-section.

10.6.1 Fluid Flow Model

The simulation model constructed in OPNET™ consisted of 18 flows and a single aggregation point. All the flows were identical and each had a Preferred mean rate = 10 Mbits/s, Preferred variance = 64 $Mbits^2/s^2$, max WtP = 20 and Priority to Mean (i.e. variance will be reduced first). Three flows are activated at the start of the simulation; after that the flows become active one at a time at 100s intervals and then they stop transmitting one at a time after 1600s. One of the first flows remains active throughout the simulation, which was run for over 2000s. The node was configured such that the Capacity = 200Mbits/s, k = 5, WtP_{net} = 40 and initial variance = 100. The parameters were chosen arbitrarily to cause congestion so that the response of the control system could be observed.

The flow has a maximum WtP and shapes the traffic according to the prices received from the network using the algorithms described earlier in this chapter. The prices were broadcasted by the bottleneck link. In order to test the performance of the control method, we have knowingly induced instability by making all the flows behave identically except for the different starting times. For example, when the total charge that a flow incurs exceeds its WtP, it will reduce the lower priority attribute, the variance in this case. In this

example, all the other flows will also simultaneously reduce their variance. This could reduce the aggregate peak leading to a sharp decrease in prices, which in turn would lead to flows increasing the variance back towards normal. This represents the worst-case scenario as all the flows are synchronised and the method needs to include some form of damping.

A number of preliminary tests were carried out to improve the algorithm. Some of the major improvements are discussed below.

10.6.2 Stability

When the prices are communicated back to the flows, the flow controllers immediately take action. While this is desirable, it can lead to oscillations. Therefore, the change in prices communicated back to the sources was damped. That is, if the price decreased from P to P', the price that was sent to the source would be $P + \beta(P' - P)$, where β is the damping factor and $0 < \beta \leq 1$. Obviously, this implies that flows reduce their rate more slowly than it may be required but since we overestimate the aggregate peak by using $m_A + 5*\sigma_A$, it is highly unlikely that the actual peak rate will exceed this value.

10.6.3 Network's WtP and Aggregate Peak

It was found that the WtPnet plays an important role in setting this margin between the operating capacity and the aggregate peak. The higher the value of WtPnet, the bigger is the difference between the controlled level of aggregate peak and the capacity. With very small values of WtPnet it is possible for the aggregate peak to exceed the capacity temporarily (during the time when a new flow joins and prices are stabilised). This will normally be a network manager's job to decide how much is a judicious margin. Again, it has to be remembered that we are always overestimating by using $m_A + 5*\sigma_A$ and hence WtPnet can be set to a low value. In our experiments, we found that WtP_{net} between $1/20^{th}$

and $1/5^{th}$ of the total WtP was acceptable. The total WtP of the flows can be a limit imposed by the network operator.

10.6.4 Floor Prices

Since the variance price calculation depends on the standard deviation of the aggregate, the result of equation shown in Eq. 10.9 will tend to infinity when the link is empty. The network, therefore, has to pretend that there is some variance when there are no flows active in order to ascertain the prices for the first flow to join. The prices based on an assumed low variance can effectively act as floor price, giving a lower bound to the price range, and can also be used when the metered variance of the flows at the aggregate is very small. This can be configured by the network operator.

10.6.5 Results of the Fluid Model

Figure 10.3 shows the pricing mechanism in action. As more flows join in, the spare capacity decreases and the mean and variance prices increase. Without the control system in place, the estimated aggregate peak ($m_A + 5*\sigma_A$) would have exceeded the capacity at 600s when 9 flows become active each with mean rate of 10 Mbits/s and variance of 64 $Mbits^2/s^2$. However, with the control method, the system starts to throttle back the flows earlier at 400s because the network is bidding for the spare capacity. As a result, the aggregate peak always remains below the total capacity, 200 Mbits/s. The spikes in the aggregate peak, and therefore in the prices, are due to the time it takes for the prices to take any effect on the flows.

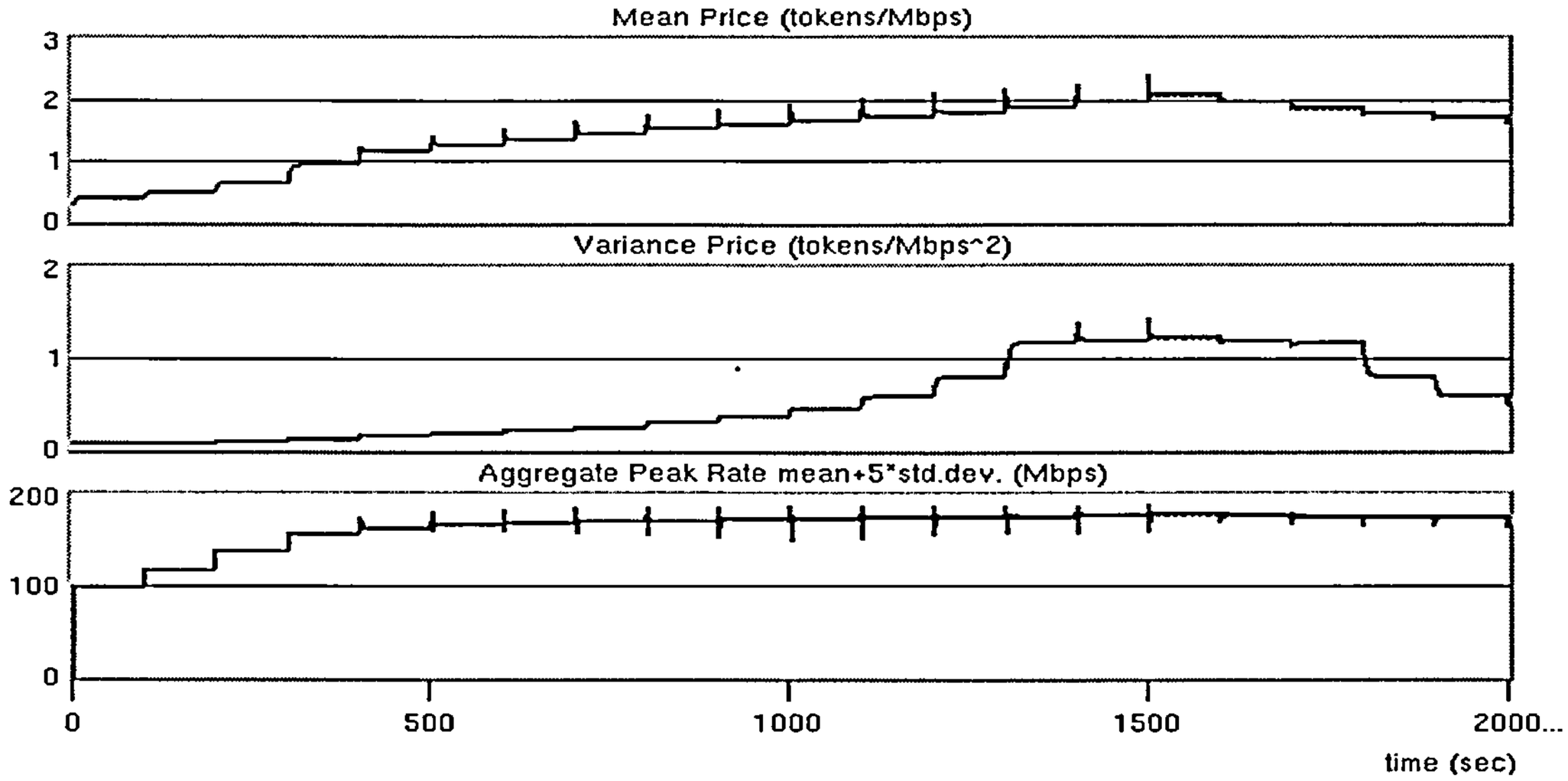

Figure 10.3: Prices calculated at the Ingress node and estimated aggregate peak

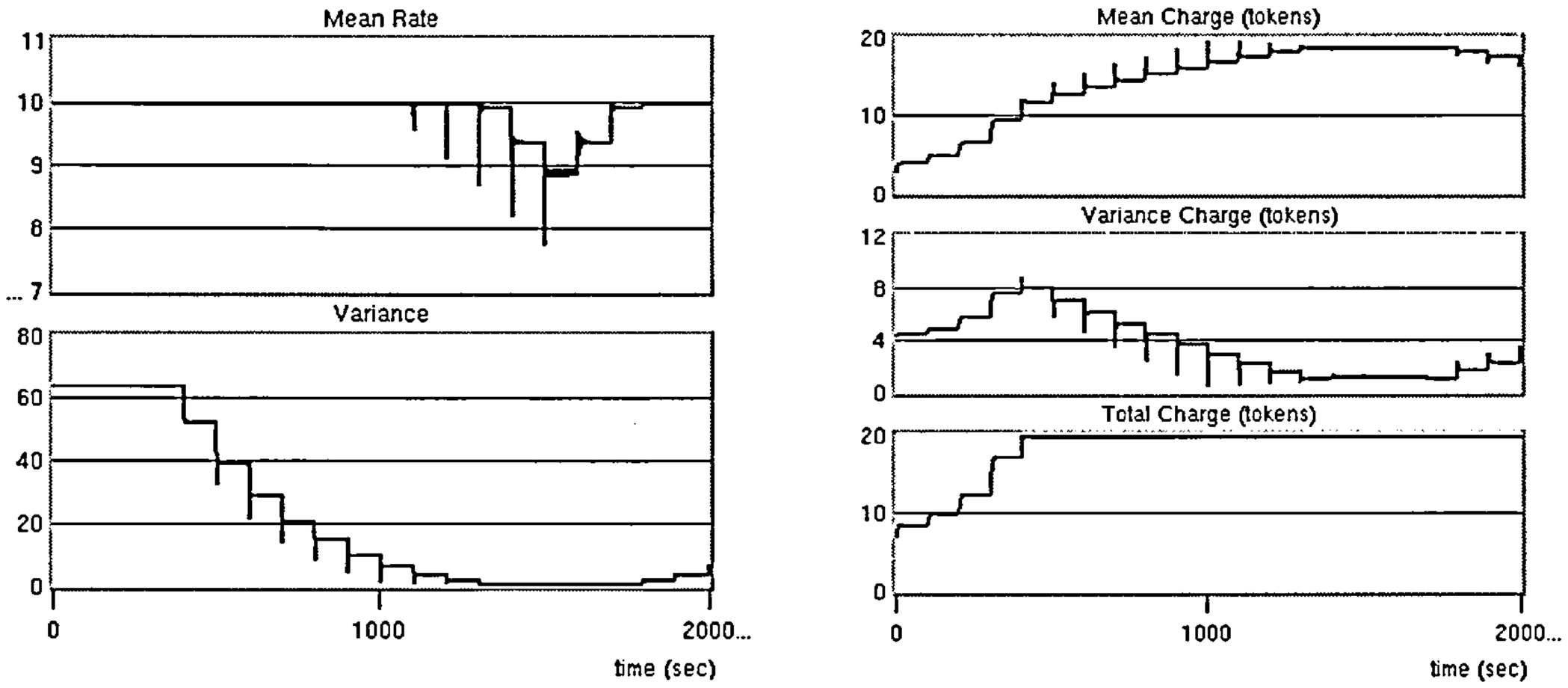

Figure 10.4: Mean and Variance and Charges incurred for the first flow

The effect of price feedback on a flow is shown in Figure 10.4. The graphs show the changes in mean variance and the respective charges incurred by the first flow. As the prices increase, the charges incurred for mean and variance increase. At about 400s, prices are high enough that the total charge reaches the limit (max WtP = 20 tokens). In order to ensure that the total charge does not exceed this limit, the flow reduces variance (lower priority attribute). The higher priority attribute, mean, remains unaffected until 1100s when

it is decreased as the variance drops to the minimum acceptable value. The downward spikes in the mean are again due to the time taken for the prices to become effective.

10.7 Packet Level Simulation

The experiments using the fluid flow model were used to refine the pricing algorithm and they showed that the method produced a stable response even in the unrealistic scenario of all the flows synchronised in terms of their requirements and priorities. To demonstrate the applicability of the mean and variance pricing control method to real networks it was decided to carry out more realistic simulations in OPNET Modeler™.

10.7.1 Differences from Fluid Flow Model

The network model used for the packet level simulation is shown in Figure 9.4. The simulation at packet level adds some complexity. The flow controller is now required to ensure that the packets from the generator are forwarded in such a way that the mean and variance of the outgoing flow comply with the values that it can afford, given the WtP. In the fluid flow model, it was assumed that the notification and response were immediate and exact. For example, when a flow received the prices it would adjust its mean and variance immediately and correctly so that there were no oscillations. In reality, the usual artefacts of the control process result in oscillations. For example, the flow controller can calculate the exact mean and variance it should use but there may be some delay before the new values are stabilised. Also, there would be an effect on the network due to the response made by the flow, which will again change the prices. Another problem is that the mean and variance of the flow may be changing over short timescales, causing more fluctuations. Therefore, in packet-based simulations, the prices were calculated periodically (the frequency can be assigned by the user). This meant that the system would periodically

measure the mean and variance of traffic arriving at the node and would base the price calculations on the traffic parameters of the last interval's measurements. The prices would be communicated back and flows would use their measurements over the last interval to estimate how much they need to pay for the next interval. So, the flow controller assumes that the mean and variance of traffic arriving from the packet generator will not change significantly between successive calculation intervals. In the model, the intervals are kept to 1 second for all periodic calculations. Further details of the simulation model are given in the following sub-sections.

10.7.2 Simulation Model Details

Although the pricing mechanism is envisaged for use in carrier networks dealing with aggregates of thousands of flows, it should also work well with individual flows. A network of 7 flows aggregating at a single ingress router output port modelled by a queue and some auxiliary processes were used. In reality, the ingress router will forward the data on to one or more routers. However, we use Backward Congestion Notification so that the price feedbacks are sent directly from the node to the flows as a broadcast signal, and hence the subsequent routers are not required. We now look at each component of the model in detail.

10.7.2.1 Packet Generator

The model now consists of 7 packet generators, one for each flow. The packet generators are similar to the ones used earlier for experiments with percentile monitoring in that they used generalised exponential function for inter-arrival delay. One difference is that the packet lengths are much shorter in order to comply with Ethernet standards.[15] The lengths,

[15] Ethernet packets have a maximum size of 1514 bytes.

as before, had a tri-modal distribution. The mean delay and squared coefficient of variance are two parameters used by the generalised exponential algorithm that calculates the delay between successive packets. Processes were parameterised so that flows with varying degrees of burstiness could be generated.

10.7.2.2 Flow Controller

The flow controller implements the adaptive behaviour and policing of the flow described in Section 10.5. In the fluid flow simulation, it was possible to have straightforward control on the mean and variance of a flow output. In packet-level simulation, it was decided that the flow controller would use an absolute rate controller along with a meter and a target calculator. The processes of the flow controller are shown in Figure 10.5.

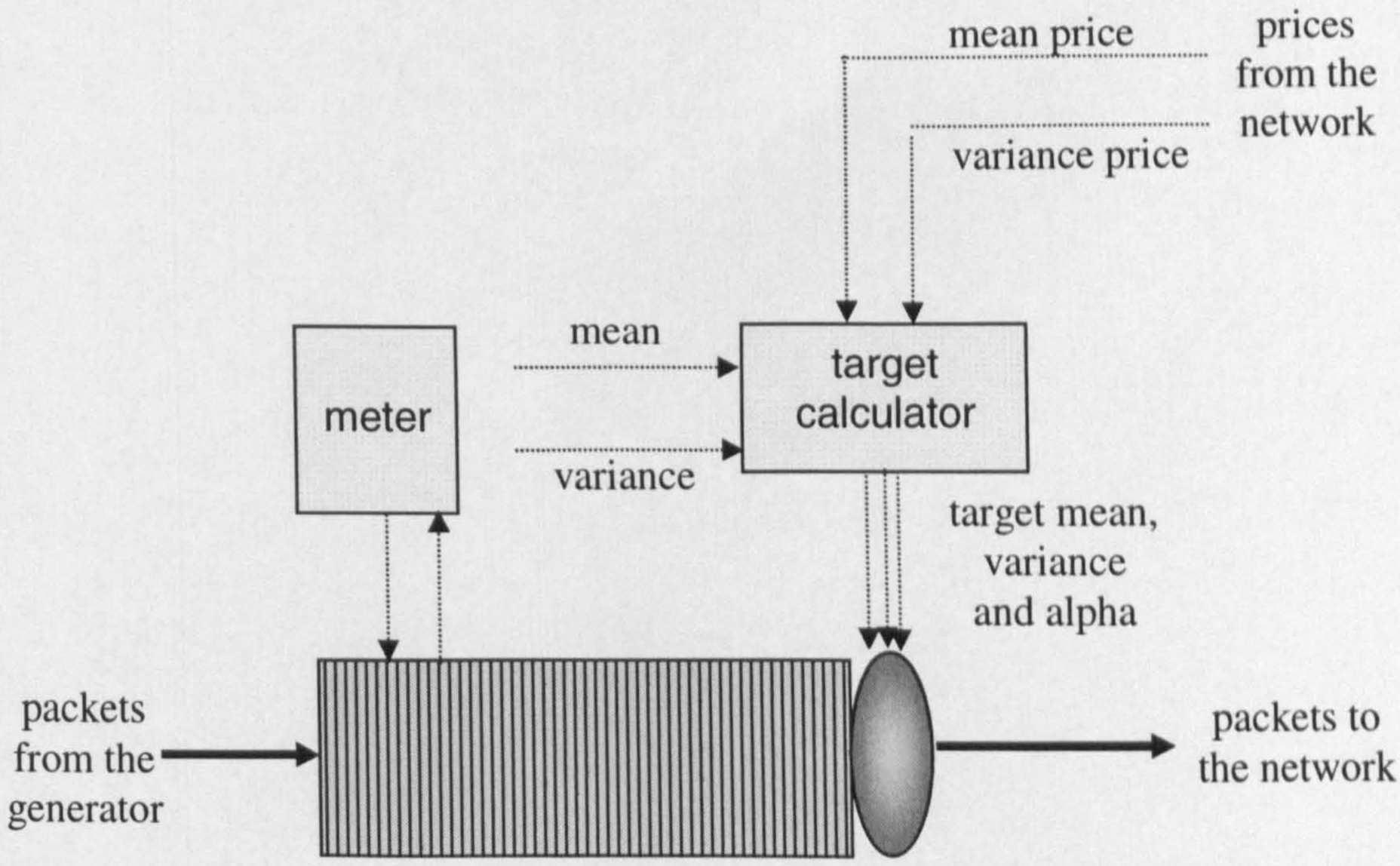

Figure 10.5: Flow Controller with Metering and Rate Controller

The meter process periodically reads the number of bits arriving at the queue. The frequency of making this measurement is higher than the frequency of calculation. Hence,

in the model a measurement is taken every 5ms to calculate the mean and variance every 1s (from 200 samples), see Appendix IV for further details. The mean and variance are then passed on to the target calculator every 1s. The target calculator works out the charges required to pay for the mean and variance according to the prices it receives from the aggregating node using the Eqs. 10.10 and 10.11. If the total charge is equal to the WtP of the flow, the target mean and the target variance are assigned as equal to the mean and variance respectively. If the total charge exceeds the WtP, then the charge used for the lower priority attribute is decremented by the excess so that the total charge is now equal to the WtP, unless the minimum charge has been reached for this attribute. Minimum charges are the charges required to pay for the minimum accepted value of mean or variance at the current price. For example, if the variance had lower priority but if the variance charge was already reduced to the minimum acceptable, then the mean charge will be reduced in order to ensure that the WtP is not exceeded.

If the total charge is less than the WtP, then the charge for a higher priority attribute is incremented until the maximum value can be achieved; any surplus WtP is then used to increase the charge for the other attribute. The maximum value is the metered value of the generated traffic flow before any adaptations are made. Since the calculation of charge begins at the start of the simulation, the initial values for mean and variance of the flow have to be specified so that they can be used before the measurements from the metering process are obtained. The algorithm also ensures that the rounding errors in the calculations does not imapir the decision. After the mean and variance charges have been finalised, the target values are calculated using the following equations:

$$target\ mean = \frac{Ch_{mi}}{P_m} \quad \& \quad target\ variance = \frac{Ch_{vi}}{P_v} \qquad \textbf{Eq 10.12}$$

In addition to the target value calculation, this process also calculates the value of α. The minimum value of α is 1 and the maximum, α_{max}, is specified at the simulation time, say 10. The value changes according to the WtP that was surplus, i.e., the total charge required for the mean and variance was less than the WtP of the flow. The value of α is calculated as:

$$\alpha = 1 + \left(\frac{\alpha_{\max} - 1}{WtP} \right) * (WtP - Total\ Charge) \qquad \textbf{Eq 10.13}$$

The importance of α will become clear later in this Section in the discussion of the queue process.

The queue process serves the packets using an absolute rate controller. The rate controller is suitable for flows tolerant to smoothing, whose priority is to maintain the mean rate. It has to be noted that although the system potentially has the option to prioritise variance, a special controller will be required to control the flow such that it has a low mean and high variance. Due to constraints of time, it was important to validate the pricing concept using the available controller before developing a special one.

The operation of the absolute rate controller is to serve the traffic arriving at the flow controller using a service rate, which was calculated as shown in Eq. 10.14.

$$service\ rate = target\ mean + k * target\ variance \qquad \textbf{Eq 10.14}$$

Through some preliminary experiments it was found that in fact this could cause a too stringent control on the packet flow causing them to queue up even when the load in the network is light. Therefore, it was enhanced to the following:

$$service\ rate = \alpha(target\ mean + k * target\ variance) \qquad \textbf{Eq 10.15}$$

If in the last interval the flow used only a small proportion of its total WtP, due to its low requirements, it is allowed to send at a proportionally higher service rate than it would be for given requirements and prices (due to a higher value of α). It was found that this avoids unnecessary queuing at the flow controllers during light load conditions. Under heavy load conditions, the prices become high and the flows have to use all (or almost all) of the WtP and then the value of α drops to 1. The target calculator process then passes these parameters to the queue process for it to calculate and implement the required service rate. The α behaves like a credit system where a user who saves credits is allowed to use them in future. Hence, if a flow is not using all its WtP it is served at a higher rate than required for its target values.

10.7.2.3 Price Calculator

The price calculation takes place at the output queues of the ingress router to the carrier network. The ingress router is assumed to be capable of multi-service with separate queues for each class of traffic, which are isolated from each other through bandwidth partitioning. The processes involved are shown in Figure 10.6. Here, we are concerned with only one class of traffic and hence, only one queue is shown. The packets from all the flows arrive at the queue. A meter process uses a sample method similar to the one used in the Flow Controller in order to calculate the mean and variance of the aggregated traffic and sends the values to the Price calculator process. From that, an estimate of the aggregate peak is derived and prices for the mean and variance are calculated for the given queue capacity. In the model, the prices are broadcasted back to the flows. If the flows are not active they will simply ignore the prices. Alternatively, direct signaling could be used. The choice depends on the type and topology of the network as well as the constraints of overheads.

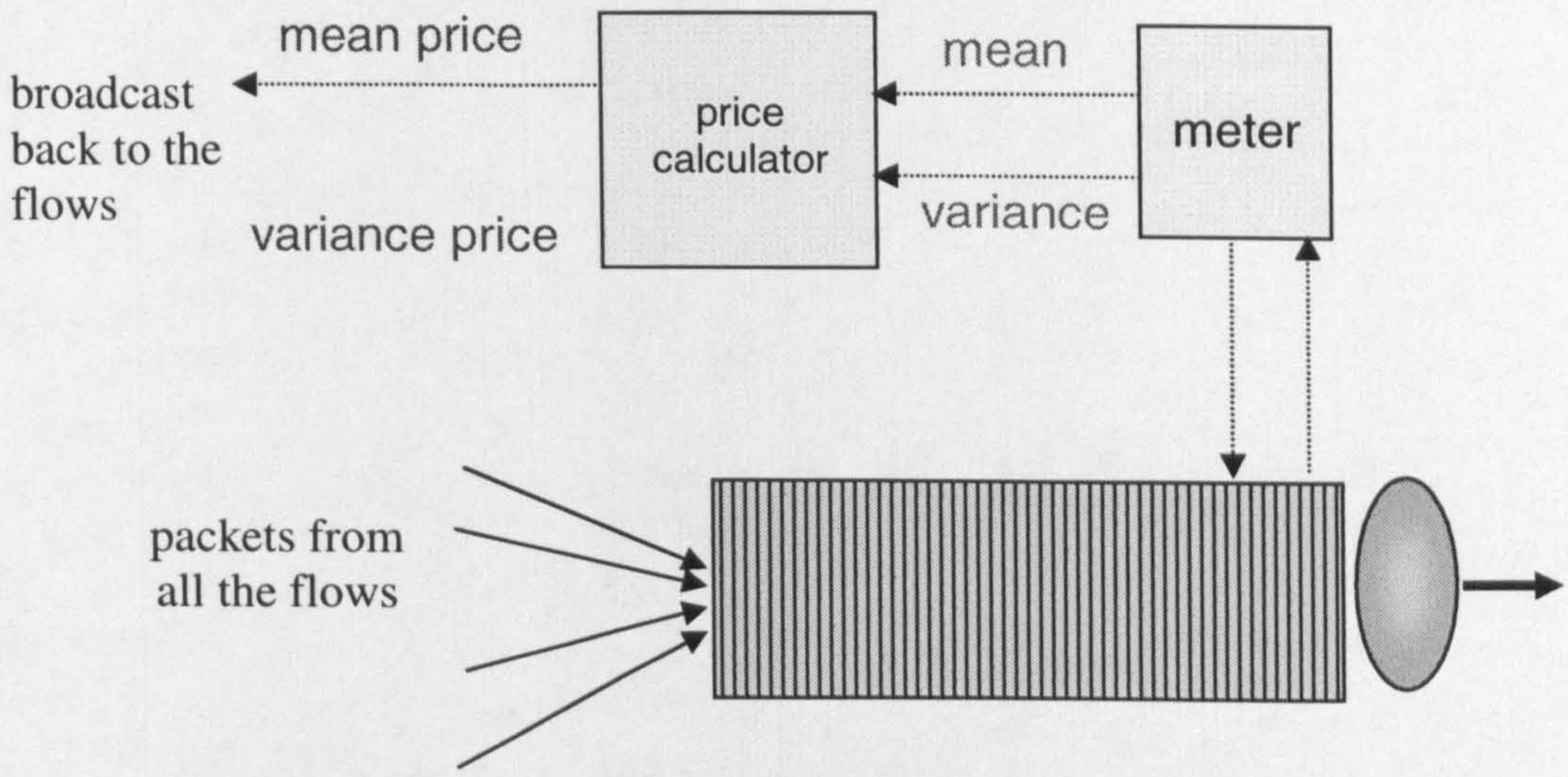

Figure 10.6: Price Calculator at the Ingress to the Carrier Network

The issue of price stability was addressed earlier in Section 10.6.2. With periodic price calculation and a dynamically changing network, another issue of stability was highlighted. It turned out that the prices could fall significantly if mean and variance measured over a given interval were low. We found that in a dynamic system like the one modelled in our experiments, such instances would be inevitable and the resulting drop in price would cause unstable behaviour, as the flows would receive occasional low prices even during periods of heavy load. In order to solve this, a 9-point moving average of the estimated peak was maintained and used to derive another pair of mean and variance prices, which would be smoother. Typically the instantaneous prices would be calculated every second and the smoother prices would be calculated every 10s. At any time the prices used to communicate back to the flows would be the higher value of the two. This allowed the prices to rise rapidly (with one feedback) but fall slowly (over ten feedbacks). This type of flow control is similar to the well-known Additive Increase Multiplicative Decrease method used in TCP protocols.

10.7.2.4 Destination

In real network scenarios, the packets from the ingress router would be forwarded to one or more routers in the carrier network and then on to some client network until it eventually reaches the end-user. However, since the concern here is congestion avoidance at the ingress router and it is assumed that core routers of the carrier network are high-speed and do not induce delay, in the model, the packets are simply destroyed as they do not affect the pricing mechanism. It is worth noting that this is different from the problem described in Section 9.1 where the DRC mechanism requires prices for the path across the carrier network.

10.7.3 Simulation Scenario

The model consisted of 7 flows, each with some differences in their parameters settings such as squared coefficient of variance (SCV) for inter-arrival delay between successive packets, start time etc. The simulation was run for 600s. A number of runs were carried out in order to verify the model behaviour but, due to the limitations of space, only the ones that demonstrate the performance of the entire control system are presented here. The flow attributes used for the simulation are summarised in Table 10.1.

Table 10.1: Flow Parameters

Flow	Squared Coefficient of Variance	Willingness to Pay (tokens)	Start time (s)
1	1	6000	20
2	3	6000	60
3	3	7000	100
4	3	8000	140
5	3	9000	180
6	3	10000	220
7	10	50000	260

The capacity of the ingress router output queue (see Figure 9.4) was fixed at 3 Mbit/s and this was the same as its service rate. The purpose of the experiment was to

evaluate the performance of the control system when the estimated aggregate peak would require about 80% of the bandwidth, which is when congestion is a problem. Although in practice the system would be used for very high capacity networks, in the order of Gigabits, the mechanism could be illustrated well with a scaled down version that would be easier to model in OPNET™. The estimated aggregate peak of our flows is about 2.5 Mbit/s (shown later in Figure 10.9), which is approximately 80% of the capacity.

The simulation starts with no flows being active. This will show the floor prices in action. At time 20s, the first flow becomes active. It is a relatively smooth flow with a low WtP. Then more flows join in during the course of the simulation. These flows are slightly more bursty due to their SCV being set to 3. They are also of progressively higher value with increasing WtP. The last flow to activate is extremely bursty and has a very high WtP.

We have some expectations from the system:

- During light load conditions, the flows should be allowed to send traffic without unnecessary delay.
- As the network gets congested, the flows with low WtP should be the first to be affected.
- A flow with small mean but high variance and high WtP must not be delayed even in heavy load conditions until all the other flows have been throttled back.
- All the queuing due to throttling back of flows should occur at the flow controllers and not at the ingress router output queue.

The results are presented in the following section but before that we look at the behaviour of some of the flows that we shall be comparing[16]. The mean and peak rates of flows

[16] All the graphs were plotted with calculations being executed every second.

depending upon their SCV values without any influence of control system are shown: Figure 10.7 shows the flow 1 and 2 while flows 6 and 7 are shown in Figure 10.8. Mean rate is defined as number of bits arriving over unit time (1s) and the peak rate is defined as mean + 5√(variance measured using 5ms samples), see Appendix IV.

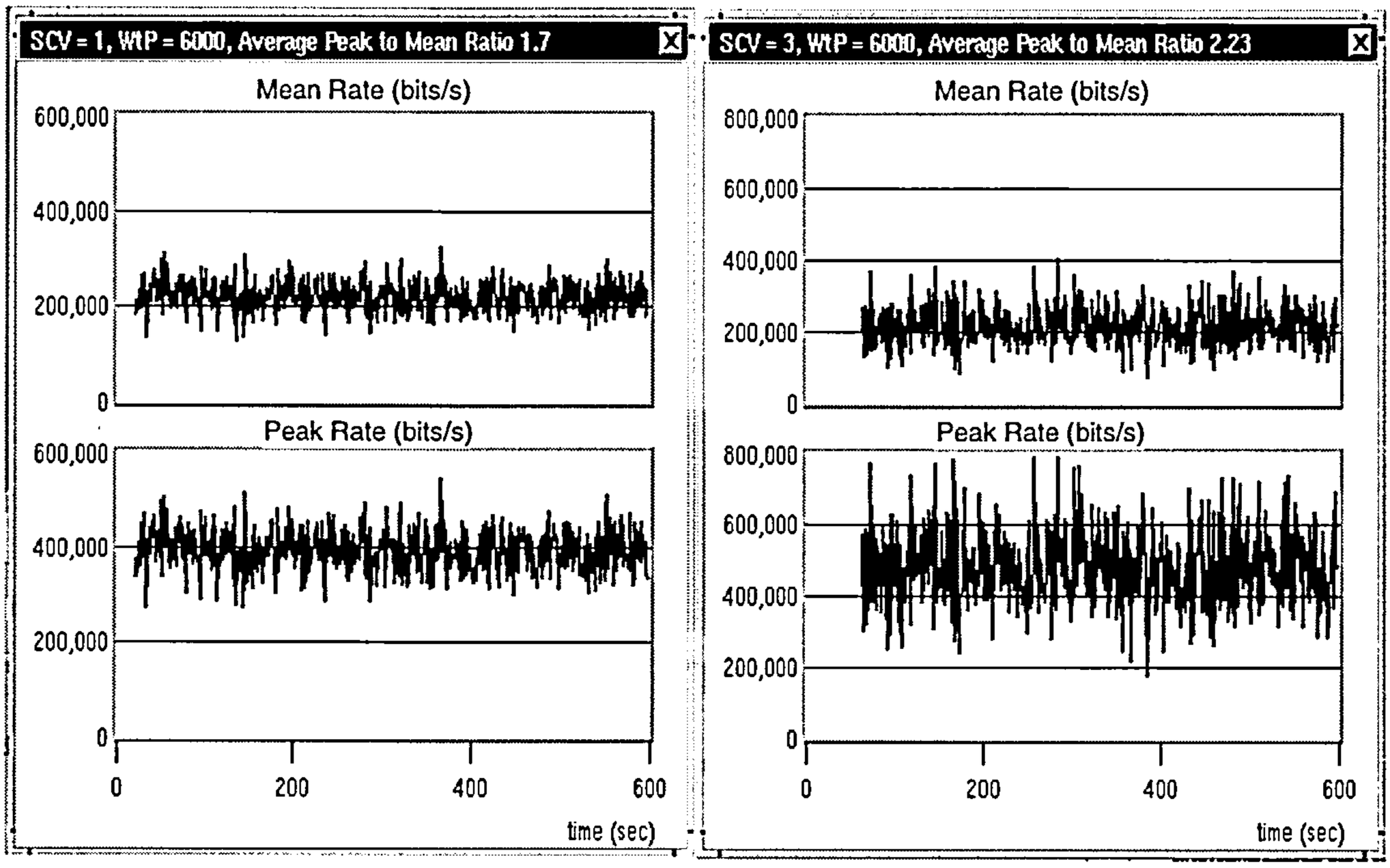

Figure 10.7: Flows with low WtP and low and medium burstiness

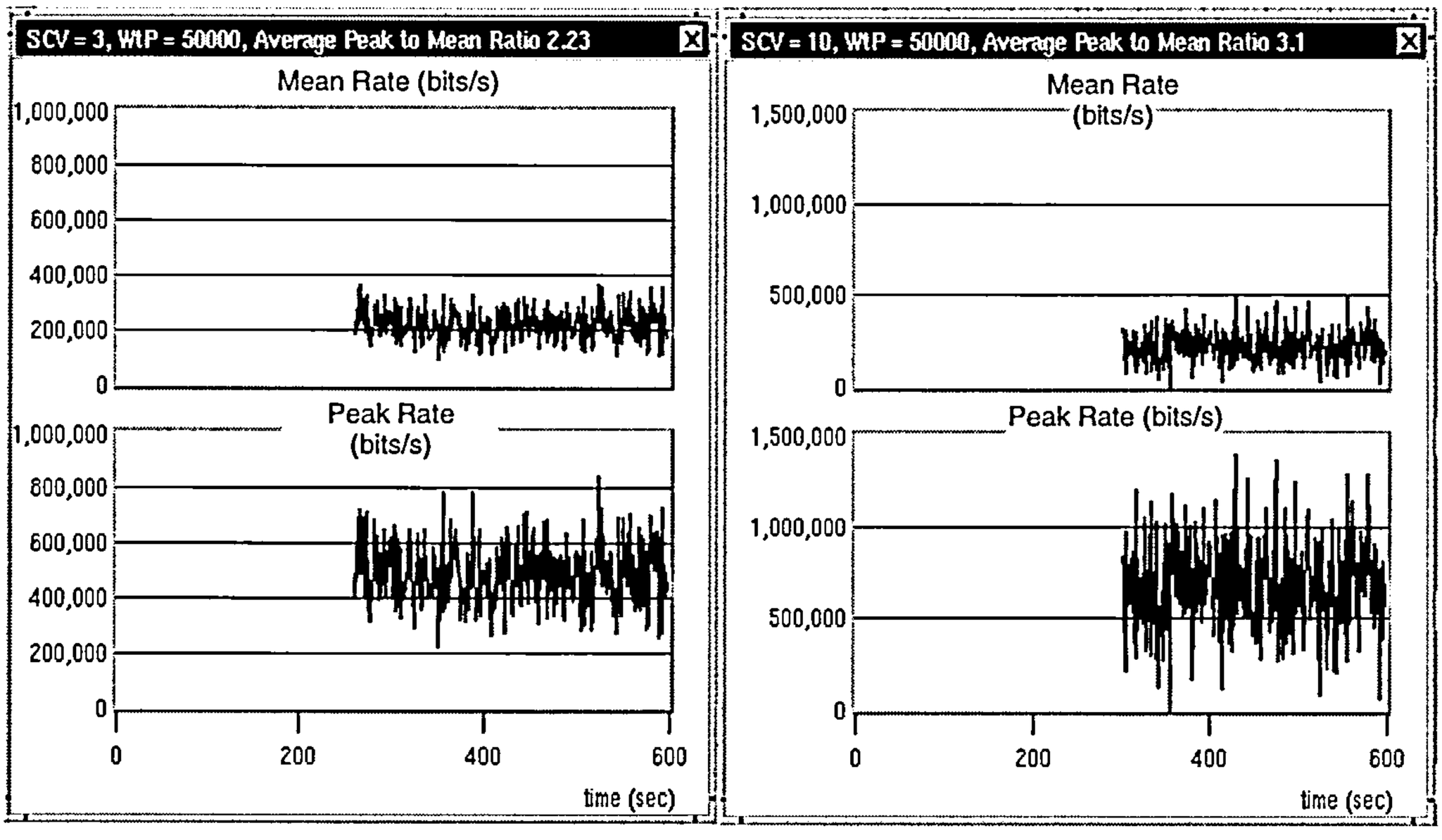

Figure 10.8: Flows with high WtP and medium and high burstiness

10.7.4 Results

The network behaviour and the corresponding response of the flows were observed and evaluated against the expectations listed earlier in this chapter. As far as possible, the graphs presented for comparison are shown at the same scales. The flows were started one at a time as shown in Table 10.1.

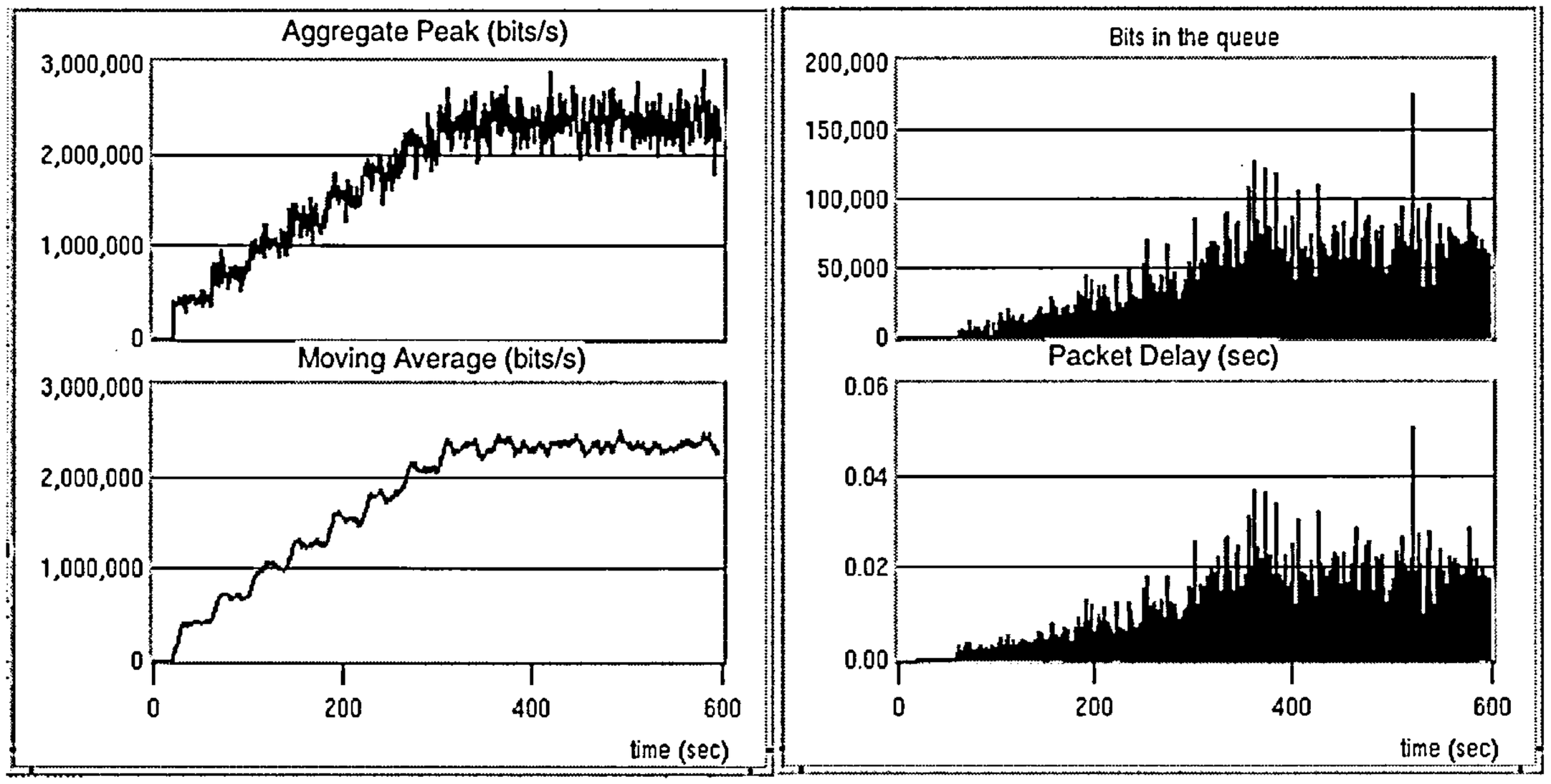

Figure 10.9: Ingress router output queue response

Figure 10.9 shows the estimated aggregate peak at the ingress router output queue and its moving average along with the instantaneous measure of bits in the queue recorded when a packet leaves the queue and the delay that an arriving packet would encounter. The maximum delay has an occasional spike of 0.05s. The 99-percentile delay was measured to be 0.0163s. This result shows that the control system was working effectively. Normally, this scheme would be used in a higher capacity system, where the capacity would be much larger than 3 Mbit/s and therefore the delays would be considerably shorter. Also, in the experiment the control capacity and queue service rate were the same. In practice, usually

the control capacity would be set at a lower rate than the service rate to create a safe buffer zone.

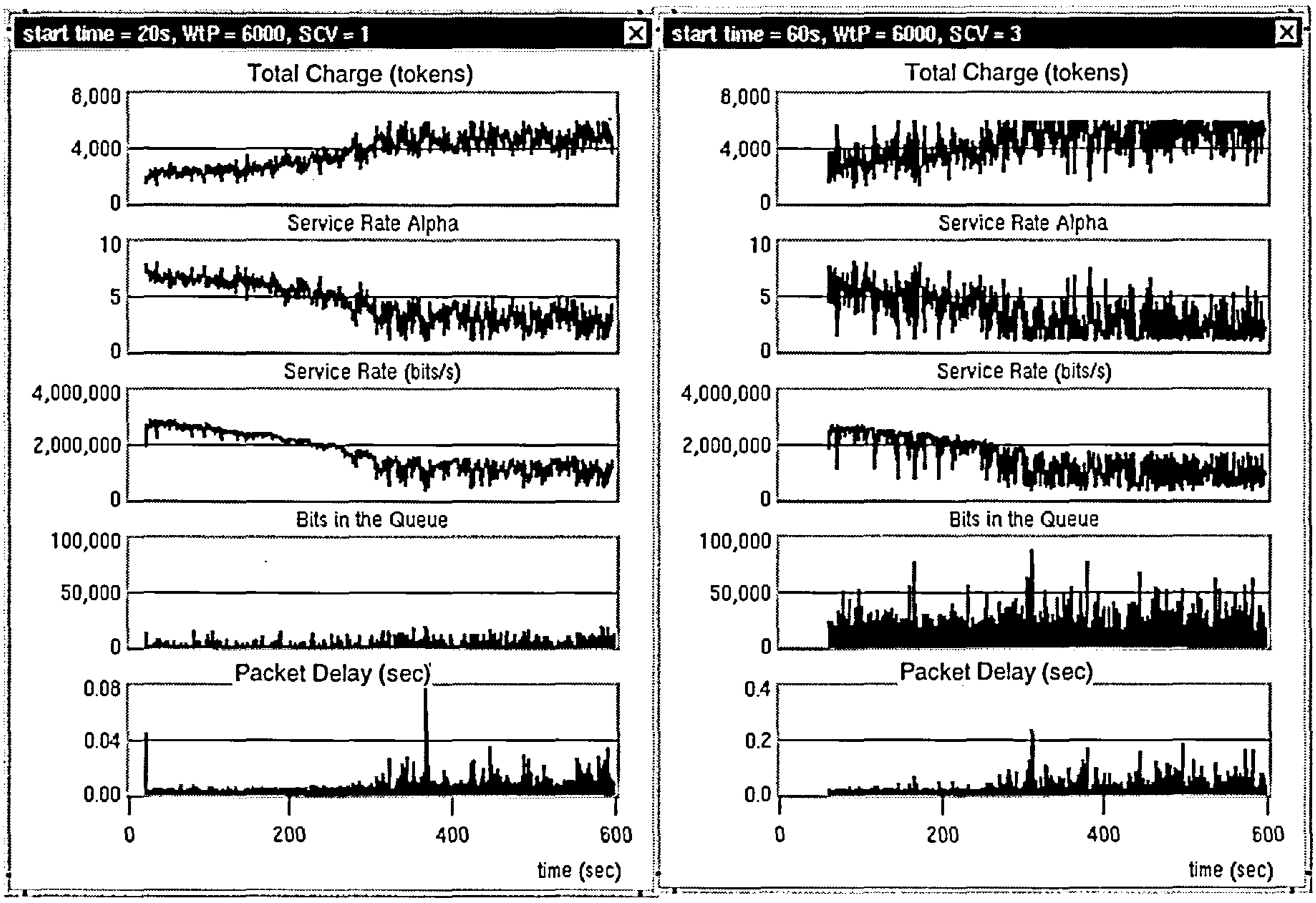

Figure 10.10: Controlled response of flows with low WtP and different burstiness

The flows shown in Figure 10.10 both have the same WtP but different SCV, (flows 1 and 2 in Table 10.1). As expected during the light load (first 400 seconds), both the flows have high service rate due to the high value of alpha. As the load increases, the more bursty flows starts using up all of its WtP (indicated by clipping in the graph labelled total charge) and therefore has to reduce its service rate significantly and incur significantly high delays in the order of 0.1 to 0.2s at the rate controller. The smoother flow on the other hand experiences a maximum delay of 0.08s.

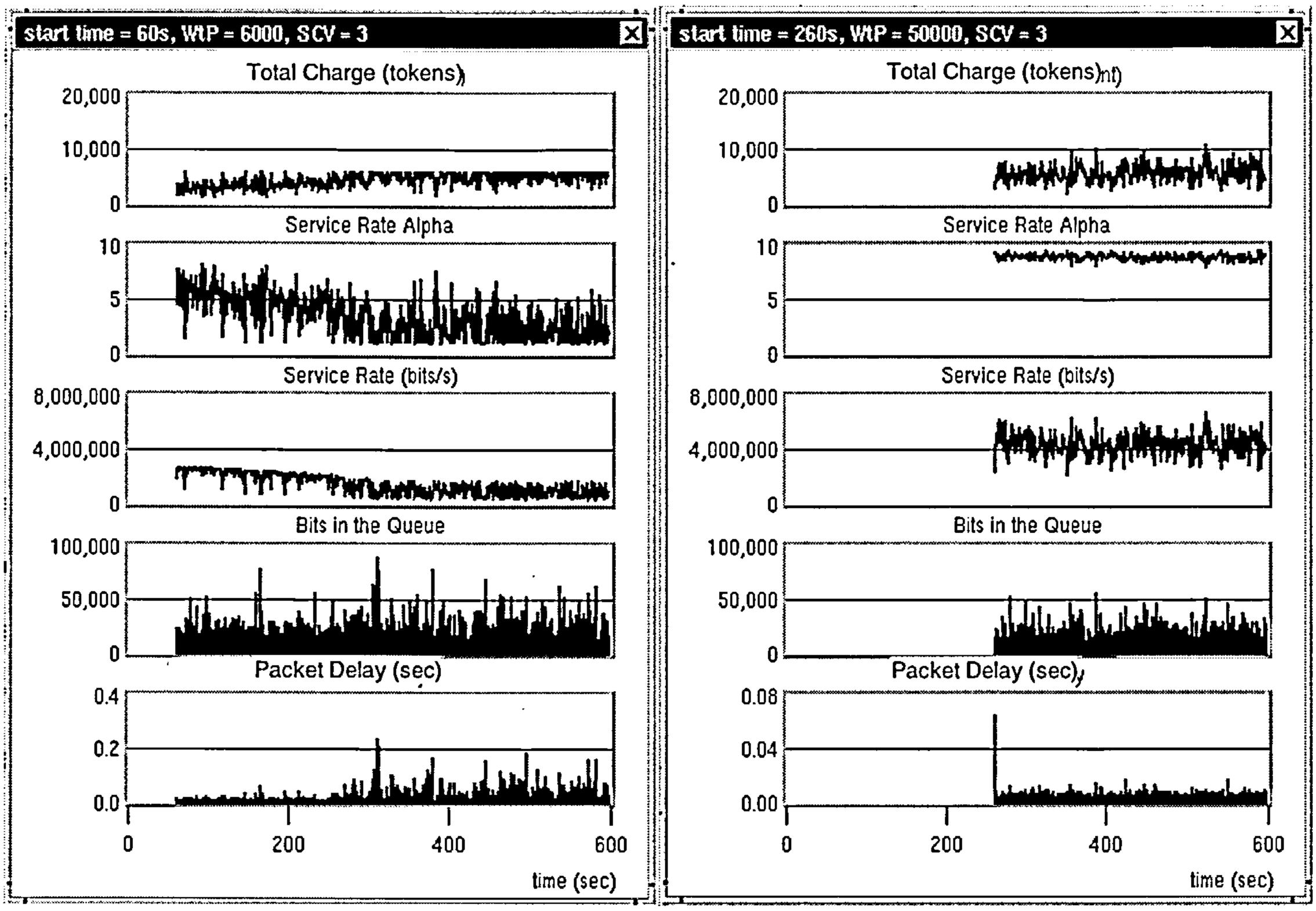

Figure 10.11: Controlled response of flows with different WtP but same burstiness

The flows shown in Figure 10.11 have the same SCV and hence the same degree of burstiness, but they differ in their value of the WtP. The flow on the right with WtP of 50000 is clearly able to send its data even as the load on the network increases. The value of α is consistently high and the packets experience delay in the order of 0.01s (after the initial heavy transient delay) whereas the flow with less WtP (6000) suffers a packet delay in the order of 0.2 to 0.4 seconds.

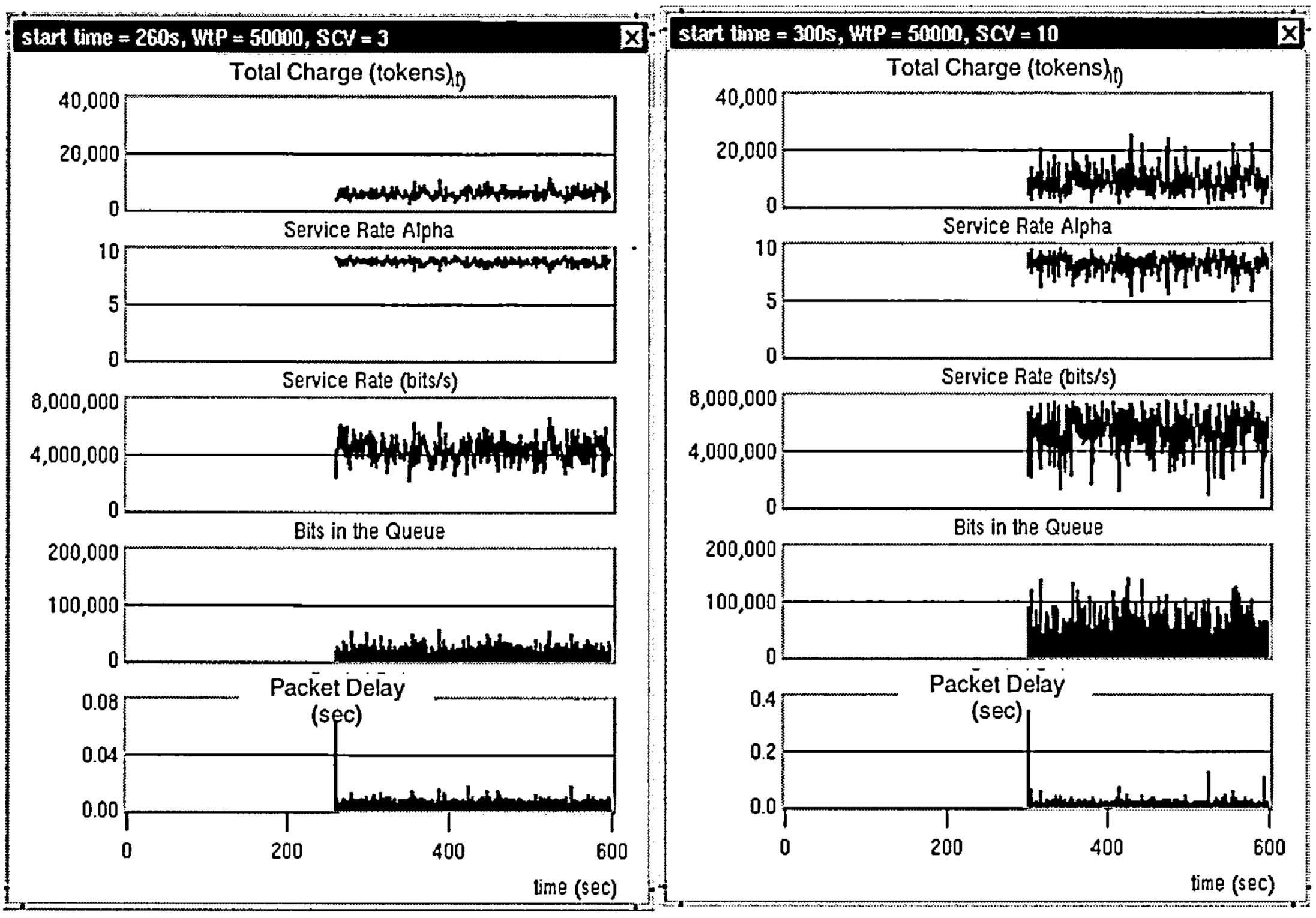

Figure 10.12: Controlled response of flows with high WtP and different burstiness

Finally, we look at Figure 10.12 where the effect on flow 6 and 7 (Table 10.1) is shown. They both have same WtP but the flow on the right is highly bursty (SCV = 10). It is clear that given the same WtP, a more bursty flow experiences more packet delay. However, the situation here is that the allocated WtP is not being fully utilised. The flow has 50000 tokens allocated but mostly uses less than 20000, and yet it is unable to increase the service rate. One of the reasons is that the upper limit on the value of α was 10. Another reason could be the way that an absolute rate controller works. Rate controllers inherently reduce the peak rate, smoothing the traffic, and hence work adequately when peak to mean ratio is not extremely high and priority is to maintain the mean rate. In situations such as

this where the peak to mean ratio is very high (see Figure 10.8), perhaps a different type of controller is needed. This would be something for future work on this method.

However, we have shown that for the majority of traffic types where a rate controller is suitable, the unique mechanism of pricing mean and variance produces a system response acceptable with regard to the expectations laid out earlier in this chapter.

10.8 Summary

In this chapter, the development of a unique price-based feedback system has been described. We started with a description of the DRC mechanism that uses a similar pricing method to route traffic evenly through a carrier network. The problem of congestion avoidance at the output queues of multi-service capable ingress router was highlighted and a practical solution was sought. The author has described her method of pricing for mean and variance which gives flexibility and fairness to flows of all types which may be using the system and gives a means to the network operator to optimise for the high value traffic.

Details of the pricing function and components of flow policing process were described along with the descriptions as to how the algorithms were refined and developed through various stages of testing. The algorithm specifies price calculation and suggests adaptive measures that can be taken by the flow controllers. The results obtained through simulations in OPNET™ Modeler show that:

- The flow controllers are allowed to have higher service rates during low load conditions at the aggregation point.

- If two flows have the same mean and variance but different WtP (see Figure 10.11), the flow with lower WtP is the first to be throttled back.

- A flow with high burstiness is allowed to continue transmitting without significant throttling back provided its WtP is very high (see Figure 10.12).

Before developing the system some objectives and expectations were laid out and the results have shown that the system meets those requirements. We have shown that the mean and variance price feedback scheme is fair to bursty and smooth flows as each flow is charged for its contribution to the congestion. The method does not require per-flow signalling and hence it does not have significant overheads when it is used to control aggregates of thousands of flows. This makes the system scalable for large-scale carrier networks. To our knowledge, this is the first time that a mean and variance price based system has been demonstrated to work and meet the requirements of continuous media traffic.

11

Conclusion and Further Work

This thesis has focussed on a very important aspect of network management and congestion control. There is a great deal of research going on in this field, but the emphasis has been either on the flows requiring rigid guarantees, or on best-effort web traffic. Little work has been carried out for the control of continuous media traffic flows that are adaptive to network congestion. These types of flows are increasingly called for as multimedia applications develop and users expect higher quality real time services to be delivered over the Internet. The study presented in this thesis has investigated methods of congestion control and their suitability for adaptive continuous media. This chapter summarises the thesis, highlights the outcomes of the research and makes a number of suggestions and recommendations for future work.

11.1 Research Summary

The initial phase of the research was to carry out an intensive study of QoS requirements of adaptive continuous media, particularly distinguishing them from the requirements of computer data flows. Chapter 1 set the scene and presented this research in the context of developments in multimedia communications. In Chapter 2, the parameters that define

Quality of Service, such as bandwidth, delay, jitter and loss were discussed. It was highlighted that different applications and media have different priorities for these QoS parameters. For example, video applications require low jitter whereas multimedia mail applications would give higher priority to high throughput. The notion of perceived quality was also discussed. Research into the user's perspective has shown that this is because the human brain quickly adapts to a sustained lower quality video and filters out "noise", whereas the viewer is very aware of problems if there are inconsistencies in the temporal rate [Watson 98].

Then, in Chapter 3, the techniques of congestion control for different types of networks were analysed with respect to reactive congestion control. Congestion in the context of continuous media means the network condition when the load on the network is so high that the QoS requirements of the existing flows are affected. It must be said that the bibliography for this topic is vast and the author has focussed on the points that are specific to adaptive continuous media. The existing forms of congestion control that are used in ATM, and Frame Relay networks were considered. Although, the focus was on packet switched networks such as IP, it is important to have an understanding of other networks as well, and perhaps adapt some of their technologies. Emerging architectures such as Intserv and Diffserv were also discussed. It was noted that the future networks will require an architecture that supports multi-service, in the form of Diffserv, Intserv or something hitherto undeveloped, which will make it possible to distinguish between various classes of traffic. It then follows that due to differentiation between classes, congestion in one class will not affect the traffic in another class. However, it was highlighted that it is still necessary to have some congestion control within the class. It was identified that this is particularly important in classes carrying bursty adaptive flows as it is inefficient to

allocate peak rate bandwidth to each flow and therefore congestion is likely to occur when rigid allocations are not made.

Chapter 4 analysed the aspects of feedback based reactive congestion control and specified the ideal control behaviour that would suit the requirements of adaptive continuous media. This was followed by a description of the simulation model structure, which would be used for evaluation, in Chapter 5.

A number of congestion control algorithms currently used for IP networks were evaluated in detail, namely Hysteresis and RED. These algorithms have been proved to work very well for controlling congestion in data traffic. They were adapted to suit the requirements of continuous media, for example, RED was used to mark the packets instead of dropping them. A requirement of continuous media is to minimise packet losses. The results of simulation were presented and analysed against the specifications in Chapter 6.

Chapter 7 presented the novel control algorithm based on percentile monitoring, with details of the algorithm and results of simulations. Under the same conditions as those for the simulation of Hysteresis and RED schemes, the 99-percentile monitoring technique was demonstrated to be better and more suitable for adaptive continuous media traffic.

Following on from this, congestion problems at the higher level of the network hierarchy were considered. Chapter 8 showed how the percentile monitoring method fits in the hierarchical network. It was noted that as we go deeper into the network hierarchy, flows are aggregated together and the characteristics of the aggregate flow change. This followed on to considering the problem of controlling the amount of traffic entering a carrier network through an ingress router. The carrier network consists of high-speed routers designed to induce minimal delay to the flows. However, the ingress router must ensure that it accepts enough traffic to maximise the revenue while ensuring that delays are

still negligible. Moreover, the ingress router may need to deal with thousands of aggregated flows and therefore, a method which employs per-flow signalling is not practical.

This work was based on the Dynamic Resource Control scheme that had been developed at Nortel. The scheme is described in Chapter 9 along with a description of concepts such as congestion pricing and usage based charging. The "price" is used as a means of control with no relation to the real money although it would be possible to relate it to money for commercial implementations.

Chapter 10 describes a control system incorporating periodic congestion pricing at the edge of a carrier network. The control system consists of the pricing method and a rate-control software that regulates the traffic generation rate at the source of individual flows. The rate control software receives the price information through a feedback mechanism. Significantly, the pricing method, which provides separate prices for mean and variance, solves the problem of how to charge the bursty flows fairly such that they are charged for their contribution to congestion. A carrier network deals with aggregates of flows and this method can be used to charge the individual flows that make up the aggregates.

The thesis thus presented the methodical investigations in congestion control for adaptive continuous media and developments of two new algorithms for controlling congestion at different levels of network hierarchy.

11.2 Research Outcomes

A number of congestion control methods have been used for data traffic where high throughput and reliable delivery is required and delay or jitter is not an issue. With the increase in multimedia services offered over the Internet, much research has been carried out on meeting the QoS requirements of real-time traffic where timely delivery is more

important. Most of the research, however, addresses the issues pertinent to guaranteed services and methods such as Measurement Based Flow Admission Control (MBAC) [Maquosi 02] have been developed to allocate peak rate resources and hence minimise congestion. This research addresses the somewhat neglected area of adaptive flows, which include flows that can adapt to the network conditions and tolerate some gradual degradation. Studies in human behaviour and perceived quality of service [Watson 96, Watson 97] have suggested that users are more likely to notice the changes in quality due to jitter which is caused by fluctuations in temporal resolution than the changes caused by quantisation or spatial resolution.

Based on these theories, a specification of an ideal control scheme was drawn up and two widely used control algorithms for data traffic, Hysteresis and RED, were adapted for use with adaptive continuous media and then simulated in the OPNET™ Modeler environment. The experiments and results are presented in Chapter 6. Very briefly Hysteresis can be described as based on standard control theory methods [Marson 97] using two thresholds and periodic feedback. The feedback is started when the queue exceeds the higher thresholds and continues until the queue reduces to below the lower threshold. RED is also based on monitoring queue occupancy against two thresholds [Floyd 93]. The packets are usually dropped with increasing probability as the queue builds up beyond the lower threshold and all packets are dropped after it exceeds the higher threshold. As the objective in this research was to develop a scheme for continuous media where dropping packets is not desirable, the RED algorithm was adapted so that it stamped the packets with a congestion notification, which was then used as a feedback. The results showed that neither of these methods was fully adequate for the requirements of adaptive continuous media.

A novel scheme of feedback congestion control based on monitoring the 99-percentile of the queue occupancy against two thresholds was developed, described in detail in Chapter 7. In this method, a target queuing delay was chosen from which a target queue size was derived, given the service rate of the queue. For a set of samples of queue size, 99-percentile of the queue size would be equal to the target queue size if only 1 in every 100 samples was larger than the target value. The 99-percentile monitoring method is based on this concept. We accounted for the statistical nature of the sampled data by deriving the confidence regions and using the confidence limits in the algorithm. In the experiments presented in this thesis, the queue size was recorded every time a packet left the service and the number of times the queue size exceeded a set threshold was recorded. If the number of times the queue size exceeded the threshold was greater than the *exceed_limit*, congestion was reported. In a sample of 2000, the *exceed_limit* would be 20 if the system did not have statistical approximations. However, as it was sampled data, the 95% confidence limits were 14 and 28. While the number of times that the queue exceeded the threshold lied between 14 and 28, we could be 95% confident that 99-percentile of packets were experiencing a delay less than or equal to the target delay. The algorithm also has measures for generating more frequent feedbacks during persistent congestion and a reciprocal method for ensuring a lower limit on the target queuing delay as well, and hence used two thresholds.

The simulations in OPNET™ demonstrated that this method clearly outperforms the adapted Hysteresis and RED. Some may argue that it is unfair to compare the algorithm with the adapted versions of Hysteresis and RED as they could be made more aggressive if they were allowed to drop the packets instead of marking them. Firstly, the adaptation was necessary because otherwise we would have an unacceptable level of packet losses and

secondly, we have shown that percentile monitoring provides an improved control on queue occupancy using packet marking. Both Hysteresis and RED are designed to monitor the average queue occupancy and therefore do not ensure the delay bounds on the majority of packets. The advantage of 99-percentile monitoring over these methods is that the target queueing delay can be set in advance. Both Hysteresis and RED can be used to track the 99-percentile queueing delay over a long term through additional feedback but the parameters of dropping/marking probability cannot be chosen for a particular queuing delay.

Meanwhile, a number of new algorithms have also been developed by other researchers, such as Virtual queues, Balanced RED, BLUE [Feng 99c, Feng 99a, Anjum 99]. For example, in Balanced RED packets from different classes of service are given different probabilities of dropping. So, for example, a packet containing video data may have a lower probability of being marked than a packet containing e-mail data. However, they are still contained in the same queue, and therefore will encounter the same delay, which is undesirable for real-time data. The percentile monitoring method has the same advantage over these methods of controlling the system for a known target delay.

Another significant outcome of this research was congestion control at the core of hierarchical networks. Specifically the problem of optimising the volume and the value of the traffic entering a carrier network through an ingress router was considered. The study of traffic behaviour at the carrier network identified that the carrier network routers deal with large aggregates of flows and are designed to perform at high speeds and minimise queuing delays. With this in mind, a method of fairly charging the flows for their contribution to congestion was developed. Based on contemporary research in usage based charging, congestion pricing [Kelly 97, Courcoubetis 97, Gibbens 99] and second moment

measurements [Knightly 99], a unique method of pricing mean and variance was developed. In the mean and variance pricing method, presented in Chapter 10, the mean and variance of the aggregate flow are measured and prices are derived for the given total capacity. Here the price is only a feedback term in the control system; they do not relate to money although commercial developments would be possible. Through simulations in OPNET™, it has been demonstrated that the scheme is able to allow the higher value flows, i.e., the flows with higher ability to pay, to continue while throttling back the lower value flows. This method has advantages over the existing usage based charging schemes, which are designed to generate a single price. The single price can be configured to charge the flows for either their mean usage or their peak usage. However, an aggregate of flows may contain a number of flows with different degrees of burstiness. It would not be fair to charge for mean usage because the smoother flows will have to pay more if a highly bursty flow joins the aggregate. It would be equally unfair to charge for peak usage, because although a highly bursty flow may have a high peak rate, most of the time the peak rate is not utilised. Fairness in this context was defined as flows being charged for their contribution to congestion. The mean and variance pricing method overcomes this problem as the bursty flows are charged more than the smooth flows for their high variance, and because the prices are dynamically changing, the flows would be charged more or less according to the load on the given node or router.

11.3 Further Work

This research has produced two novel congestion control algorithms suitable for adaptive continuous media and applicable at different levels of the network hierarchy. The thesis has presented detailed documentation of the investigations and simulations of the algorithms in OPNET™. While simulations are necessary to formulate the ideas and develop schemes,

their trade-off is that a number of assumptions and simplifications have to be made. It is now necessary to experiment with the parameters, observe the effects and eventually implement and test these algorithms in the real network scenarios to evaluate how they perform with the complexities of real life. In the following paragraph, a number of pointers and recommendations are made for further work building on the findings of this research.

The higher and lower thresholds in the simulations of Hysteresis, RED and Percentile Monitoring were set at 450000 bits and 360000 bits respectively. These values were chosen as they related to queuing delay of 37.5 ms and 30ms respectively for the given service rate of 12Mbits/s. Other values of service rates and target queuing delays and corresponding thresholds can be experimented with for different applications. However, it is important that the target queuing delays form a small fraction of the typical end-to-end delay of 150 – 200 ms that can be tolerated for video because it includes the propagation delay which cannot be avoided and also the difference between the two thresholds must be small to ensure low jitter, ideally in the same order as the acceptable jitter.

In the percentile-monitoring algorithm, the 99-percentile queue occupancy was calculated using a sampling technique. The choice of values for parameters such as initial sample size and elongated sample size meant that for an average packet size of 80000 bits and queue service rate of 12 Mbit/s, this method generated a feedback every 60ms during congestion. Since the average time between successive packets was 40ms in these experiments, more frequent feedback was not necessary. It is important that feedbacks are sent frequently enough to instigate fast response but not so frequently that they clog up the system. The frequency of the feedback is governed by the *sample_size* and the related *exceed_limit* (the number of times the queue is allowed to exceed the threshold), and the frequency of packets leaving the node, which depends on the service rate and average

packet size. These parameters must be experimented with in order to tailor the control system to the system that needs to be controlled. For example, in systems with a higher frame rate, it would be necessary to choose a smaller sample size so that the feedback could be generated more frequently.

Higher service rates will also generate more frequent feedback during congested periods. Also, the queue was sampled every time a packet left the service. This was useful because it avoided unnecessary positive feedbacks instructing the sources to increase the data rate if the node was almost idle and ensured more frequent feedbacks when the node was congested. For some applications, it may be useful to sample the queue periodically (at regular time intervals). This would be a minor change to the front-end of the algorithm.

The simulation model used for Hysteresis, RED and percentile-monitoring incorporated a very simple packet truncation method to model changes in spatial resolution. It would now be worthwhile to construct a lookup table based on the behaviour of an adaptive encoder for the source model so that it can change the data rate accordingly instead of the method of fractional reduction of packet size used here.

The percentile-monitoring model is presented here to control congestion at a single node. Before collaborating with Nortel, it was the author's intention to develop a multi-hop solution. Monitoring percentile delay across multiple nodes is, however, a complex problem. The delay experienced by packets at a node will be affected by the service rate of the previous node as well as the service rate of the given node. As a starting point, for homogenous routers at least, a possible method would be to configure the first node for the given target delay and the successive nodes with much lower target delays and smaller differences between the delay limits.

It is also necessary to further develop the congestion-pricing scheme for its effective use. The flow controllers in our experiments used absolute rate controllers, which effectively smooth the flow when there is heavy congestion. For flows to benefit from dual pricing such that they are allowed to reduce their mean rate but keep a high variance a different controller needs to be designed. Adaptive encoders at the traffic source can be used for this. Alternatively, the rate control software can be adapted to control the traffic at the input queues of the ingress router. Additionally, it would be worthwhile to formalise the mean and variance dual-pricing algorithm algebraically.

Finally, it would be interesting to take an entirely different approach to solving congestion problems. Recent research has shown that there are strong correlations in the inter-arrival processes of network traffic. Self-similarity and Long Range Dependence (LRD) have been very hot topics. There is a school of thought which believes that because of these attributes of network traffic, the conventional methods of controlling congestion cannot be applied directly. Research is ongoing into chaotic mapping of traffic and into using the concepts of chaos theory to control congestion in the network. A chaotic system has many possible periodic states that can co-exist but one does not have the knowledge as to which orbit the system is going to follow next. Researchers are actively working on methods of taming a chaotic system by applying small perturbations, bringing order out of chaos [Crook 99]. This is a very interesting development and it may have applications for the control of network traffic as well [Samuel 98a, Samuel 98b]. If it is possible to predict congestion, we could indeed eliminate the congestion altogether by sending a feedback before the queuing delays occur so that the flows can take corrective measures. However, it is yet to be ascertained if the traffic processes are indeed chaotic and, so far, real-time video traffic has not been shown to exhibit a self-similar pattern.

11.4 Summary

In a nutshell, this thesis is effectively a detailed account of the author's research in the vast field of network congestion and control, focussing on adaptive continuous media. The technology is changing rapidly and architectures such as Internet2, Diffserv etc may not be too far in future. Future networks will most likely have multi-service capabilities and different types of classes. However, congestion is still going to be a problem within a traffic class even if it is isolated from other types of flows through bandwidth partitioning. The choice of focussing on adaptive continuous media was made as it seems to be an area with a high potential for providing a range of services from networked video games to video-conferencing for which people, in keeping with the usual human tendencies, would like to pay as little as they can to get the best quality possible. It was seen that with the technology available and that which is being developed, it should be possible to offer services to this user-base.

The outcomes of the work, particularly the percentile monitoring algorithm at flow level control and mean and variance pricing for aggregates of flows have been shown to be more suitable to the requirements of adaptive flows. The work has been published in a number of papers [Ball 99d, Ball 99b, Tater 00a, Tater 00b]. As is always the case, research opens up opportunities for further investigations and it is hoped that these algorithms would be developed further in future and become integral parts of network congestion control.

References

[Andersen 00] Andersen N E, Azcorra A, Bertelsen E, Carapinha J, Dittman L, Fernandez D, Kjaergaard J K, McKay I, Maliszewski J, and Papir Z, "Applying QoS Control through Integration of IP and ATM", IEEE Communications Magazine, July 2000, Vol.38, No.7, pp.130 - 136

[Anjum 99] Anjum F M and Tassiulas L, "Balanced-RED: An Algorithm to Achieve Fairness in the Internet", Technical Report CSHCN T.R. 99-9, 1999, URL: http://www.isr.umd.edu/TechReports/CSHCN/1999/

[ATM Forum 01] The ATM Forum: Technical Specifications, URL: http://www.atmforum.com/techspecfs1.html, August 2001

[Bahner 98] Bahner T, Carlson S, Exter A, Kaplan M, Nicoll C, and Hyon C V, "The Basic Guide to Frame Relay Networking", URL: http://www.frforum.com/basics.pdf, 1998

[Baldi 00] Baldi M and Ofek Y, "End-to-End Delay Analysis of Videoconferencing over Packet-Switched Networks", IEEE/ACM Transactions on Networking, August 2000, Vol.8, No.4, pp.479 - 492

[Ball 96a] Ball F, "*Supporting Quality of Service Guarantees across Multi-hop Heterogenous Networks*", PhD Thesis, Lancaster University, May 1996

[Ball 99a] Ball F and Callinan P, "Supporting Guaranteed Services in Packet Switched Networks: A Study of Two Alternative Networks", **International Conference on Parallel and Distributed Processing Techniques Applications, PDPTA '99** , 1999, Eds. Arabania H R, Vol. 5, ISBN: 1-892512-13-0

[Ball 99c] Ball F, Callinan P, Kouvatsos D D, and Skianis H, "A Measurement Based Admission Control Mechanism for use with CBQ in Packet Switched Networks", **Proc.15th Annual UK Performance UKPEW'99**, 22 - 23 July, 1999, Bristol, UK, Eds. Bradley J T and Davies N J, ISBN: 0-9524027-8-5, pp.295 - 303

[Ball 96b] Ball F, Hutchison D, and Kouvatsos D D, "Matching the Temporal Characteristics of Continuous Media and Networks", **Proc.13th IEE UK Teletraffic Symposium**, 1996, Glasgow, Scotland

[Ball 96c] Ball F, Hutchison D, and Kouvatsos D D, "VBR Video Traffic Shaping for ATM Networks", **ATM, Networks and LANs Proceedings**, November 1996, Eds. Faulkner D W and Hamer A L, ISBN: 90-5199-272-2, pp.68 - 74

[Ball 99b] Ball F and Tater S, "Reactive Congestion Control for Adaptive Continuous Media", **Proc.2nd GEMISIS technical symposium on Multimedia - network - technology**, 17 - 18 May 1999, Salford, UK

[Ball 99d] Ball F and Tater S, "Supporting Adaptive Video Applications in Future IP Networks", **IEE European Workshop on Distributed Imaging**, 18 - 19 November 1999, Savoy Place, London

[Bansal 01] Bansal D, Balakrishnan H, Floyd S, and Shenker S, "Dynamic Behaviour of Slowly- Responsive Congestion Control Algorithms", **Proc.ACM SIGCOMM 2001**, 27 - 31 August 2001, San Diego, California, pp.263 - 274

[Bernet 00] Bernet Y, "The Complimentary Roles of RSVP and Differentiated Services in the Full-Service QoS Network", IEEE Communications Magazine, February 2000, Vol.38, No.2, pp.154 - 162

[Biddiscombe 00] Biddiscombe M D, "*Free Market Communications*", PhD Thesis, University College London, 2000

[Bouch 99] Bouch A and Sasse M A, "It Ain't What Your Charge, It's the Way That You Do It: A User Perspective of Network QoS and Pricing", **Proc.IFIP/IEEE International Symposium on Integrated Network Management, IM'99**, 24 - 28 May 1999, Boston, MA, pp.639 - 655, URL: http://www.cs.ucl.ac.uk/staff/A.Sasse/pub.html

[Busse 95] Busse I, Deffner B, and Schulzrinne, "Dynamic QoS Control of Multimedia Applications based on RTP", URL: http://www.fokus.gmd.de/step/acontrol/ac.html, 1995

[Callinan 00a] Callinan P, "*Provision of Service Guarantees to Real-time Traffic in Packet Switched Networks*", PhD Thesis, Oxford Brookes University, February 2000

[Callinan 00b] Callinan P, Witwit M, and Ball F, "A Comparative Evaluation of Sorted Priority Algorithms and Class Based Queueing using Simulation", **Advanced Simulation Technologies Conference ASTC 2000**, 16 - 20 April 2000, Washington DC

[Carroll 99] Carroll J E and Kirkby P A, "Proportionally Fair Pricing in Hierarchial Networks: Dynamics and Stability", **IEE Colloquium 'Control of Next Generation Networks'**, 18 October 1999

[Case 02] Case K E and Fair R C, "The Price System, Demand and Supply, and Elasticity", in *Principles of Economics*, Prentice-Hall, 2002, ISBN: 0-13-073772-0, URL: http://www.prenhall.com/casefair/virtual/ppts/ch04.ppt

[Casetti 96] Casetti C, Kurose J, and Towsley D, "An Adaptive Algorithm for Measurement-based Admission Control in Integrated Services Packet Networks", Technical Report TR 96-76, University of Massachusetts, 1996

[Courcoubetis 00] Courcoubetis C, Kelly F P, Siris V A, and Weber R, "A Study of Simple Usage-Based Charging Schemes for Broadband Networks", Telecommunication Systems, 2000, Vol.15, No.3-4, pp.323 - 343

[Courcoubetis 97] Courcoubetis C, Kelly F P, and Weber R, "Measurement-Based Usage Charges in Communications Networks", Statistical Laboratory Research Report 1997-19, University of Cambridge, 1997

[Crook 99] Crook N T, Dobbyn C H, and olde Scheper T, "Chaos as a Desirable Stable State of Artificial Neural Network", in *Advances in Soft Computing: Soft Computing Techniques and Applications*, **Ed(s).** John R and Birkenhead R, **Series Ed(s).**Kacprzyk J, Physica-Verlag, Heidelberg, NewYork, 1999, ISBN: 3790812579

[Dagiuklas 96] Dagiuklas A and Ghanbari M, "Preventive Flow Control method for Packet Video", IEE Proceedings Communications, 1996, Vol.143, No.2, pp.98 - 104

[DaSilva 00] DaSilva L A, "Pricing for QoS-Enables Networks: A Survey", URL: http://www.comsoc.org/pubs/surveys, Second Quarter 2000

[Feng 98] Feng W C, Kandlur D D, Saha D, and Shin K G, "Adaptive Packet Marking for Providing Differentiated Services in the Internet", **ICNP'98**, October 1998

[Feng 99c] Feng W C, Kandlur D D, Saha D, and Shin K G, "Adaptive Packet Marking for Maintaining End-to-End Throughput in a Differentiated-Services Internet", IEEE/ACM Transactions on Networking, October 1999, Vol.7, No.5, pp.685 - 697

[Feng 99a] Feng W C, Kandlur D D, Saha D, and Shin K G, "BLUE: A New Class of Active Queue Management Algorithms", Technical Report CSE-TR-387-99, University of Michigan , April 1999

[Floyd 94] Floyd S, "TCP and Explicit Congestion Notification", ACM Computer Communication Review, October 1994, Vol.24, No.5, pp.10 - 23

[Floyd 98] Floyd S and Fall K, "Promoting the Use of End to End Congestion Control in the Internet", IEEE/ACM Transactions on Networking, 10 February 1998, URL: http://ftp.ee.lbl.gov/floyd

[Floyd 93] Floyd S and Jacobson V, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, August 1993

[Fluckiger 95] Fluckiger F, *Understanding Networked Multimedia Applications and Technology*, Prentice-Hall, 1995

[Ghanbari 99] Ghanbari M, *Video Coding: An Introduction Standard Codecs*, IEE Telecommunications Series 42, **Series Ed(s).** Hughes C J, Parsons D, and White G, The Institution of Electrical Engineers, 1999, ISBN: 0-85296-762-4

[Gibbens 97] Gibbens R J and Kelly F P, "Measurement-based Connection Admission Control", **Proc.15th International Teletraffic Congress, ITC-15**, June 1997

[Gibbens 99] Gibbens R J and Kelly F P, "Resource pricing and Evolution of Congestion Control", Automatica 35 (1999), pp. 1969-1985. Also available at Statistical Laboratory, University of Cambridge. URL: http://www.statslab.cam.ac.uk/~frank/evol.html

[Gross 98] Gross D and Harris C M, "Poisson Process and the Exponential Distribution", in *Fundamentals of Queueing Theory*, 3rd Editon, Wiley Series in Probabilty and Statistics, **Series Ed(s).** Barnett V and et al, John Wiley & Sons, 1998, ISBN: 0-471-17083-6, pp.16 - 22

[Ibrahim 98] Ibrahim A A M, "Statictical rate Control for Efficient Admission Control of MPEG-2 VBR Video Sources", **Proc.IEEE ATM Workshop ATM'98**, May 1998, Fairfax, VA, pp.300 - 305

[ITU 98a] Methodology for the Subjective Assessment of Quality of Television Pictures ITU-R BT.500-7, ITU Recommendations, 1998

[ITU 98b] Methods for Subjective Determination of Transmission Quality ITU-T P.800, ITU-T Recommendations, 1998

[Jacobson 88] Jacobson V, "Congestion Avoidance and Control", **Proc.ACM SIGCOMM '88**, August 1988, Palo Alto, CA, URL: http://www- nrg.ee.lbl.gov/nrg-papers.html

[Kalyanaraman 98] Kalyanaraman S, Vandalore B, Jain R, Goyal R, and Fahmy S, "Performance of TCP over ABR with Long Range Dependent VBR Background Traffic over Terrestrial and Satellite ATM Networks", URL: http://www.cis.ohio-state.edu/~jain/papers.html, 1998

[Kanakia 96] Kanakia H and Mishra P P, "Packet Video Transport in ATM Networks with Single-Bit Feedback", Multimedia Systems, December 1996, Vol.4, No.6, pp.370 - 380

[Kanakia 93] Kanakia H, Mishra P P, and Reibman A R, "An Adpative Congestion Control Scheme for Real Time Packet Video Transport Model", **Proc.ACM SIGCOMM '93**, September 1993, San Fransisco, USA, pp.20 - 31

[Kelly 97] Kelly F P, "Charging and Rate Control for Elastic Traffic", European Transactions on Telecommunications, 1997, Vol.8, No.2, pp.33 - 37

[Key 99] Key P B, McAuley D R, Barham P, and Laevens K, "Congestion Pricing for Congestion Avoidance", Technical Report MSR-TR-99-15, Microsoft Research, February 1999

[Kirkby 99b] Kirkby P A and Kadengal R, "Traffic Management and Control Using a Single 'Congestion Price' like Variable across Multiple Layers of network Hierarchy", **IEE Colloquium 'Control of Next Generation Networks'**, 18 October 1999, London

[Kirkby 99a] Kirkby P A, Kadengal R, Midwinter J E, Biddiscombe M D, Carroll J E, and Sabesan S, "The Use of Economic and Control Theory Analogies in the Design of Policy Based Dynamic Resource Controlled Network Architectures", **Proc.16th International Teletraffic Congress, ITC-16**, June 1999, Edinburgh, UK

[Knightly 97] Knightly E W, "Second Moment Resource Allocation in Multi-Service Networks", **Proc.ACM SIGMETRICS'97**, 1997, Seattle, pp.181 - 191

[Knightly 99] Knightly E W, "Resource Allocation for Multimedia Traffic Flows using Rate-Variance Envelopes", ACM Multimedia Systems Journal, November 1999, Vol.7, No.6, pp.477 - 485, URL: http://www.ece.rice.edu/networks/publications.html

[Kunniyur 00] Kunniyur S and Srikant R, "End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks", **Proc.IEEE INFOCOM 2000**, November 2000, Tel-Aviv, Israel, URL: http://comm.csl.uiuc.edu/~kunniyur/research.html

[Lakshman 99] Lakshman T V, Mishra P P, and Ramakrishnan K K, "Transporting Compressed Video Over ATM Networks with Explicit-Rate Feedback Control", IEEE/ACM Transactions on Networking, October 1999, Vol.7, No.5, pp.710 - 723

[Leland 94] Leland W E, "On the Self Similar Nature of Ethernet Traffic", IEEE Transactions on Networking, February 1994, Vol.2, No.1, pp.1 - 15

[Lougher 92] Lougher P, Shepherd W D, and Pegler D, "A Scalable Hierarchical Video Storage Architecture", **Proc.SPIE Conference on Multimedia Computing and Networking**, 28 - 31 January, 1992, San Josè, California, USA

[Maquosi 02] Maquosi A, Tater S, and Ball F, "Traffic Monitoring Techniques for Measurement Based Flow Acceptance Control", **35th Annual Simulation Symposium 2002**, 14 - 18 April 2002, San Diego, California

[Marson 97] Marson M, Bianco A, Cigno R, and Munafo M, "Four Standard Control Theory Approaches for the Implementation of EFCI ABR Services", in *ATM Networks: Performance Modelling and Analysis*, **Ed(s)**.Kouvatsos D D, Chapman & Hall, 1997, ISBN: 0-412-80970-2

[McDysan 00] McDysan D, *QoS & Traffic Management in IP and ATM Networks*, McGraw-Hill, 2000, ISBN: 0-07-134959-6

[Michalareas 01] Michalareas T, Sacks L, and Kirkby P A, "Dynamic Value-Based Lightpath Allocation in DWDM Networks", **IEEE GLOBECOM 2001**, 24 - 29 November 2001, San Antonio, Texas, USA

[MPEG 00b] MPEG Pointers and Resources, URL: http://www.mpeg.org/, May 2000

[MPEG 00a] The MPEG Home Page, ISO/TEC JTC1/SC29 WG11, URL: http://mpeg.telecomitalialab.com/, June 2000

[Podolsky 98] Podolsky M, Romer C, and McCanne S, "Simulation of FEC-based Error Control for Packet Audio on the Internet", **Proc.IEEE INFOCOM '98**, 1998

[RFC 1121] Postel J, Kleinrock L, Cerf V, and Boehm B, "Internet RFC 1121 -- Act One - The Poems", URL: http://www.faqs.org/rfcs/rfc1121.html, September 1989

[RFC 1883] Deering S and Hinden R, "Internet RFC 1883 -- Internet Protocol, Version 6 (IPv6) Specification", URL: http://www.faqs.org/rfcs/rfc1883.html, December 1995

[RFC 2205] Braden R, Zhang L, Berson S, Herzog S, and Jamin S, "Internet RFC 2205 -- Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification", URL: http://www.faqs.org/rfcs/rfc2205.html, September 1997

[RFC 2481] Ramakrishnan K K and Floyd S, "Internet RFC 2481 -- A Proposal to add Explicit Congestion Notification (ECN) to IP", URL: http://www.faqs.org/rfcs/rfc2481.html, January 1999

[RFC 2581] Allman M, Paxson V, and Stevens W, "Internet RFC 2581 -- TCP Congestion Control", URL: http://www.faqs.org/rfcs/rfc2581.html, April 1999

[Press 93] Press W H, Teukolsky S A, Vetterling W T, and Flannery B P, *Numerical Recipies in C: The Art of Scientific Computing*, Cambridge University Press, 1993, ISBN: 0-521-43108-5, URL: http://www.ulib.org/webRoot/Books/Numerical_Recipes/bookcpdf.html

[Qiu 01] Qiu J and Knightly E W, "Measurement-Based Admission Control with Aggregate Traffic Envelopes", IEEE/ACM Transactions on Networking, April 2001, Vol.9, No.2

[Ramakrishnan 90] Ramakrishnan K K and Jain R, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks", ACM Transactions on Computer Systems, 1990, Vol.8, pp.150 - 181

[Ramanujan 97] Ramanujan R S, Newhouse J A, Kaddoura M N, Ahamad A, Chartier E R, and Thurber K J, "Adaptive Streaming of MPEG Video over IP Networks", **Proc.22nd Annual Conference on Local Computer Networks, 1997, LCN'97**, 2 - 5 November, 1997, pp.398 - 409

[Romanow 95] Romanow A and Floyd S, "Dynamics of TCP Traffic over ATM Networks", IEEE Journal on Selected Areas in Communications, May 1995

[Sahinoglu 99] Sahinoglu Z and Tekinay S, "On Multimedia Networks: Self-Similar Traffic and Network Performance", IEEE Communications Magazine, Jan 1999, Vol.37, No.1, pp.48 - 57

[Samuel 98b] Samuel L, Pitts J M, and Mondragón R J, "Applications of Non-Linear Dynamics to Network Modelling", **Proc.15th UK Teletraffic Symposium on Performance Engineering in Information Systems**, 23 - 25 March 1998, Durham, UK

[Samuel 98a] Samuel L, Pitts J M, Mondragón R J, and Arrowsmith D K, "The MAPS Control Paradigm: using Chaotic Maps to Control Telecoms Networks", in *Broadband Communications - The Future of Telecommunications*, **Ed(s).**Kühn P and Ulrich R, Chapman and Hall, London, 1998, pp.371 - 382

[RFC 1889] Schulzrinne H, Fokus GMD, Casner S, Frederick R, and Jacobson V, "Internet RFC 1889 -- RTP: A Transport Protocol for Real-Time Applications", URL: http://www.faqs.org/rfcs/rfc1889.html, January 1996

[Sikora 97] Sikora T, "MPEG-1 and MPEG-2 Digital Video Coding Standards", in *"MPEG Digital Video Coding Standards" in Digital Electronics Consumer Handbook*, **Ed(s).**Jurgen R, Mc-Graw Hill, 1997, ISBN: 007-034-1435

[Tannenbaum 96] Tannenbaum A S, *Computer Networks*, 3rd Editon, Prentice-Hall, 1996, ISBN: 0-13-394284-1

[Tater 00c] Tater S, "An Analytical Comparison of Bandwidth Usage between Vanilla Diffserv and DRC", Internal Report, Nortel Networks, Harlow Labs, UK, 6 November 2000

[Tater 00d] Tater S, "Effect of Increased Flows on Queuing Delay", Internal Report, Nortel Networks, Harlow Labs, UK, 23 November 2000

[Tater 00b] Tater S and Ball F, "Delay Monitoring Techniques for Adaptive Continuous Media", **Proc.1st Annual PGNet Symposium on the Convergence of Telecommunications, Networking and Broadcasting**, 19 - 20 June 2000, Liverpool

[Tater 00a] Tater S and Ball F, "Meeting the Needs of Adaptive Video Applications in Packet Switched Networks", **Proc.18th IASTED International Conference on Applied Informatics 'AI 2000**, 14 - 17 February 2000, Innsbruck, Austria

[Thom 99] Thom D, Purnhagen H, Pfeiffer S, and MPEG Audio Subgroup, "MPEG Audio FAQ", URL: http://www.tnt.uni-hannover.de/project/mpeg/audio/faq/, December 1999

[Tuan 98] Tuan T and Park K, "Congestion Control for Self Similar Network Traffic", Technical Report CSD-TR 98-014, Purdue University, May 1998

[Watson 97] Watson A, "Evaluating Real-Time Multimedia Audio and Video Quality", URL: http://www.cs.ucl.ac.uk/staff/awatson/aw.htm, 1997

[Watson 96] Watson A and Sasse M A, "Evaluating Audio and Video Quality in Low-cost Multimedia Conferencing Systems", Interacting with Computers, 1996, Vol.8, No.3, pp.255 - 275, URL: http://cs.ucl.ac.uk/staff/awatson/aw.htm

[Watson 98] Watson A and Sasse M A, "Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications", ACM Multimedia 98 - Electronic Proceedings, 1998, URL: http://www.cs.ucl.ac.uk/staff/awatson/mm98.html

[Willebeek-LeMair 97] Willebeek-LeMair M H and Shae Z Y, "Videoconferencing over packet-based networks", IEEE Journal on Selected Areas in Communications, 1997, Vol.15, No.6, pp.1101 - 1114

[Williams 92] Williams N and Blair G S, "Distributed Multimedia Application Study", Technical Report MPG-92-11, Lancaster University, 1992

[Wonnacott 90] Wonnacott T H and Wonnacott R J, *Introductory Statistics*, 5th Editon, John Wiley & Sons, 1990, ISBN: 0-471-61518-8

[Wrege 96] Wrege D E, Knightly E W, Zhang H, and Liebeherr J, "Deterministic Delay Bounds for VBR Video in Packet-Switching Networks: Fundamental Limits and Trade-Offs", IEEE/ACM Transactions on Networking, June 1996, Vol.4, No.3, pp.352 - 357

[Zhang 93] Zhang L, Deering S, Estrin D, Shenker S, and Zappala D, "RSVP: A New Resource ReSerVation Protocol", IEEE Network Magazine, 1993, Vol.7, No.5, pp.8 - 18

[Zheng 99] Zheng B and Atiquazzaman M, "Traffic Management of Multimedia over ATM Networks", IEEE Communications Magazine, Jan 1999, Vol.37, No.1, pp.33 - 38

Glossary of Terms

ABR	Available Bit Rate
ACK	Acknowledgement (Packet)
ADPCM	Advanced Differential Pulse Code Modulation
AF	Assured Forwarding
ARPA	Advanced Research Projects Agency
ATM	Asynchronous Transfer Mode
BCN	Backward Congestion Notification
BE	Best-Effort
BECN	Backward Explicit Congestion Notification
CAC	Call Admission Control
CBQ	Class Based Queuing
CBR	Constant Bit Rate
CCTV	Closed Circuit TeleVision
CLR	Cell Loss Ratio
CSCW	Computer Supported Cooperative Work
DCT	Discrete Cosine Transform
Diffserv	Differentiated services
DRC	Dynamic Resource Control
DTE	Data Terminal Equipment
ECN	Explicit Congestion Notification
EF	Expedited Forwarding
EFCN	Explicit Forward Congestion Notification
FCI	Forward Congestion Indication
FECN	Forward Explicit Congestion Notification
FIFO	First In First Out
GOP	Group of Pictures
HDTV	High Definition TeleVision
IETF	Internet Engineering Task Force
Intserv	Integrated services
IP	Internet Protocol
ISO	International Standards Organisation
MBAC	Measurement Based Admission Control
MFAC	Measurement based Flow Admission Control
MPEG	Moving Pictures Expert Group
MPLS	Multiple Path Label Switching
PCM	Pulse Code Modulation
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RED	Random Early Detection
REM	Random Early Marking
RFC	Request For Comments
RSVP	Resource reSerVation Protocol
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
SCV	Squared Coefficient of Variance

TCP	Transport Control Protocol
ToS	Type of Service
UDP	User Datagram Protocol
UPC	Usage Parameter Control
VBR	Variable Bit Rate
VCI	Virtual Circuit Identifier
VoD	Video on Demand
VoIP	Voice over IP
VPI	Virtual Path Identifier
WFQ	Weighted Fair Queuing
WtP	Willingness to Pay

Appendix I

MPEG Video Packet Trace[17]

Bytes	Bits
24940	199520
6848	54784
6592	52736
12928	103424
7040	56320
7424	59392
13312	106496
7104	56832
7168	57344
13504	108032
7168	57344
7296	58368
23468	187744
7360	58880
7424	59392
12864	102912
7168	57344
7232	57856
13312	106496
6592	52736
6720	53760
13312	106496
6720	53760
7232	57856
25388	203104
6400	51200
6720	53760
13248	105984
6848	54784
6976	55808
13312	106496
6784	54272
6912	55296
13376	107008

	bytes	bits
Average	9989.2972	79914.378
Min	1152	9216
Max	27436	219488
Variance	23264962	1.489E+09
SCV	0.2331484	0.2331484

[17] The complete trace had 5430 packet lengths. This is just a selection.

Ethernet Packet Trace[18]

Bytes	Bits
1374	10992
377	3016
363	2904
712	5696
387	3096
409	3272
733	5864
391	3128
394	3152
744	5952
394	3152
401	3208
1293	10344
405	3240
409	3272
708	5664
394	3152
398	3184
733	5864
363	2904
370	2960
733	5864
370	2960
398	3184
1398	11184
352	2816
370	2960
729	5832
377	3016
384	3072
733	5864
373	2984
380	3040
736	5888
387	3096
370	2960

	bytes	bits
Average	549.85691	4398.8552
Min	63	504
Max	1511	12088
Variance	70624.404	4519961.8
SCV	0.2335906	0.2335906

[18] This was derived from the MPEG trace but the values were downsized to comply with Ethernet standards where the maximum packet size is 1514 bytes.

Appendix II

Specifications for Model used in Hysteresis, RED and Percentile Monitoring Experiments

Packet Generator

The packet generator process has a number of attributes such as average interarrival time, squared coefficient of variance (SCV), and trace file name that are promoted so that they can be set at the time of simulation (see Table II-1). This enables the same process model to be used for generating different types of traffic. The trace file contains a number of null terminated strings each of which is read in as the packet length. Trace files used in this work are shown in Appendix I. The average packet length and squared coefficient of variance were derived from the packet length statistics.

Table II-1: Attributes of Background and other sources used in Feedback based Control Simulations

	Variable Type	Background Source	Other Sources
Average Interarrival Time	Double	13 ms	40 ms
scv of interarrival time	Double	1.0	1.0
Packet Trace File	String	MPEG	MPEG
Average Packet Length	<derived>	80000 bits	80000 bits
scv of Packet length	<derived>	0.233	0.233
Average Bit Rate	<derived>	6.15 Mbit/s	2 Mbit/s

The packet generators have two more attributes specified at simulation time which are start time and stop time. Both of them are stored as double. They were specified according to the scenario being modelled.

Packet Format

The packet format used in the models is shown in Figure II-1. It was customised to consist of only the fields that were relevant to the simulations.

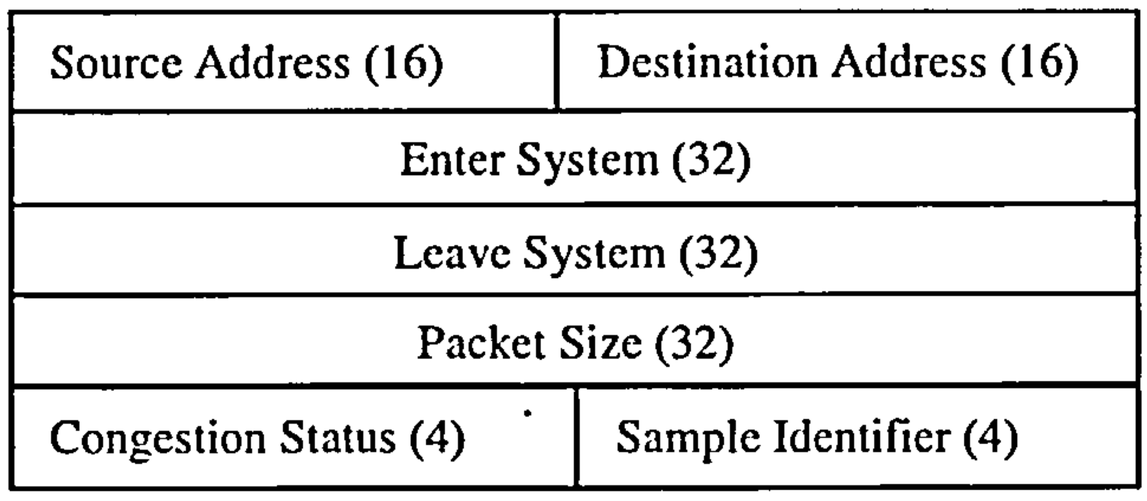

Figure II-1: Packet format used by generators

The Sample Identifier field is only used in Percentile Monitoring. This is a customised packet format used for simulation and hence the length of the packet is not measured but read from the Packet size field.

Bottleneck Link

The bottleneck link implemented the congestion detection algorithm and hence changed significantly for each set of simulated experiments. Some of the attributes were always kept consistent while other attributes, such as filter coefficient were not always required. The specifications are summarized in Table II-2.

Table II-2: Attribute Summary for Bottleneck Link

Attribute Name	Variable Type	Hysteresis	RED	Percentile Conservative	Percentile Liberal
Filter coefficient (α)	Double	0.5	0.5	n.a.	n.a.
Service Rate	Double	12 Mbit/s	12 Mbit/s	12 Mbit/s	12 Mbit/s
Higher Threshold	Integer	4500000 bits	4500000 bits	4500000 bits	4500000 bits
Lower Threshold	Integer	3600000 bits	3600000 bits	3600000 bits	3600000 bits
Base Sample	Integer	n.a.	n.a.	2000	2000
Increment Sample	Integer	n.a.	n.a.	1000	1000
Base Deviation	Integer	n.a.	n.a.	14	28
Increment Deviation	Integer	n.a.	n.a.	9	11

The filter coefficient is used in obtaining the average queue occupancy before checking against the thresholds in case of Hysteresis and RED. Service rate is the link rate of the queue output. Higher and lower thresholds were specified as integers. The last four attributes shown in the table above are used in the percentile monitoring method. Base Sample gives the *sample_size* to begin the monitoring. The Base Deviation specifies the value of *exceed_limit* for this *sample_size*. During persistent congestion, the sample has to be elongated. The Increment Sample value is used to increase the *sample_size* and the new corresponding *exceed_limit* is obtained by adding Increment Deviation to the Base Deviation.

Flow Controller

The flow controller process is designed to react to the feedback signals that it receives. It has one attribute specified at simulation time as shown in Table II-3. The Buildup Interval attribute is used in the case of Hysteresis and RED only. It is used to schedule the buildup interrupt after a feedback signal has been received. In the case of Percentile monitoring, an explicit feedback is received to indicate that the controller may increase the data rate and hence the buildup function is not required. The actions are summarized in Table II-4.

Table II-3: Attributes for Flow Controller

Attribute Name	Variable Type	Hysteresis	RED	Percentile Conservative	Percentile Liberal
Buildup Interval	Double	2s	2s	n.a.	n.a.

Table II-4: Events and Actions of Flow Controller

Event	Hysteresis	RED	Percentile
Feedback packet arrives	Clear pending interrupts Feedback is always negative Decrement f by 0.1 unless f = 0.5 Schedule interrupt to be invoked when build-up interval time has elapsed	Clear pending interrupts Feedback is always negative Decrement f by 0.1 unless f = 0.5 Schedule interrupt to be invoked when build-up interval time has elapsed	If feedback is negative Decrement f by 0.1 unless f = 0.5 If it is positive, increment f by 0.1 unless f = 1.0
Build-up interrupt occurs. This can only occur if build-up interval duration has elapsed since last feedback was received because a feedback clears the interrupt.	Increment f by 0.1 unless f = 1.0 Schedule another interrupt to be invoked when build-up interval time has elapsed	Increment f by 0.1 unless f = 1.0 Schedule another interrupt to be invoked when build-up interval time has elapsed	Not applicable

Destination

The destination process has the job of creating the feedback packet. The packet format used is very simple and contains only a congestion field. A Boolean value would suffice because in the case of Hysteresis and RED feedback is sent only to indicate congestion, therefore a packet with the congestion field set to 1 would be adequate. In the case of Percentile monitoring the value will be 1 to indicate congestion and 0 to indicate no congestion. The actions of the Destination Process under each algorithm are summarized in Table II-5.

Table II-5: Events and Actions of Destination

Event	Hysteresis	RED	Percentile Monitoring
Packet arrives	Compare the Congestion Value with the Last Congestion Value (stored) If there is no change, do nothing Else if value changed from not congested to congested, - send a negative feedback. - schedule an interrupt for next feedback. Else the value has changed from congested to not congested, then do nothing. Finally, store the congestion value and destroy the packet	If packet indicates congested, send a negative feedback Else do nothing. In all cases, destroy the packet	Compare the congestion value with the last congestion value and Sample ID with the last Sample ID. If both are unchanged, then do nothing Else create a feedback packet, copy the congestion value and send it. Finally, store the congestion value and sample id and destroy the packet
Self Interrupt occurs	If the last packet indicated congestion, generate feedback signal and schedule another interrupt Else do nothing.	Not applicable	Not applicable

Appendix III

Proof for Model used in Hysteresis, RED and Percentile Monitoring

In order to prove that the network model worked correctly, a reference model has been designed. This was created using already verified models built in OPNET™ Modeler.

Reference Model

The reference model was created using three processes: Packet Generator, Queue and Sink. In the model, there were 7 nodes with the Packet Generator process generating and sending packets to a single queue node, which would forward the packets to the sink where they would be destroyed [Figure III-1].

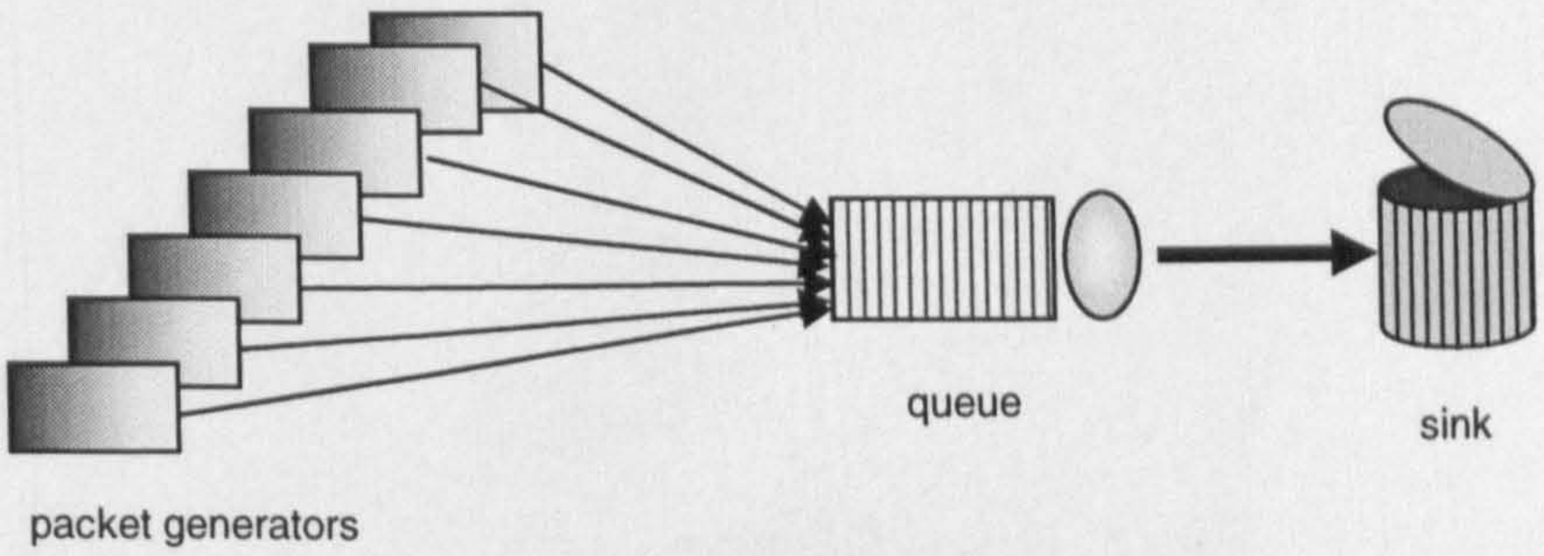

Figure III-1: Reference Model

The packet generator was based on the generalised exponential algorithm which has already been evaluated in [Ball 96a]. The bit pattern generated by the process using the MPEG video trace, see Appendix I, at average interarrival of 40 ms is shown in Figure III - 2. The graph shows the number of bits sent in 1second intervals and also shows an average of the bit rate at 2Mbit/s. This is expected with an average packet length of 80000 bits and average time between packets of 40 ms.

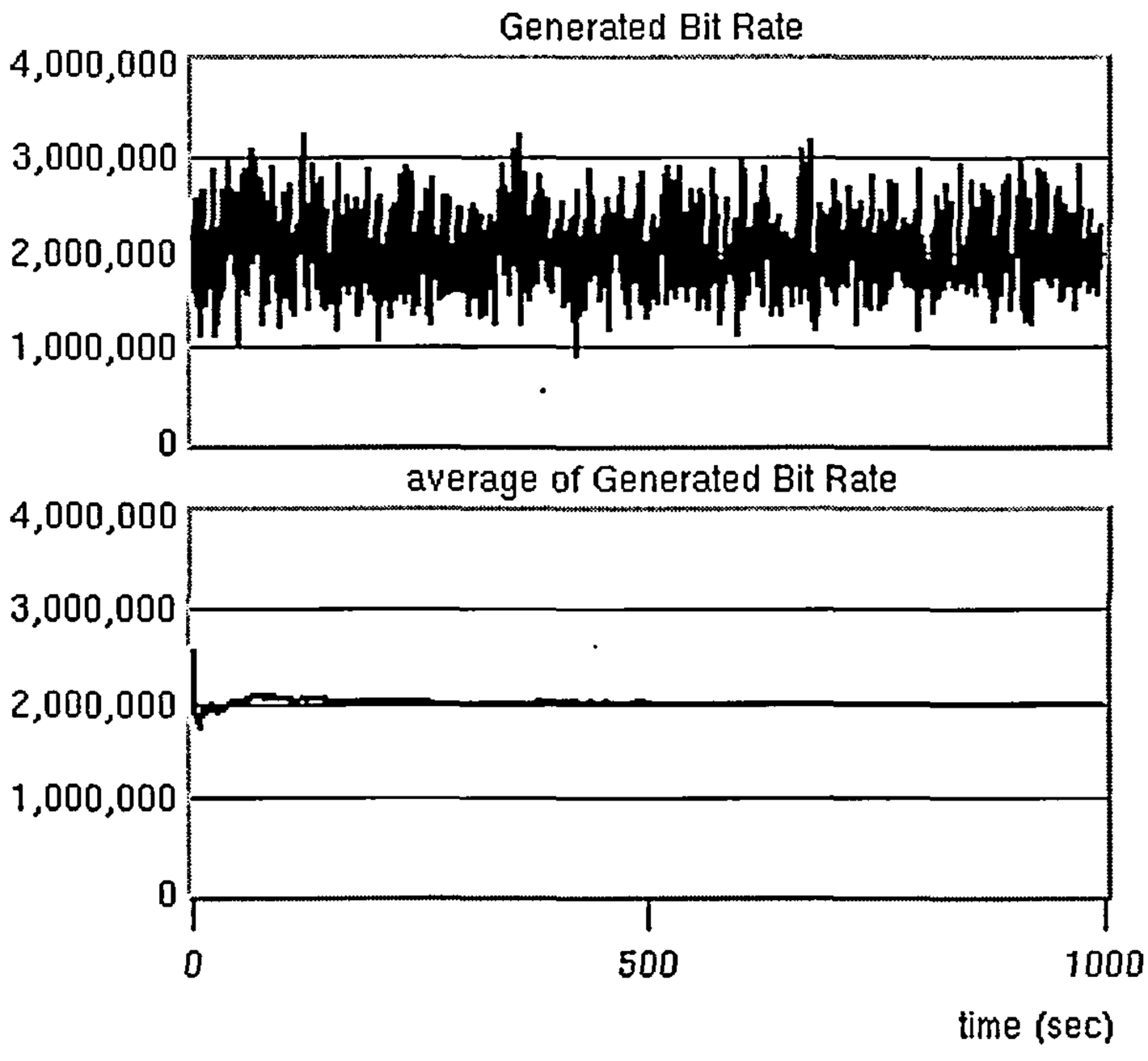

Figure III-2: Bit Rate Generated by Packet Generator

The queue process was acb_fifo queue that is distributed with OPNET™ Modeler with a slight difference. When the packet is enqueued, the length is obtained by looking up the size given in the packet header instead of the actual length of the packet being sent. This is because in the simulation a customised packet format was used for simplicity. The length of the packet was stamped as a header value. A few simple tests showed that the process reads the values correctly. The sink process was also built in OPNET™ and simply destroys the packets.

Testing the Models

The purpose of constructing the reference model was to determine the queue behaviour when there is no control in place. Then Hysteresis, RED and Percentile monitoring models will be set with the threshold set very high (*400,000,000* bits and *350,000,000* bits) such that the control algorithm does not activate. In all cases the gradual load increase scenario

is used and the service rate of the queue is set at 12 Mbit/s. The graphs in Figure III-3 show that the queue occupancy behaviour is identical. It was found that the value of the Packet length Reduction Factor (f) remained 1.0 which is the maximum.

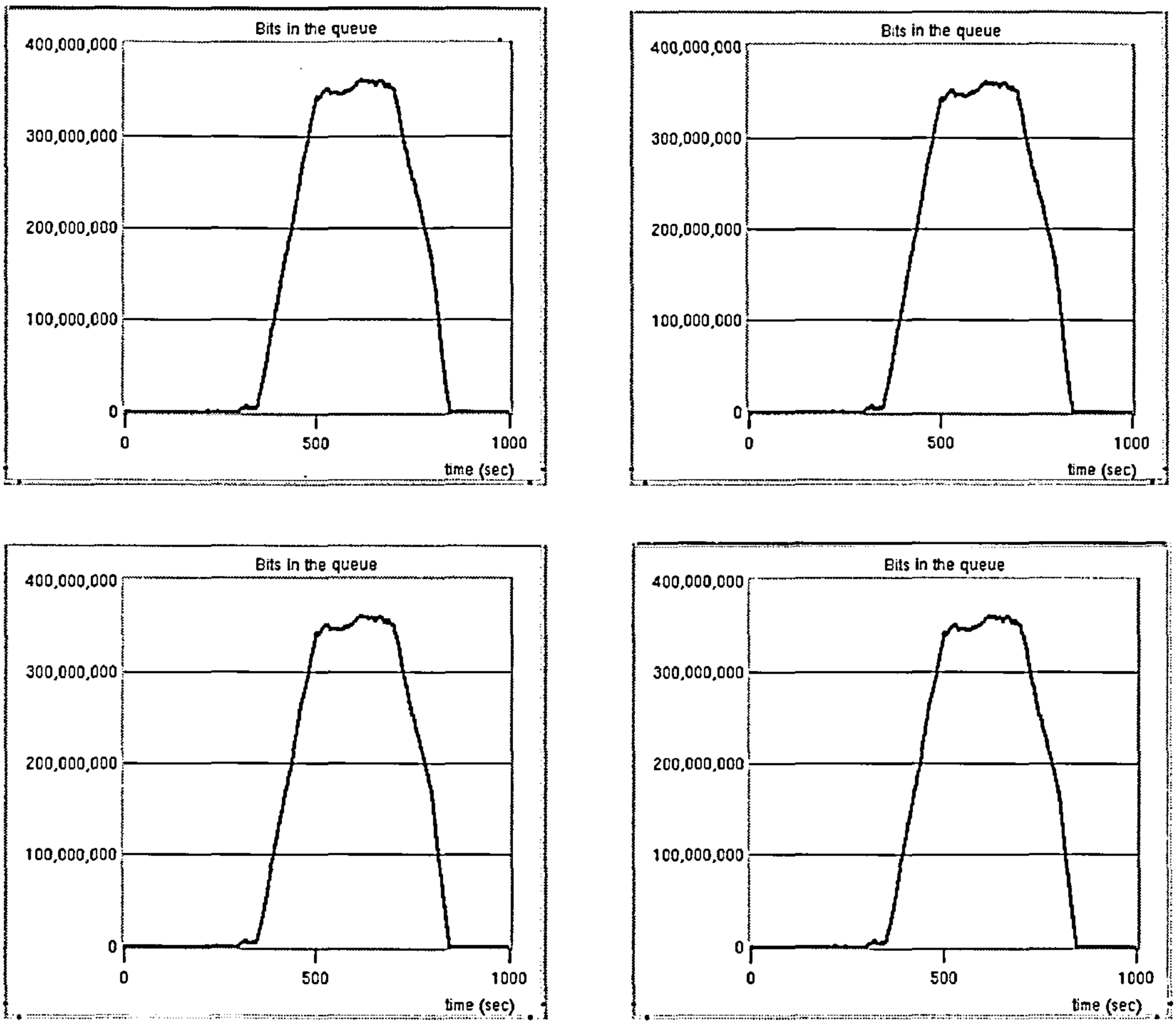

Figure III-3: Queue Occupancy with very High Thresholds

Top Left: No control ; Top Right: Hysteresis; Bottom Left: RED; Bottom Right: Percentile Monitoring

The test was repeated by changing the reference model so that every packet arriving at the bottleneck queue was reduced to half its length before being enqueued. This models a strict system. The control algorithms were run with thresholds set very low, at 2000 bits and 1000 bits for higher and lower thresholds respectively. This time it was found that in all

cases the value of f immediately drops to 0.5 (minimum) which effectively reduced the packet lengths to half the original value. The queue occupancy graphs are shown in Figure III-4. For clarity in comparison the graphs are shown with same scale although the initial values differ.

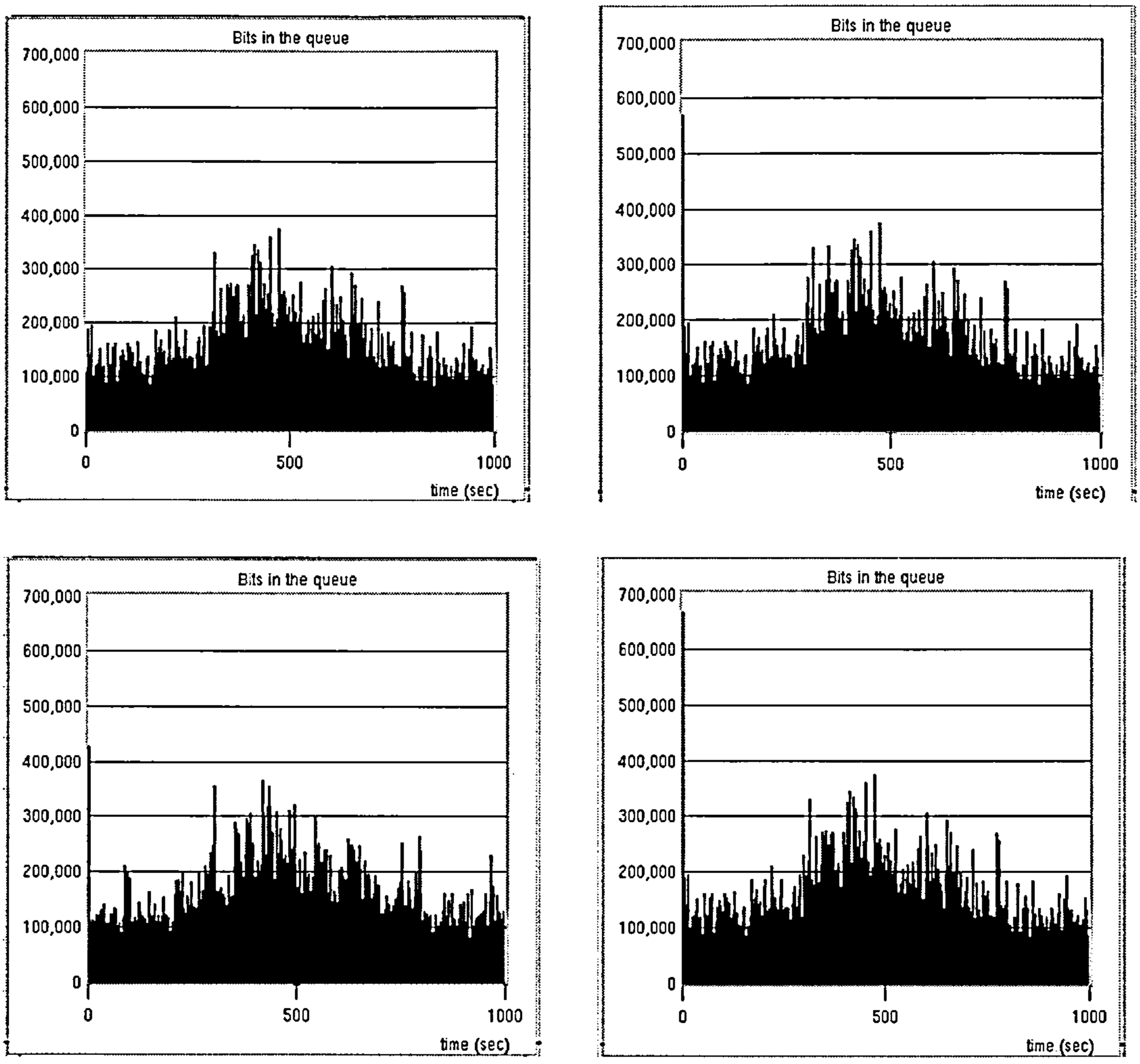

Figure III-3: Queue Occupancy with very Low Thresholds

Top Left: Strict control ; Top Right: Hysteresis; Bottom Left: RED; Bottom Right: Percentile Monitoring

As it is quite apparent the behaviour of the bottleneck neck is almost identical in each case when the control system is operating against extreme conditions and the performance degenerates to the reference models. Therefore, when the normal thresholds (450000 bits and 360000 bits) are restored, the difference in the queuing behaviour is purely due to the

performance of the algorithms. As shown in the graphs of Figure III-4, the queue builds up to varying lengths in each case.

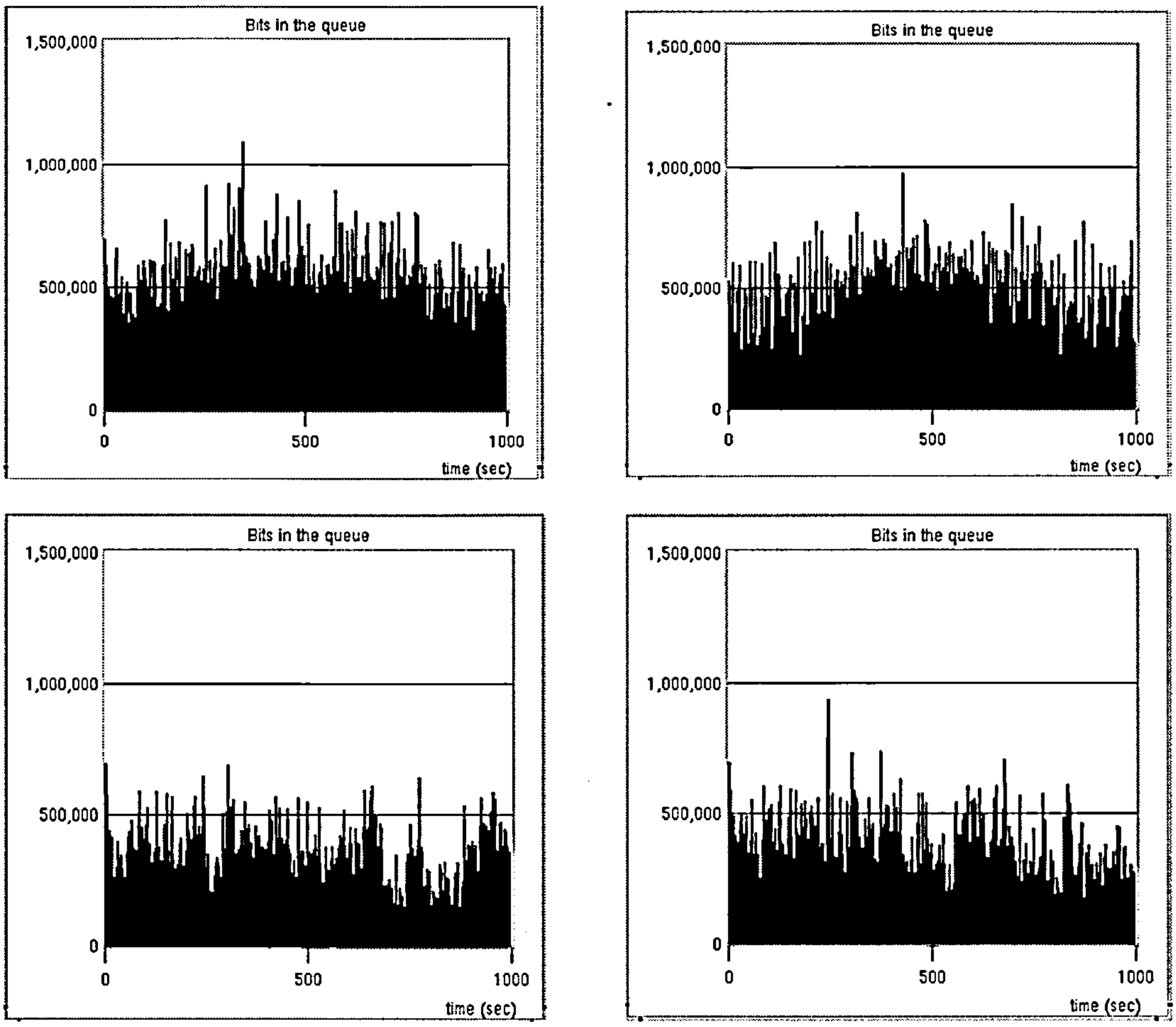

Figure III-4: Queue Occupancy with Algorithms in action with thresholds as normal

Top Left: Hysteresis ; Top Right: RED; Bottom Left: Percentile Monitoring with conservative settings; Bottom Right: Percentile Monitoring with liberal settings

In the main text we have focussed on the fluctuations in the value of f which results from the queuing behaviour. As we can see here, the queue occupancy is best controlled in case of Percentile Monitoring with both conservative and liberal parameter settings.

Appendix IV

Mean and Variance Calculations and Peak Rate Estimation

The mean and variance have to be calculated and the peak rate estimated at regular intervals, termed here as *calc_intrvl*. We use standard techniques to calculate these by a sampling method. The sampling has to be done more frequently than the calculations and the interval between successive samples is called *samp_intrvl*. The number of samples made within a calculation interval is called *num_samples*.

Consider,

Calc_intrvl = 1s, Samp_intrvl = 5ms and Num_samples = 200

We count the number of bits arriving in both these intervals and store them as $Num\_bits_{calc\_intrvl}$ and $Num\_bits_{samp\_intrvl}$.

$$Mean_{calc\_intrvl} = Num\_bits_{calc\_intrvl}$$

$$\Rightarrow Mean_{samp\_intrvl} = \frac{Mean_{calc\_intrvl}}{Num\_samples}$$

From the definition of variance,

$$Variance_{samp\_int rvl} = \frac{\sum_{i=1}^{Num\_samples} \left(Num\_bits_{samp\_intrvl} - Mean_{samp\_intrvl}\right)^2}{Num\_samples}$$

Assuming that all values would lie within mean +5*standard deviation, we estimate the peak rate as:

$$Peak_{samp\_intrvl} = Mean_{samp\_intrvl} + 5 * \sqrt{Variance_{samp\_intrvl}}$$

But, in order to estimate the peak rate over the *calc_intrvl*, we need the measure of variance over the *calc_intrvl*. We can infer that, using the same assumption as above,

$$Peak_{calc\_intrvl} = Mean_{calc\_intrvl} + 5 * \sqrt{Variance_{calc\_intrvl}}$$

And, it is also true that,

$$Peak_{calc\_intrvl} = Num\_samples * \left(Mean_{samp\_intrvl} + 5 * \sqrt{Variance_{samp\_intrvl}}\right)$$

The above two expressions can only be congruent if

$$Variance_{calc\_intrvl} = Variance_{samp\_intrvl} * Num\_samples^2$$

Appendix V

Numerical Proof of Fairness of Mean and Variance Pricing Algorithm

The flow controllers use the prices from the ingress router (to the carrier network) to ascertain the charge that the flow must pay for its mean and variance and consequently to adapt the mean and variance if the total charge exceeds the WtP. The prices must be fair such that the flow controller has complete flexibility to change its mean or variance without being penalised for its choice. For this to be true, it must be ensured that if flows reallocate their charge (or payment) without net increase, then the resultant aggregate peak (that is the peak rate at the pricing link) must remain the same.

The aggregate peak at the link is estimated by measuring the mean and variance of the aggregate flow arriving at this link and using Eq 10.3 rewritten here as:

$$Aggregate\ peak = m_A + k\sqrt{v_A}$$

where, m_A = aggregate mean,

v_A = aggregate variance

and, $k = 5$

Consider that there are 5 flows each with mean of 20 Mbits/s and variance 20 $Mbit^2/s^2$ aggregating onto a link with capacity 100 Mbits/s and network WtP of 40 tokens. Assume that the aggregate mean is sum of the mean rates of the flows and the aggregate variance is the sum of the flow variances.

$$m_A = 5*20 = 100\ Mbits/s \text{ and } v_A = 5*20 = 100\ Mbits^2/s^2$$

$$Aggregate\ Peak = 100 + 5*\sqrt{100} = 150\ Mbits/s$$

Then using the price equations derived in Section 10.4, rewritten here as:

$$P_m = \frac{WtP_{net}}{Capacity - Aggregate\ Peak} \text{ and } P_v = \frac{WtP_{net}}{Capacity - Aggregate\ Peak} * \frac{k}{2\sigma_A},$$

we calculate the mean price (P_m) and variance price (P_v) as follows:

$$P_m = \frac{40}{200-150} = 0.8\ tokens/bits/s$$

$$P_v = \frac{40*5}{(200-150)*2*5\sqrt{5*20}} = 0.2\ tokens/(bits/s)^2$$

Consider that one of the flow controllers, upon receiving these prices, works out the charge required to pay and finds that it is operating at its maximum WtP of 20 tokens and cannot increase the charge.

Option 1 -- Decrease Mean Charge Increase Variance Charge

The flow reduces its mean charge from 16 to 15 tokens and increases the variance charge from 4 to 5 tokens keeping the total unchanged at 20 tokens.

The resultant flow mean (m_i) and variance (v_i) will be as follows:

$$m_i = \frac{15}{0.8} = 18.75\ Mbits/s \text{ and } v_i = \frac{5}{0.2} = 25\ Mbits^2/s^2$$

Using the equations shown above to calculate the estimated aggregate peak, we calculate the aggregate now.

$$m_A = 4*20+18.75 = 98.75\ Mbits/s \text{ and } v_A = 4*20+25 = 105\ Mbits^2/s^2$$

$$Aggregate\ Peak = 98.75+5*\sqrt{105} = 149.984\ Mbits/s$$

Option 2 – Increase Mean Charge Decrease Variance Charge

The flow increases its mean charge from 16 to 18 and reduces the variance charge from 4 to 2. The resultant mean and variance of the flow, the aggregate mean and variance and the aggregate peak are calculated by repeating the process shown above.

We find that,

$$m_i = \frac{18}{0.8} = 22.5\ Mbits/s \text{ and } v_i = \frac{2}{0.2} = 10\ Mbits^2/s^2$$

$$m_A = 4*20+22.5 = 102.5\ Mbits/s \text{ and } v_A = 4*20+10 = 90\ Mbits^2/s^2$$

$$Aggregate\ Peak = 102.5+5*\sqrt{90} = 149.934\ Mbits/s$$

Option 3 – Extreme Case

The flow decreases its mean charge from 16 to 2 and increases the variance charge from 4 to 18. The resultant mean and variance of the flow, the aggregate mean and variance and the aggregate peak are as follows:

$$m_i = \frac{2}{0.8} = 2.5\ Mbits/s \text{ and } v_i = \frac{18}{0.2} = 90\ Mbits^2/s^2$$

$$m_A = 4*20+2.5 = 82.5\ Mbits/s \text{ and } v_A = 4*20+90 = 170\ Mbits^2/s^2$$

$$Aggregate\ Peak = 82.5+5*\sqrt{170} = 147.69\ Mbits/s$$

Hence, we see that by choosing to give priority to mean or variance (i.e., Option 1 or 2) the resultant aggregate peak is equal to the initial value within the bounds of numerical error. Even in the extreme case, where the flow controller changes the allocation drastically the effect on the aggregate is not significant.

Appendix VI

OPNET™ Screenshots

Following are some screenshots of the network model simulated in OPNET™. These screenshots are taken from the price based simulation model but essentially the models used for evaluation of Hysteresis, RED and percentile monitoring were similar.

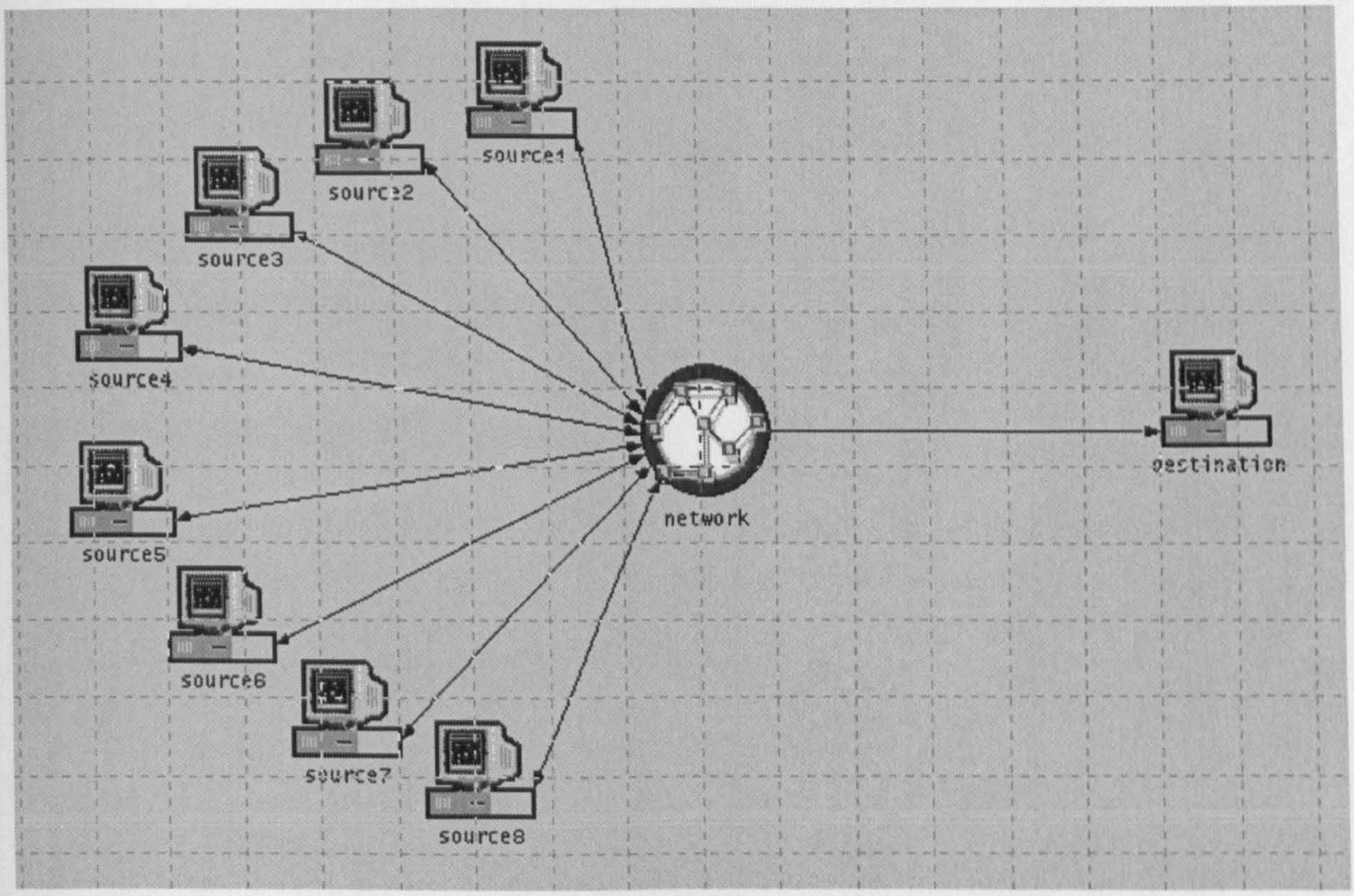

Figure V-1: Top Level Model of Simulated Network

Figure V-1 shows sources (of individual flows), network with aggregating ingress router and destination. The icons used here were chosen from a selection available in OPNET™ and hence are only closest matches.

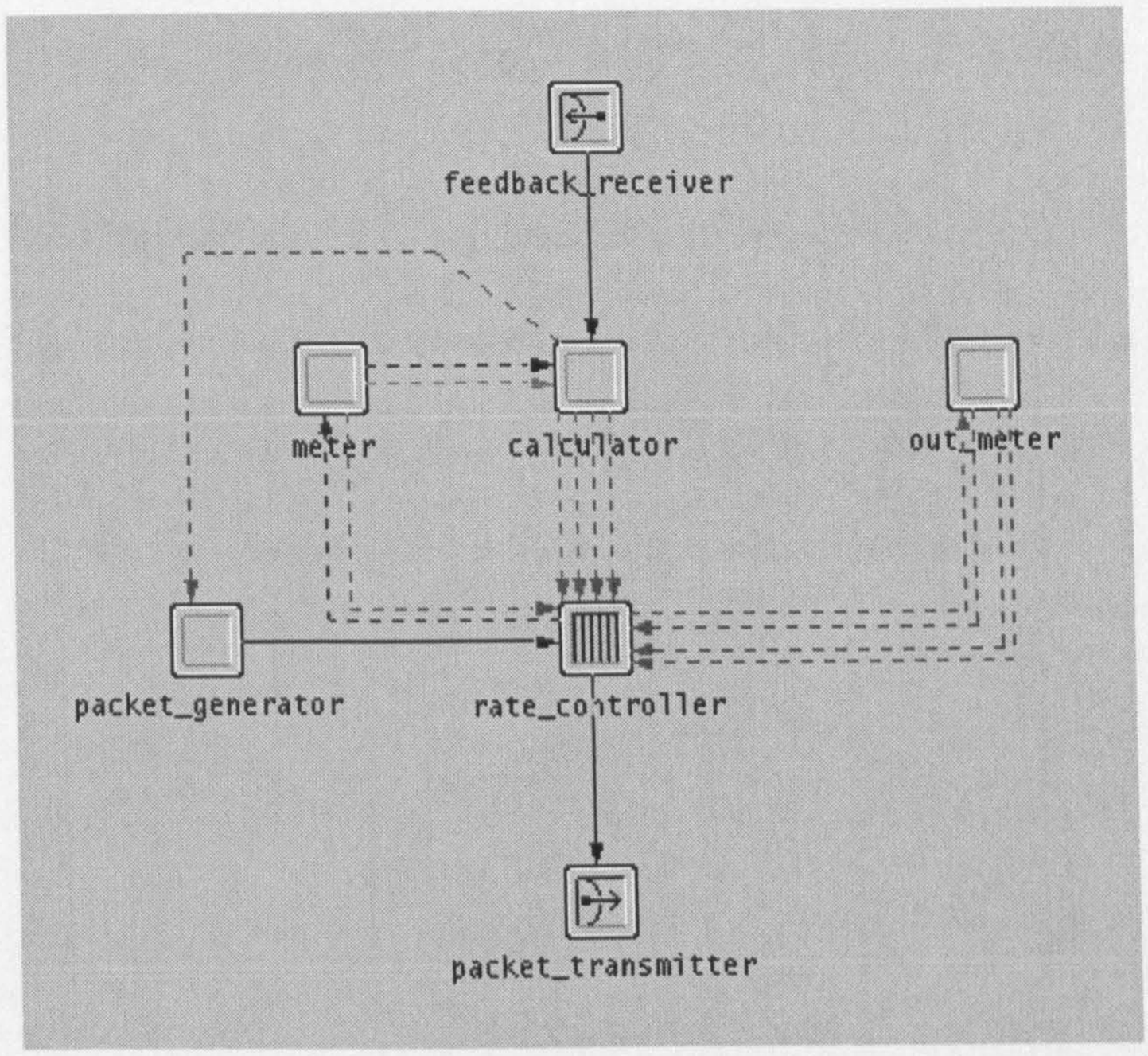

Figure V-2: Node model of the source

Figure V-2 shows the node model of a source (see Figure V-1). Other nodes such as the network ingress router and destination also have their corresponding node models. Here, only the source node is shown as an example. The source node contains a packet generator, target calculator and rate controller with meters at input and output of the rate controller queue. The two links labelled as feedback receiver and packet transmitter are used for receiving prices from the network and sending packets to the network respectively. The solid lines show the paths of the packet stream whereas the dotted lines show the flow of statistics for control and for producing results.

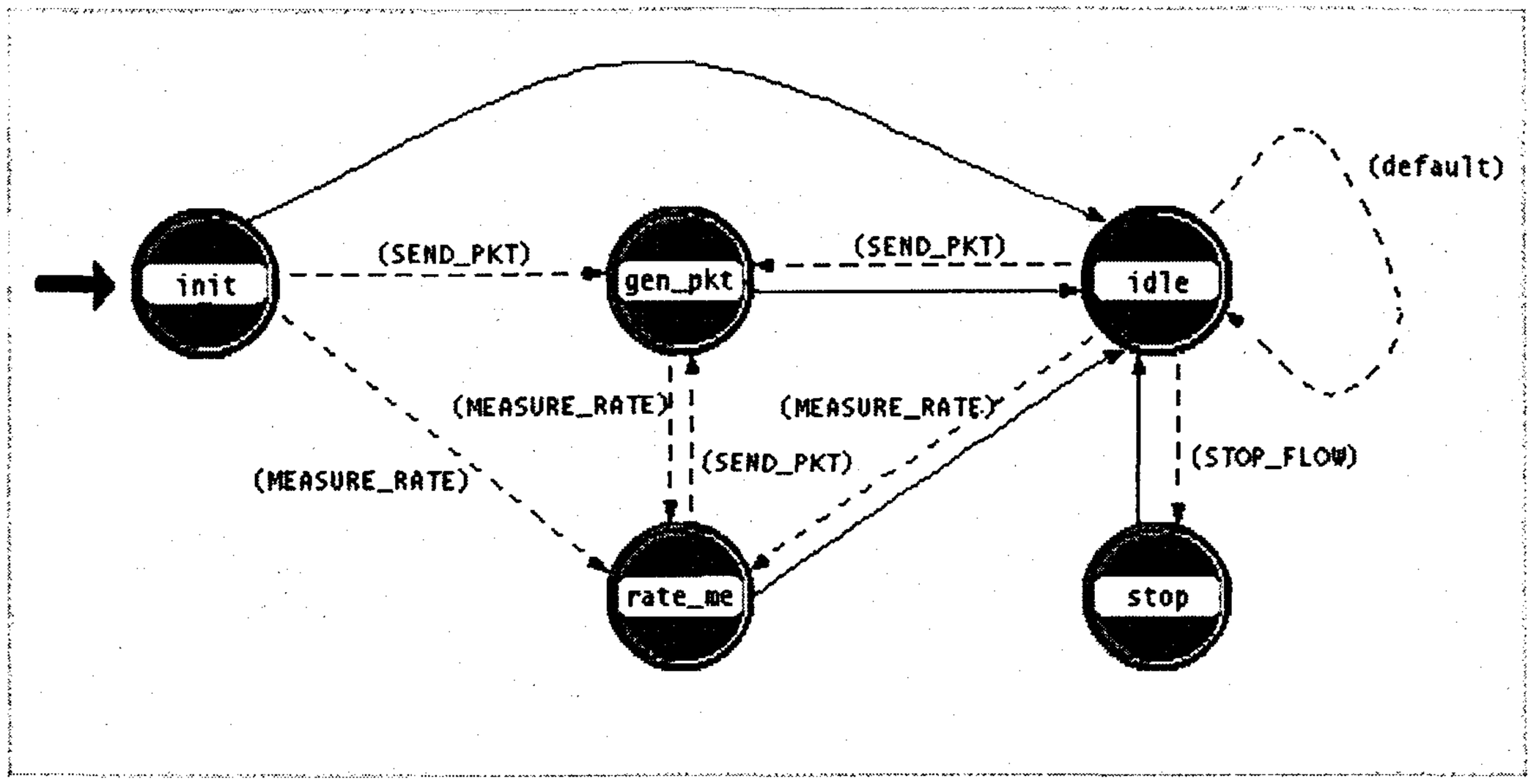

Figure V-3: Process model of Packet Generator

Each component of a node model has an associated process model. The same process model can be used by a number of nodes. The process model of the packet generator is shown in Figure V-3. The process model is essentially a state machine diagram. Any function that needs to be executed in a state has to be written in a version of the C language which is optimised for OPNET™. The lines between two states show transitions from one state to another. The dotted lines indicate that the transition is conditional whereas the solid lines indicate the transition is default. It is possible for the process to loop around and wait in a state (here the "idle" state) provided it is configured as an "unforced" state.

The OPNET™ screenshots presented here are simply to give a very brief illustration of the models that were constructed in OPNET™ and by no means include all the processes involved in the simulations.