# A Virtual Loss-Load Congestion Control Strategy for High Speed Networks

A Thesis Submitted to the College of

Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in the

Department of Computational Science

University of Saskatchewan

Saskatoon, Saskatchewan

by

Narayanan Prithviraj

July, 1994

## PERMISSION TO USE

## ACKNOWLEDGEMENTS

A special thanks to my supervisor Dr. Carey Williamson for providing the primary motivation and idea behind this thesis. But for his encouragement, and his effortless yet thorough ability to wade through those many drafts, this thesis would not have been completed on time.

I'm grateful to *TR*labs for their financial support and the fine facilities offered to me. The many stimulating cups of coffee I enjoyed there certainly did contribute to raising my productivity that extra bit.

Thanks are also due to members of the DISCUS group for the many interesting sessions held each week, and to the other members of my thesis committee: Dr. Rick Bunt, Dr. Winfried Grassman and Dr. Surinder Kumar (external) for their insightful comments on my work.

I appreciate the efforts of Dr. Derek Eager as Graduate Chairman, and of the department's secretarial staff for the ready help they provided whenever the need arose. I'd also like to thank the numerous friends who helped me survive two Prairie winters, and helped make my stay in Saskatoon a pleasant and memorable experience.

Finally, I would like to dedicate this thesis to the memory of my brother, who was, and remains, a constant source of inspiration in my life.

# A Virtual Loss-Load Congestion Control Strategy for High Speed Networks

*Candidate:* Narayanan Prithviraj

*Supervisor:* Carey L. Williamson

Master of Science Thesis

Presented to the College of Graduate Studies and Research

University of Saskatchewan

July, 1994

## ABSTRACT

Congestion control is critical in the design, deployment and operation of future high speed networks. Thus, several congestion control strategies have been proposed in the literature. These strategies may broadly be classified as preventive or reactive strategies.

The objective of this thesis is to evaluate a hybrid congestion control strategy called the virtual loss-load model. The approach combines the leaky bucket traffic shaper (a preventive congestion control mechanism) with the loss-load model (a reactive congestion control mechanism). Simulation is used to evaluate the virtual loss-load model, and to compare its performance to that of five other reactive congestion control strategies. The evaluation is done using a benchmark suite of network scenarios proposed by Kanakia, Keshav and Mishra. Three main performance metrics are used in the evaluation: file transfer time, mean cell delay, and cell loss probability.

The simulation results show that the virtual loss-load model is an effective congestion control strategy, providing performance comparable to (or superior to) several other reactive congestion control strategies. While transient effects due to dynamic changes in cross traffic result in significantly higher cell loss than expected in the virtual loss-load model, the parameters of the virtual loss-load model make it easily tunable for good performance.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Recent advancements in fiber optic technology and switching techniques have provided an opportunity for the deployment of high bandwidth communication networks on a commercial basis. Designed to operate in the gigabits per second range, Broadband Integrated Services Digital Networks (B-ISDN) will provide residential and business customers an unprecedented assortment of capabilities to provide voice, data and video services [1, 15]. According to CCITT recommendation I.121, Asynchronous Transfer Mode (ATM) will provide the transport mode in these networks [7, 43]. ATM is a low delay, high bandwidth statistical multiplexing technique where all traffic is carried in fixed size packets called "cells".

Voice, data and video applications all have different Quality of Service (QOS) requirements, such as the amount of cell delay and loss tolerable [30, 41]. Certain applications, such as those carrying voice and video traffic, are delay-sensitive but loss-insensitive. Others, such as data traffic applications (e.g., file transfer and electronic mail applications) are loss-sensitive, but delay-insensitive. Still others, such as interactive graphics or interactive computing applications, are both loss-sensitive and delay-sensitive. The challenge lies in the efficient design of the underlying network to support the diverse performance requirements of different types of traffic applications.

Congestion in B-ISDN networks is likely to be the dominant problem that needs to be overcome by the network designers [42]. Congestion in a network occurs whenever the demand on a network resource, such as the bandwidth of links, buffer space or processing capacities of intermediate switches, is exceeded. The persistence of congestion at intermediate switches for even small periods of time may lead to many cells being dropped due to the high delay-bandwidth product (or the number of cells 'in flight'). Congestion left uncontrolled leads to serious consequences such as long transfer delays and high cell loss. As a result, the network will be unable to meet the stringent QOS requirements of the emerging applications. Controlling congestion is therefore essential in broadband networks.

## 1.1 Congestion Control Strategies

Congestion control strategies can be classified broadly as *preventive* or *reactive* [40]. Preventive approaches require an end-to-end allocation of resources before the commencement of data transfer. In reactive schemes, feedback received from the network or the receiver is used to alter the sources' sending strategy while a connection is active.

Preventive congestion control strategies may operate at the call-level or the cell-level. *Admission Control* [41] is a preventive congestion control procedure where the network decides, at the time of call arrival, whether to accept or reject a new connection. In making this decision, the network uses the *traffic descriptor* of the connection that characterises the nature of the traffic generated by the connection. The mean cell rate, peak cell rate and the maximum burst length are some of the commonly accepted parameters in a traffic descriptor [31]. It is the responsibility of the source to ensure that the cells released by it conform to the traffic descriptor declared to the network. *Traffic Shaping* [4, 5, 13] is a cell-level control mechanism employed by the source in enforcing its declaration. The *leaky bucket* [35] is an example of one such traffic shaping approach. Cells are buffered at the sender and released into the network at a controlled rate. The scheme smoothes out bursts, helping to reduce network congestion.

The leaky bucket strategy shapes traffic effectively and is simple to implement. Furthermore, the parameters determining the operation of the leaky bucket are easy to control and can be set based on the traffic descriptor of a connection. However, since the parameters are set statically at the time of call admission, the basic leaky bucket model tends to operate conservatively [24] since it may not exploit available network capacity optimally. Although a variation, the *virtual leaky bucket* [14], does allow senders to operate at higher rates, no information exists at the sender about the expected loss rate of the extra cells released into the network.

Reactive congestion control strategies adjust sender transmissions dynamically based on feedback received from the network. Reactive schemes are either *window-based* or *rate-based* [18]. Window-based mechanisms, which specify only the maximum number of packets that can be transmitted at a time, have been used in traditional protocols, such as the Internet family of protocols. Rate-based approaches, which specify both the number of packets and the time interval of the transmission, are however gaining popularity in broadband networks, as they tend to be less bursty than window-based schemes. In addition, since some of the broadband applications such as those based on voice and video are inherently rate-based, the adoption of a rate-based scheme is more natural. *Loss-Load Curves* [38] are an example of a rate-based reactive congestion control scheme where switches provide congestion feedback to senders. Senders dynamically adjust their rate of operation based on the feedback information provided by the loss-load curves.

The advantage of loss-load curves is that they provide a concise mathematical representation of aggregate load in the network. The feedback information provided by loss-load curves allows senders to react dynamically to changing network conditions. Senders are provided with precise knowledge of the expected loss rate of their traffic within the network, allowing them to control the load they generate on the network. Loss-load curves are, however, aggressive in their operation, and at equilibrium generate a load greater than the network capacity, causing excessive cell loss.

## 1.2 Objectives

The objective of this thesis is to explore the concept of a "virtual loss-load model" that combines the preventive leaky bucket scheme with the reactive loss-load approach. The idea is to exploit the advantages and to eliminate the disadvantages that each of the schemes possesses in isolation. The virtual loss-load model potentially offers the following advantages:

- *Simplicity and Controllability*: By employing the basic leaky bucket scheme at the network edges, senders continue to exploit the simplicity and controllability of the leaky bucket model. The virtual loss-load model also contains a controlling parameter, the *virtual factor* $v$. By choosing appropriate values of $v$, the virtual loss load model can be tuned to satisfy the performance requirements of different types of traffic.

- *Increased Network Throughput*: Since senders use loss-load curve feedback in adjusting their rate of operation, senders can exploit available network capacity depending on dynamic network conditions, offsetting the conservative nature of the basic leaky bucket.

- *Reduced Network Overload*: Switches in the network use the concept of virtual capacity while implementing loss-load computations. Virtual capacity computations allow the virtual loss-load model to operate at close to network capacity, without overloading the network, thereby overcoming the aggressive nature of operation of the original loss-load model.

- *Sender Control*: The virtual leaky bucket allows senders to operate at higher rates of operation by 'colouring' (tagging) violation cells that are released into the network, thereby providing two levels of cell priority: 'green' cells (conforming cells), which are guaranteed delivery, and 'red' cells (violation cells), which are delivered on a space-available 'best-effort' basis with no guarantee at all. The 'colouring' strategy is incorporated in the virtual loss-load model as well. The fraction of cells coloured is chosen so as to match the cell loss probability associated with the current rate of cell release, and is calculated using loss-load feedback. The colouring strategy used in the virtual loss-load model offers three primary benefits. First, it provides a predictable bound on the number of red cells in the network. Second, senders can decide which cells need to be coloured, thereby allowing senders a greater degree of control over the colouring process. Third, senders are provided with a good guarantee that the cell priority that they select will be honoured by the network. That is, the virtual loss-load model provides a more predictable bound on the delivery or non-delivery of cells of each class.

The virtual loss-load model is intended for traffic that does not have stringent real-time performance requirements. Such traffic, also known as *reservationless* or *datagram* traffic, [42] is assumed to be delay-insensitive and/or loss-insensitive. Traffic belonging to this category, for example, requires only a 'best-effort' by the network, without any performance guarantee. In ATM networks, for example, a new service class known as Available Bit Rate (ABR) traffic has been created to identify traffic belonging to this category.

Simulation is used in this thesis to study the performance of the virtual loss-load model. Experiments were conducted based on a benchmark suite developed by Kanakia *et al.* [19]. In addition to testing the efficacy of the virtual loss-load model, the benchmark suite also provides for comparison of the model with other congestion control strategies.

The particular research questions that are addressed in this thesis are:

- How do the factors controlling the virtual loss-load model affect its performance?
- To what degree does the virtual loss-load model improve on the original loss-load model?
- How does the performance of the original loss-load model and the virtual loss-load model compare with that of other congestion control strategies described in the literature?

## 1.3 Thesis Outline

The remainder of this thesis is organized in the following manner:

Chapter 2 presents background material on B-ISDN networks. It explores emerging applications and their QOS requirements.

Chapter 3 explains various congestion control techniques in detail and reviews some of the commonly used congestion control strategies. The chapter concludes with background material on two of the components of the virtual loss-load model: the leaky bucket traffic shaper and the loss-load model.

Chapter 4 describes the virtual loss-load model and simulator in detail. It explains the advantages the virtual loss-load model offers, and illustrates the approach for a simple two-sender traffic scenario.

Chapter 5 explains the various experiments that were performed using the virtual loss-load simulator. The experiments investigate how the controlling parameters in the virtual loss-load model need to be set to satisfy different traffic performance requirements. The experiments compare the virtual loss-load model with the original loss-load model and with other feedback-based congestion control strategies.

Chapter 6 provides a summary of the results presented in this thesis, and suggests directions for future work.

# Chapter 2

# Background

Broadband networks represent a revolutionary advance in communication technology. This chapter presents background material on broadband networks. It explores the types of applications that are likely to shape the future of broadband networks, and examines the QOS requirements of these applications.

## 2.1 High Speed Network Evolution

Communication networks have undergone dramatic changes in the past two decades. These changes have occurred both in the protocols as well as in the switching techniques underlying communication technology. Current packet network protocols are influenced by *fast packet switching* [36]. Fast packet switching is based on the fact that transmission links have become increasingly reliable over the years. Switches need to operate at higher speeds due to the increase in available bandwidth. There is no need to perform error recovery procedures on a link-by-link basis; instead it can be performed on an end-to-end level. Switches based on fast packet switching have the primary tasks of connection management and packet routing using streamlined protocol processing. *Frame relay* [26, 27] networks based on fast packet switching allow variable length packets to be transmitted. The main application in frame relay networks has been that of local area network (LAN) interconnection. *Cell relay* [3] is a variation of fast packet switching wherein all packets are transmitted in fixed size units known as “cells”. Cell relay is likely to be the dominant transport mode in the emerging B-ISDN networks.

Fiber optic technology has enabled the communication bandwidth to jump from the kilobits or megabits per second to the gigabits per second level in the present period. This represents a massive leap forward and has resulted in the research and development of B-ISDN and Asynchronous Transfer Mode (ATM) networks. Unlike earlier generations, switches now represent the interconnection bottleneck rather than the links. Switches need to process packets just above the physical layer so as to realise the speeds at which they need to operate.

The evolution of high speed networks is not restricted to the wide area network (WAN) environment, although that still represents the most dramatic advance in communication technology. The Switched Multimegabit Data Service (SMDS) [44] operating over the IEEE Distributed Queue Dual Bus (DQDB) [21] at 45 Mb/s and 150 Mb/s represents efforts at the Metropolitan Area Network (MAN) level. Gigabit LANS have also emerged. An example is the High Performance Parallel Interface (HIPPI) [45] protocols that are popular in supercomputing centers and support data rates of 800 Mb/s and 1.6 Gb/s. The ATM standard has also received attention in the LAN environment in addition to the public WANs for which they were originally targeted. The ATM standard is now seen as a scalable method for connectivity to the wide area network and poses serious competition to the deployment prospects of other LAN and MAN standards.

## 2.2 B-ISDN Applications

The success of the B-ISDN network will ultimately be driven by the end applications. The B-ISDN architecture allows for combining voice, data and video traffic over a single network. The flexibility gained from this approach allows for the realisation of several applications which were not possible in previous network architectures. The following lists some of the likely applications [1]:

For business subscribers:

- High speed LAN-to-LAN interconnection offers the ability to access large data files from remote servers. The increased bandwidth available at the wide area network level allows for the extension of traditional LAN services to a larger geographical area, rendering an organization's network transparent.

- In the medical field, B-ISDN could be used for the transmission of large data files containing medical images such as those created by tomographic imaging of digitised X-ray images. By combining file transmission with video conferencing, multiple medical opinions can be garnered resulting in improved medical care.

- Multimedia conferencing is expected to be a major application, as it would allow such techniques as the joint editing of documents within a work team together with video conferencing. This cuts down the cost of business travel and makes more productive use of the work team's time. Video conferencing can also be used to provide interactive teaching from a remote industrial/academic site to the employees/students at a different location.

For residential subscribers:

- Video-on-demand is a popular application wherein subscribers may retrieve any video from a video library "on demand", both for educational and entertainment purposes. The challenge is to provide the service at a cost that is comparable to existing video rental charges.

- Tele-shopping is an application that combines video retrieval with transaction shopping. This is a video equivalent of catalogue shopping whereby the customer can have on-demand access to a video presentation of the chosen product range.

## 2.3 Classification of Services

With the multitude of possible applications, it is useful to classify the applications within a broader framework. CCITT recommendation I.121 identifies two broad classes of service, *interactive* services and *distribution* services, each of which is further divided into service *classes* [43].

Interactive services consist of the following service classes:

- *Messaging Services*: Messaging services imply the use of store-and-forward mechanisms that may allow some form of processing function. Messaging services are unidirectional in nature, and include electronic mail, voice mail and video mail applications.

- *Conversational Services*: Conversational services are generally bidirectional and real-time with no store-and-forward mechanism. Examples of this service are video telephony and video conferencing. Terminal data access is another example.

- *Retrieval Services*: Retrieval services allow for information to be stored in a central library and transmitted to a particular user on request. An example of such a service is video-on-demand.

Distribution services provide a continuous flow of information from a central source to an unlimited number of users. They are divided into the following classes:

- *Distribution services without user control*: Here, an individual can access the service whenever he or she wishes, but cannot control the start of the information transmission. Broadcast television is a good example of this service.

- *Distribution services with user control*: When the service offers user presentation control, the information from the central source is transmitted as a cyclical repetition of information entities (e.g., pictures and frames), and the user can control his or her access to the information flow so that the entities selected will appear from the beginning of the sequence. Teletext is a current example of this service.

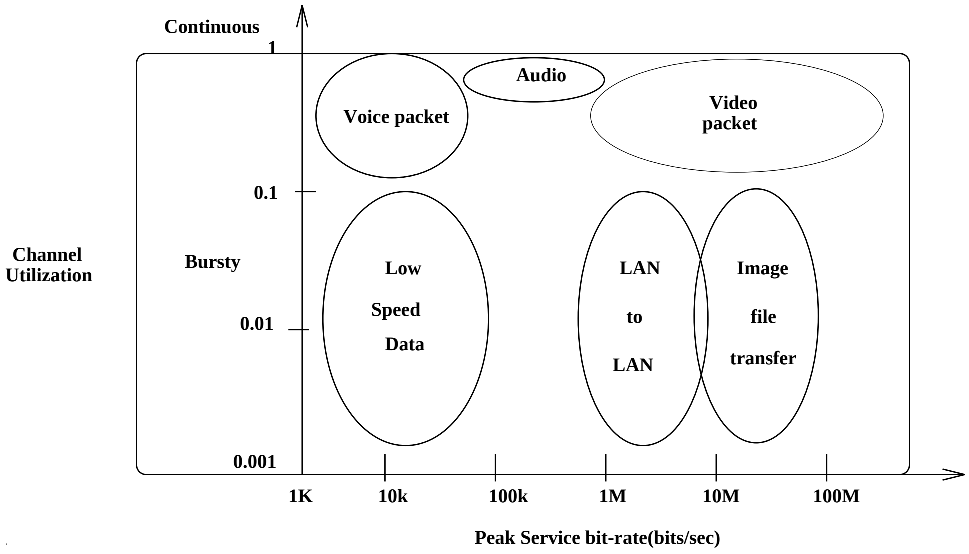



Figure 2.1: Approximate B-ISDN Traffic Characteristics [41]

## 2.4 Quality of Service

The applications and services that B-ISDN networks are required to support have very different traffic characteristics and performance requirements. Figure 2.1 depicts traffic characteristics of typical broadband applications by plotting the source utilisation (fraction of time during the connection for which the source actively transmits packets) versus the peak bit rate during such periods for typical broadband applications [41]. Voice, audio and video traffic sources all have high source utilizations, but vary in their bandwidth requirements. Data traffic, LAN traffic, and image traffic cover a wide range of bit rates. The need for statistical multiplexing is apparent in the case of these traffic types, as their source utilizations vary across a vast range, indicating burstiness.

The guarantees required by the applications from the network to achieve adequate performance are known as the Quality of Service (QOS) parameters. A few of the possible QOS parameters are [6]:

- *Cell Loss Probability*: Cell loss may occur in an ATM network due to buffer overflows or due to uncorrectable header errors on transmission links. Cell loss cannot be avoided completely within the network because of the statistical nature of the traffic. Cell loss probability (also called cell loss ratio in the literature) is the probability that a cell is discarded in an end-to-end connection.

- *Mean Cell Delay*: This is the mean time required for a cell to travel across the network from source to destination. The delay includes the accumulated processing time within the switches and the accumulated waiting times in the queues of the switching stages.

- *Cell Delay Variance*: Also known as *cell jitter*, this parameter is intended to measure the deviation of cell delay from the mean. Low cell delay variance is normally desirable, since cell arrivals tend to be smoother, more predictable and centred around the mean cell delay.

- *Bit Error Probability*: The parameter measures the probability that a bit in the information field is incorrect.

- *Service Bit Rate*: This is the rate at which cells are inserted by a connection. The connection requires a guarantee that the network will be able to support this rate.

## 2.5 Summary

Advances in transmission, switching and computer technologies have driven networks to move into an era of high bandwidth communication systems. Packet switching is the favoured technology for integrating transmission and switching resources. Fast packet switching will be the technology of choice for broadband networks to support the diverse traffic requirements efficiently. The new applications being developed will ultimately determine the success of B-ISDN networks.

# Chapter 3

# Congestion Control

Congestion control is one of the foremost problems to be addressed in the design, deployment, and operation of future high speed networks. This chapter presents background material on the congestion control problem, provides a brief summary of existing approaches to congestion control, and then presents a detailed description of the two congestion control approaches central to this thesis, namely, the leaky bucket traffic shaper and the loss-load model.

## 3.1 Congestion and Congestion Control

Congestion occurs in a network whenever the demand on a network resource, such as the bandwidth of links, buffer space, or processing capacities of intermediate switches, is exceeded. Congestion in a network often results in increased packet delay (due to queueing in the network), increased packet loss (due to buffer overflow), and reduced network throughput (due to packet loss and retransmission). Much like congestion on highways and freeways (i.e, traffic jams), congestion in computer networks results in poor network throughput, degraded service, and dissatisfied users.

The problem of congestion in B-ISDN high speed networks is more severe than in conventional networks for several reasons. First, as network speeds increase, the peak rate of cell arrivals tends to increase faster than the average rate, contributing to a marked increase in the burstiness of traffic in high speed networks [23]. Since connections in a B-ISDN network are expected to be multiplexed statistically to take advantage of the variable information transfer rates generated by many applications and to maximise bandwidth usage, there is a finite probability that the offered load at a multiplex point will exceed the output capacity, particularly when all connections generate bursts simultaneously. This overload can result in congestion, causing cells to be dropped.

A secondary reason for congestion is network interconnection [17]. Existing local area networks will likely be connected to high speed backbones or B-ISDN links. Congestion can result at the intermediate router where traffic from the high speed link may need to be routed to a slower speed LAN. Because of a mismatch in network speeds, router buffers may fill up, causing buffer overflow and cell loss.

Uncontrolled congestion in B-ISDN networks can result in dire consequences as the QOS requirements of a connection may not be satisfied. A voice call could experience a degradation in speech quality, whereas a constant bit rate application could lose synchronisation due to lost packets at the time of congestion. A data application would cause packets to be retransmitted when packet loss is detected. The retransmission could in turn lead to further congestion, resulting in further packet loss and delay. Congestion could result in distortion of the image signal for video connections. Also, a single connection responsible for causing congestion could adversely affect the performance of other active connections causing an overall degradation in the QOS of all connections. Congestion control strategies are therefore indispensable for the proper operation of B-ISDN networks.

The following lists some of the desirable features of a congestion control strategy [9, 17, 18]:

- *Fairness*: The congestion control strategy should be fair to all users. At the time of congestion, the available shared resources should be allocated equitably amongst the competing users, based on priority or need, or whatever other criterion is deemed appropriate.

- *Responsiveness*: The congestion control scheme should match the demand dynamically to the available capacity. Since the available capacity in a network varies dynamically as connections start or stop, or as links go up or down, a congestion control scheme must be able to respond to load changes, both increases and decreases.

- *Robustness*: The scheme should be able to function under varying network conditions such as transmission errors, out-of-sequence packets, misbehaving users or deadlocks.

- *Optimality*: The scheme should guarantee the performance requirements of the connections, while operating close to network capacity.

- *Low Overhead*: The nodes involved in the congestion control strategy should not consume too many processor cycles while implementing the strategy. If a congestion control scheme is too complicated or too slow, this will only increase the congestion problem.

## 3.2 Congestion Control Approaches

A comprehensive congestion control scheme consists of control policies that operate on various time scales ranging from distributed real-time controls to long-term networking procedures, as shown in Figure 3.1. The following subsections consider these policies in greater detail.

### 3.2.1 Short-Term Controls

Controls that operate on the short end of the time operating scale (close to the transmission time of a packet or cell) are referred to as short-term controls. There are various mechanisms for implementing short-term controls.

*Traffic shaping* is a short-term approach wherein traffic cells are released at a controlled rate at the endpoints of the network. Shaping the traffic makes the traffic smoother and more predictable, thereby reducing congestion within the network.

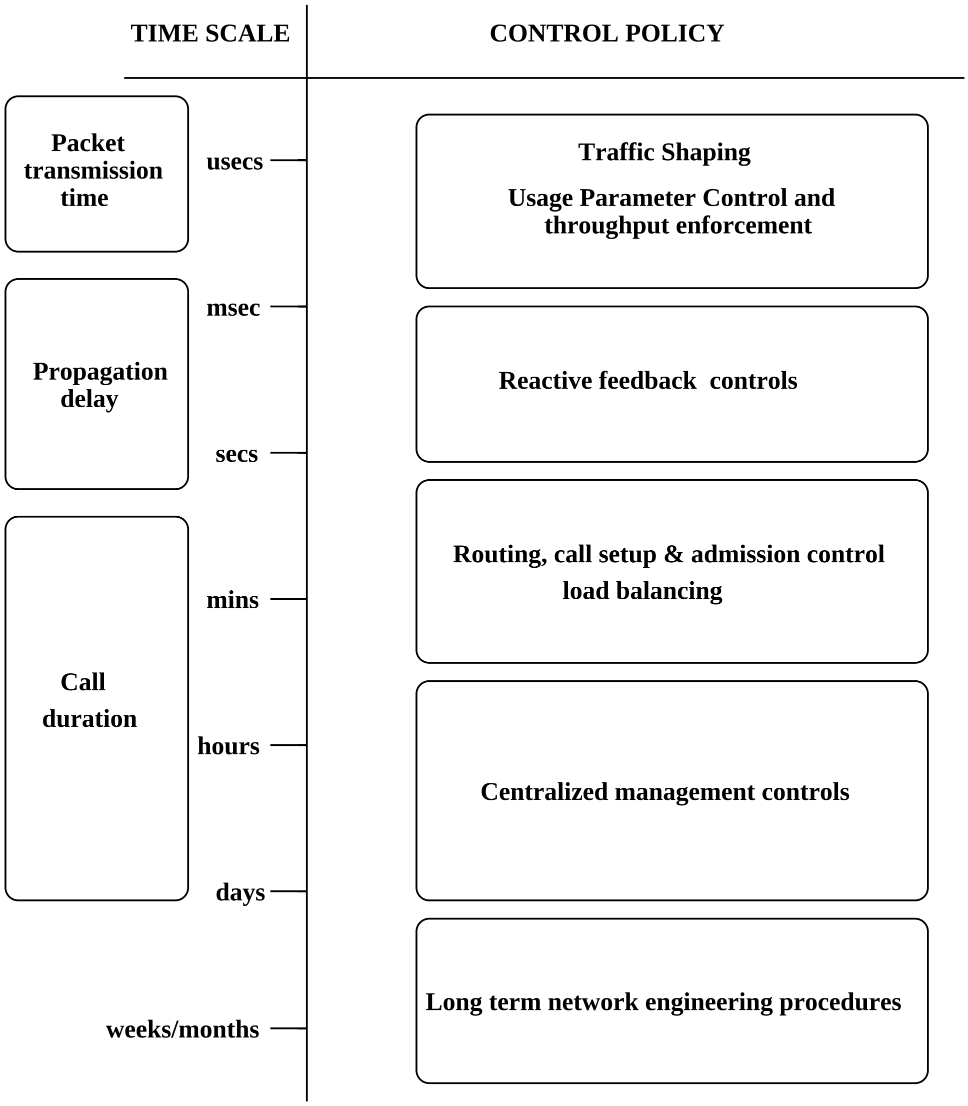



Figure 3.1: Congestion Control Approaches [3]

Several traffic shaping mechanisms have been proposed, such as the jumping window [8], the exponentially weighted moving average (EWMA) [34], and the leaky bucket [2, 14, 24, 35]. The basis for all these schemes is the desire to enforce the rate at which traffic cells are released into the network.

The jumping window approach limits the maximum number of cells ($N$) that can be accepted from a source within a given time interval or window. The first $N$ cells arriving in a given time interval are allowed to enter the network. Excess cells arriving within this interval are dropped. The new interval begins immediately following the preceding interval.

The EWMA scheme uses consecutive fixed time windows similar to the jumping window. However, the number of accepted cells in the $i$th window ($N_i$) is a function of the allowed number of cells per interval ($N$) and an exponentially weighted sum $S_i$ of the number of cells $X_i$ in the preceding intervals.

The leaky bucket mechanism is the most common scheme for traffic shaping due to the simplicity and flexibility of its implementation. The leaky bucket allows for the injection of cells into the network at a controlled rate thereby shaping the traffic. The leaky bucket and a variant, the virtual leaky bucket, are described in greater detail in Section 3.3.1.

*Usage parameter control* is a short-term approach where the control mechanisms reside within the network, at the switches, rather than at the sender. The goal of this approach is to police the arriving cells to make sure that the rate of arrival of these cells does not violate previously negotiated values. The main purpose is to protect network resources from malicious as well as unintentional misbehaviour that could affect the QOS of other established connections. Several strategies have been proposed:

- *Discarding Cells*: The simplest action is to discard those cells that violate the traffic parameters. In this case, the main objective is achieved: a source will never be able to transfer more cells across the network than the pre-negotiated values.

- *Selective Discarding*: Selective discarding [11] assumes that cells are tagged in a network. Rather than discarding any cell that violates the negotiated rate, violation tagged cells are discarded by a switch when they encounter a threshold in a switch queue. The mechanism thereby accords greater priority to the transport of untagged cells.

- *Dropping Connections*: This is the most drastic action in which the switch simply drops the connection as soon as it detects that a connection is in violation of its contract.

- *Fast Resource Management*: When a connection needs to operate at a higher rate, it negotiates its intention with the network dynamically. The renegotiation allows for more resources being reserved in the network to support the connection's new rate of operation.

- *Burst Scale Multiplexing*: This is a class of strategies [37] that recognises that bursty sources such as LANS operate in an on/off mode. They are transmitting either at the full speed of the link or not at all. Higher layer protocol packets such as those belonging to TCP/IP need to be fragmented into many small sized cells before they can be transmitted onto the B-ISDN network. The loss of a single cell would therefore cause the entire packet to be discarded at the receiver. Switches recognise this condition and discard the burst of cells corresponding to a higher level packet whenever it senses the loss of a single cell. The consumption of additional resources by the cells is thereby prevented.

### 3.2.2 Medium-Term Controls

Medium-term controls are generally feedback-based *reactive* controls that send information about the network state back to senders. Senders then use the feedback information to adjust the rate of traffic release. These controls operate on a time scale on the order of end-to-end propagation delays in the network.

There are several feedback-based schemes that have been proposed or are in operation:

- The *DECbit* scheme is a *window-based congestion avoidance* approach proposed by Ramakrishnan and Jain [33]. In the DECbit scheme, a single bit per packet is used by the network to provide congestion information back to the traffic sources. The switches in the network detect congestion and set a congestion indication bit for traffic flowing in the forward direction. The congestion indication is transmitted back to the source in the form of a transport layer acknowledgment sent by the destination. The sources adaptively adjust their send window sizes to avoid congesting the network. This scheme is used in the DECNET architecture.

- The *Slow Start* scheme [16], another window-based approach and adopted in the TCP protocol, is based on the principle that packet loss is a good indicator of network congestion. New senders begin a connection at a minimal flow control window size, and increase the amount they send whenever an acknowledgment is received. When a packet loss is detected at timeout by a sender, the window size at the sender is decreased to a minimal size, and is increased gradually with each acknowledgment.

- *Loss-Load Curves* [38, 39] are a *rate-based* mechanism whereby switches in the network detect congestion and send rate-based feedback information in the form of loss-load curves to the sources. For each source, the loss-load curve expresses a cell loss probability as a function of the source's offered load. Sources can use this information to adjust their rate to achieve the desired level of packet loss. Loss-load curves are described in greater detail in Section 3.3.2.

  Other rate-based schemes include the Packet Pair scheme [20], and the Hop-by-Hop congestion control scheme [29].

### 3.2.3 Long-Term Controls

Long-term controls operate on a time scale that is approximately equal to the duration of a connection across a network. This approach is particularly suitable for those connections that have stringent QOS requirements on tolerable cell loss and delay. Upon a user request, a logical connection will be established hop-by-hop through the network with the state of the network being maintained at individual switches. *Admission Control* is a reservation-based procedure where the network decides whether to accept or reject a new connection. The network has to ensure that the admission of a new connection does not adversely affect other active connections, while simultaneously making sure that the network resources are utilised efficiently. In making this decision, the network uses the *traffic descriptor* of the connection that characterises the nature of the traffic released by the source. Several traffic descriptor parameters have been proposed, such as the mean cell rate, the peak cell rate, the maximum burst length and the arrival process distribution of a connection.

*Centralised Network Management* controls are needed to fine tune the operation of the network during its operation. These controls may include adjusting the parameters of the distributed controls or enabling/disabling network actions depending on the state of the network. An example is reconfiguring the network to route packets away from a congested node.

## 3.3 Components of the Virtual Loss-Load Model

The virtual loss-load model proposed in this thesis, and described in Chapter 4, combines the leaky bucket traffic shaper with the loss-load model together with some modifications to provide a hybrid congestion control strategy. This section examines the existing leaky bucket and loss-load models in greater detail.

### 3.3.1 Leaky Bucket Model

As described previously, the leaky bucket model is a traffic shaping approach implemented at the network edges. Senders release traffic cells at a controlled rate into the network, thereby smoothing out bursts. This section examines the basic leaky bucket traffic shaper in more detail, along with a variant, the virtual leaky bucket traffic shaper.

#### Basic Leaky Bucket Traffic Shaper

Figure 3.2 depicts the operation of a buffered basic leaky bucket traffic shaper. Arriving cells enter a queue in the traffic shaper. If the queue is full, the arriving cell is dropped. For a cell at the head of the queue to enter the network, the cell must first obtain a token from a token pool. Tokens are generated at a fixed deterministic rate and enter this pool. If the token pool is full, newly generated tokens are discarded.

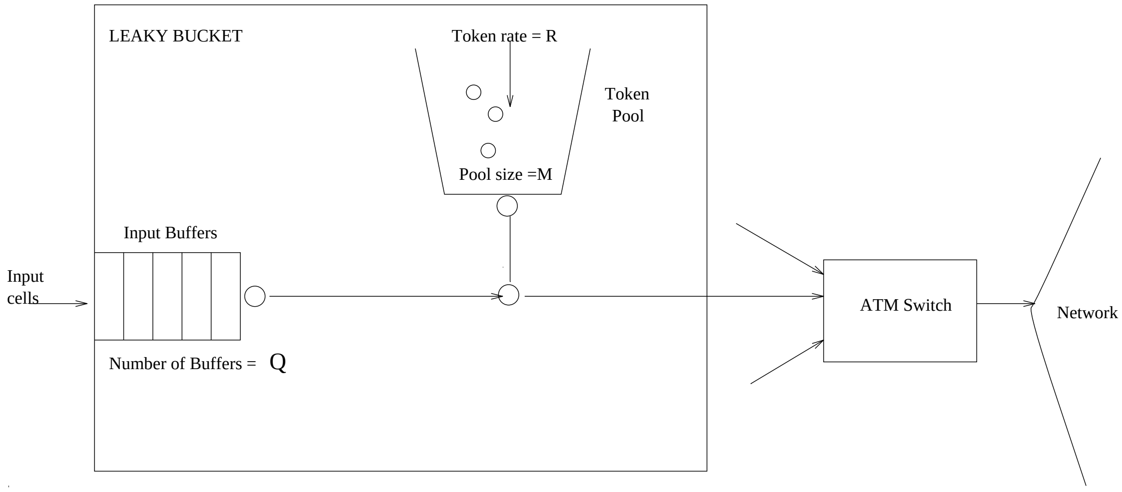



Figure 3.2: A Buffered Leaky Bucket Traffic Shaper

Several parameters govern the operation of the leaky bucket. The size of the token pool $M$ limits the maximum number of back-to-back cells that can enter the network. The token rate $R$ determines the mean rate at which cells enter the network. Typically, this rate is set to the rate allocated to the connection at the time of call admission. The size of the queue $Q$ represents a tradeoff between cell delay and cell loss. Increasing the queue size decreases the probability of cells being dropped at the shaper. However, the resulting longer queues imply that cells must wait for longer periods before being transmitted. Earlier work by Liu [24] has established the following results for the basic leaky bucket traffic shaper:

- The shaper has been shown to result in very good network performance. Traffic flows entering the network after passing through the shaper are smooth, and cause little or no delay, even when the network load is high. The leaky bucket traffic shaper also helps in reducing the buffer requirements at the switches, and provides protection to the network from misbehaving senders.

- This good network performance, however, comes at the expense of extra queuing and delays at the shaper. There is a definite tradeoff to be made between user level performance and network level performance, in terms of cell loss and delay.

- The leaky bucket traffic shaper is conservative in its operation, and may not exploit available capacity when network conditions change dynamically.

**Virtual Leaky Bucket Model**

The basic leaky bucket traffic shaper possesses the disadvantage of operating at a fixed token generation rate, thereby preventing a connection from exploiting possibly increased available capacity in the network. To overcome this problem, a virtual leaky bucket traffic shaper has been proposed [14]. In a virtual leaky bucket, cells arriving at an empty token pool are tagged red while other cells are marked green before being dispatched. Red cells are considered violators of available bandwidth as their transmission implies operation at a rate higher than that allocated. However, because of the statistical nature of arriving traffic, the network may have sufficient capacity to carry the red cells, provided other connections are not adversely affected. If congestion is detected somewhere in the path of the connection, the red cells are discarded so as not to affect the transmission of the green cells. Cell tagging allows senders to operate at higher rates, but provides no guarantees for the delivery of red cells in the network.

In summary, the leaky bucket model offers the following advantages:

- *Simplicity* : The leaky bucket model shapes traffic effectively and is simple to implement.

- *Controllability*: The parameters determining the operation of the leaky bucket are easy to control and can be set based on the traffic descriptor of the connection.

The model, however, has certain drawbacks:

- *Conservative Operation*: As the leaky bucket traffic shaper parameters are set statically, it may not be possible to exploit available network bandwidth when network conditions vary dynamically.

- *No Sender Control*: Although the virtual leaky bucket traffic shaper does allow a sender to operate at a higher rate, no information exists at the sender about the expected loss rate of red cells released into the network. Furthermore, the colouring of the cells is based entirely on the traffic arrival characteristics of the workload. All violating cells generated by the workload are coloured red, providing the sender with little control over the colouring process.

### 3.3.2 The Loss-Load Model

The loss-load model is a reactive congestion control mechanism that offers the advantage of allocating bandwidth depending on changing network conditions. Switches monitor network load and provide explicit loss-load feedback information to senders. A loss-load curve exists for each switch's outgoing link. Each loss-load curve is based on the total load on an outgoing link and the capacity of that link, and, for each sender, expresses the loss probability as a function of the sender's load. Senders select the rate of operation depending on the packet loss they desire.

Switches monitor each sender's rate of operation. If the load on an outgoing link is less than its capacity, no packets are dropped at the switch. If the capacity of a link is exceeded, a switch drops the excess cells by associating a packet loss probability

$p_i$ with each sender using the link. By dropping excess packets, switches manage to restrict the load on a link to operate at or below its capacity.

Figure 3.3 depicts an example of a loss-load curve. The packet loss probability $p$ ($0 \leq p \leq 1$) is a non-decreasing function of the sender's transmission rate $f$ ($0 \leq f \leq 1$), where $f$ is expressed as a fraction of the switch's output link capacity. The packet loss probability approaches 1.0 for high values of $f$ to protect the network from misbehaving (greedy) senders.

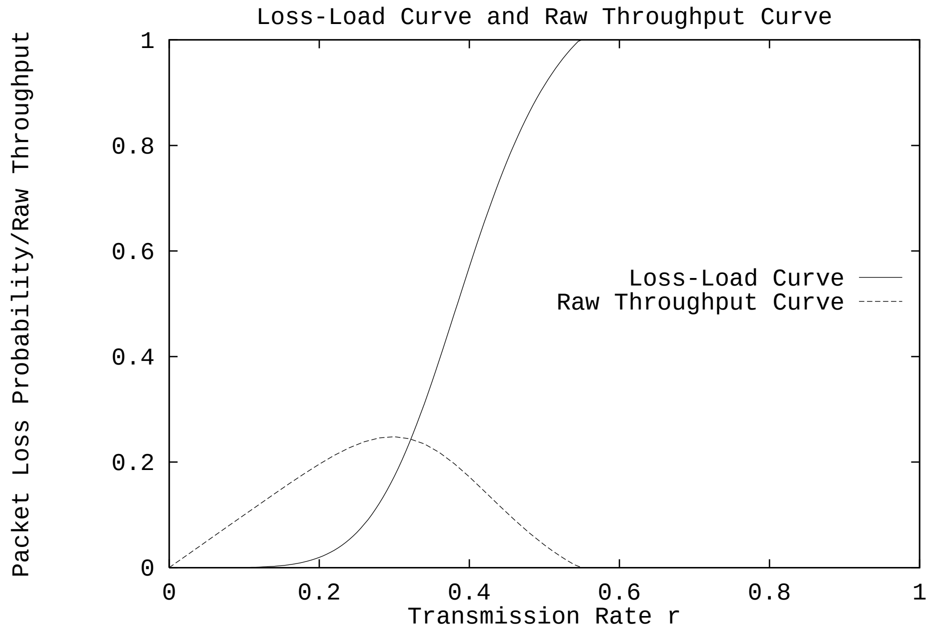



Figure 3.3: The Loss-Load Curve for a Sender ($k = 5$) [38]

Associated with each loss-load curve is a raw throughput curve that measures the throughput attainable by a sender for a given rate of operation. The throughput curve is directly computable from the loss-load curve using the relation $t = (1 - p)f$.

The entire loss-load feedback is represented using only two floating point numbers $a$ and $b$, that are computed at each switch and fed back to senders. Senders compute the loss-load curve based on the values of $a$, $b$ and a constant integer parameter $k$, which is known globally by all senders and switches in the network.

The loss-load curve for a sender is always expressed in terms of the aggregate load offered by *other* senders in the network. If $M + 1$ senders share a single outgoing link at a switch, the switch computes the following for the $M+1$st sender:

- $f_i$, the rate of operation of the $i$th sender for all $i$ varying from 1 to $M$,
- $f_{total} = \sum_{i=1}^{M} f_i$ the aggregate load on the link of capacity $C$,
- $a = f_{total} - C$, the excess load on the link,
- $b = \sum_{i=1}^{M} f_i^{k+1}$, the distribution of the load among senders.

The computed values of $a$ and $b$ are sent back to the $M+1$st sender by the switch. The $M+1$st sender uses the values of $a$ and $b$ to compute the packet loss probability at a rate $f$ using the formula $p(f) = \frac{f^{k+1}+af^k}{f^{k+1}+b}$. $k$ is an important parameter in the loss-load formula as it determines how severely senders with high transmission rates are punished relative to slower senders. The larger the value of $k$, the greater is the packet loss probability for a sender at a given rate.

Senders can select their rate of operation based on several criteria as follows:

- **Throughput Optimization**:

  Senders may base their rate of operation on the raw throughput curve. For low rates of operation, the throughput remains low as the sender does not exploit available network bandwidth. At very high rates of operation, a sender's throughput decreases as the network discards most of the packets belonging to the sender. Somewhere between these two extreme rates of operation is a rate that allows senders to maximise raw throughput in the network. Raw throughput is maximised at a rate $f_t$, obtained by solving for $f$ the equation $\frac{dt}{df} = 0$, a polynomial of degree $k+1$.

  The loss-load model offers some important results when senders maximise throughput [38]:

  - *Bounded Packet Loss*: When a sender maximises raw throughput, the packet loss probability for a sender at $f_t$ is $\frac{1}{k+1}$. This result is important as it provides a bounded and predictable cell loss probability for a sender at $f_t$.

- *Convergence*: If $M$ identical senders sharing a single bottleneck link of capacity $C$ all maximise raw throughput, the network converges to a stable equilibrium operating point, with each sender operating at its individual $f_t$. The total offered load in the network is $\frac{k+1}{k}C$. The property of convergence is important as it shows that the loss-load model reaches stability eventually. The equilibrium reached is independent of a sender's initial rate and feedback delay. Smaller values of $k$ increase the rate of convergence as compared to larger values. However, since the total offered load at equilibrium is $\frac{k+1}{k}C$, larger values of $k$ are useful in decreasing overload.

- **Response Time Optimization**:

  Instead of operating at a maximum throughput rate, senders may wish to operate at a rate that minimises the response time or transfer delay for a connection. The expected response time for a transfer of $N$ packets for a sender with a round trip time $R$ is given by [39]:

  $Response(f) = \frac{N}{(1-p)f} + (1 - \frac{logN}{logp})R$,
  where $p$ is the packet loss probability at rate $f$.

  To minimise response time, a sender solves for $f$ in the equation $\frac{d(Response(f))}{df} = 0$, a polynomial of degree $k$. Minimum response time is obtained at a rate $f_r$. For $R \rightarrow 0$, or $N \rightarrow \infty$, $f_r$ approaches $f_t$.

- **Controlling Packet Loss**:

  A sender concerned with obtaining a low loss rate may decide to operate conservatively on the flatter portion of the loss-load curve. By operating in this region of the curve, the packet loss probability for a sender can be made relatively insensitive to transient changes of network load.

In summary, the loss-load model offers the following advantages:

- *Concise Representation*: The loss-load model provides a concise mathematical representation of aggregate load in the network, as opposed to schemes which just indicate congestion in the network without quantification.

- *Sender Control*: Since the loss-load model captures all the information about load conditions in the network, senders have a precise knowledge of the expected loss-rate at different rates of operation. Senders thus have control in determining the rate at which they need to operate.

The model has the following drawbacks:

- *Aggressive Nature*: When all senders maximise raw throughput, the overall load in the network is higher than the capacity of the network, depending on the value of $k$. The packet loss probability is, in many cases, far too high (e.g., 9.1% packet loss probability for $k = 10$) to support the QOS requirements of expected B-ISDN applications.

- *Implementation Complexity*: Although large values of $k$ decrease the overload at network equilibrium, they also increase the complexity of computation, both at the senders and at the switches. When senders optimise throughput (response), they need to solve an equation of degree $k+1$ ($k$). Each time a switch computes a loss-load curve, or when it needs to drop a cell belonging to a sender, it needs to perform a multiplication computation which is proportional to $k$. Further, large values of $k$ also result in slower convergence to network equilibrium. Large values of $k$, however, are suitable for reducing network overload at equilibrium.

## 3.4 Summary

The presence of diverse traffic types, broadband speeds, and large delay-bandwidth products all make the design of efficient congestion control mechanisms a challenging task. Controlling congestion in a network involves a combination of short-term, medium-term and long-term approaches. The leaky bucket shaper is employed at the network edges and allows senders to smooth out traffic bursts. The loss-load model is a reactive approach that allows senders to choose their rate of operation based on the packet loss they desire. The virtual loss-load model unifies both the leaky bucket and the loss-load approaches, to offer a (possibly) improved congestion control strategy for the short-term and medium-term time scales.

# Chapter 4

# Virtual Loss-Load Model

The objective of this thesis is to explore a 'virtual loss-load' congestion control strategy that combines the preventive leaky bucket scheme with the reactive loss-load approach. The virtual loss-load model is intended to exploit the advantages and mitigate the disadvantages that each of the two schemes possesses in isolation. The virtual loss-load model is based on the premise that an integrated approach acting both within the network and at the source is likely to result in an improved congestion control strategy.

This chapter presents a detailed description of the components of the virtual loss-load model, a description of the methodology and metrics used to evaluate the virtual loss-load model, and a simple network scenario to illustrate the operation of the virtual loss-load model.

## 4.1 Virtual Loss-Load Model Description

Senders in the virtual loss-load model use the general model of the basic leaky bucket traffic shaper at the network edges to control cell release as shown in Figure 4.1.

The leaky bucket shaper used in the virtual loss-load model contains two major changes. First, the token generation rate is not maintained at a fixed rate as is the case in the basic and virtual leaky bucket traffic shapers. Senders can modify the token generation rate (and hence the cell release rate) based on the loss-load feedback that

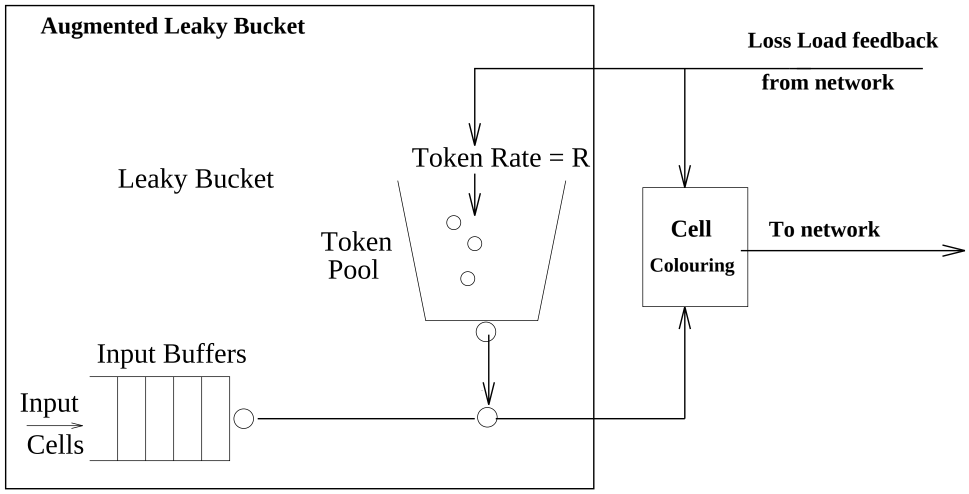



Figure 4.1: Augmented Leaky Bucket Shaper in the Virtual Loss Load Model

they receive from the network. Senders can use any criteria in modifying the token rate such as optimising throughput, minimising response time, or controlling packet loss. Senders may also increase the token generation rate in response to an increased rate of cell arrivals to the shaper. Doing so allows senders not to discard the excess cells at the buffers of the shaper. Second, the cell colouring strategy employed in the virtual loss-load model differs from that used in the virtual leaky bucket shaper. Senders colour a fraction of the cells departing from the leaky bucket. The fraction of cells coloured is chosen so as to match the cell loss probability associated with the current rate of cell release, and is calculated using the loss-load feedback. Senders are free to choose any departing cell for colouring, as long as the overall fraction of cells coloured matches the expected cell loss probability.

The switches in the virtual loss-load model also contain several modifications from those in the original loss-load model. First, the switches associate a virtual capacity $C_v$ with each outgoing link connected to the switch. If the actual capacity of a link is $C$, then $C_v = vC$ where $v$ $(0 \leq v \leq 1)$ is a parameter called the virtual factor. All loss-load feedback computations made by a switch are based on $C_v$ rather than $C$, so that senders perceive operation over links of virtual capacity. Switches, however, discard senders' cells based on the actual capacity $C$ of a link, similar to the loss-load model. The scheme, however, requires sender co-operation in order to work properly.

Non-cooperating greedy senders, for example, that are aware of the network 'lying' about its capacity may choose to operate at a higher rate to exploit available capacity. Second, switches give higher priority to the transmission of green cells compared to red cells. Whenever a discard decision is made for a sender's cell at a switch, the switch discards the arriving cell if it is red. If the arriving cell is green, the switch scans its output buffers looking to drop a red cell belonging to that sender. If no red cell is found, the arriving green cell is discarded. Finally, switches in the virtual loss-load model can use the concept of *delayed resolution* [38] in discarding cells. Rather than discarding cells when they arrive, cells are queued in the output buffer of the destination link. The queueing process continues until an arriving cell finds the buffer to be full, in which case the cell discard decision is made in a manner similar to that described earlier, based on cell priority. Delayed resolution allows superior discard decisions to be made at a switch, at the cost of increased overhead. The switch buffer size plays a key role in the delayed resolution scheme. Larger values of the buffer size cause the discarding decision to be deferred, resulting in lower cell loss, although at the expense of increased cell delay.

## 4.2 Advantages of the Virtual Loss-Load Model

The virtual loss-load model is not intended to replace reservation-based schemes that offer a guaranteed quality of service to different traffic sources. Like other reactive schemes, the virtual loss-load model is intended to operate under dynamic network conditions, and there will be periods during which a connection's QOS requirements will not be met. Rather, traffic operating in a virtual loss-load model environment is assumed to have relatively relaxed QOS requirements, with the expectation that the network will make the best-effort in satisfying its requirements. Such traffic, also referred to as *reservationless* or *datagram* traffic, has relatively low priority, or is delay-insensitive or loss-insensitive. Traffic belonging to this category can thus tolerate greater amounts of packet loss, or can recover lost packets by retransmissions.

Even though it cannot provide performance guarantees, by combining the leaky bucket shaper with the original loss-load model, the virtual loss-load model can provide the following advantages:

- *Simplicity and Controllability*: The leaky bucket shaper offers the advantage of simplicity and controllability. These advantages are preserved in the virtual loss-load model. The virtual factor $v$ is an additional controlling parameter in the virtual loss-load model. By choosing appropriate values of $v$, the congestion control strategy can be tuned to satisfy the performance requirements of different types of traffic.

- *Increased Network Throughput*: Since the token generation rate of the leaky bucket shaper can be modified using loss-load feedback, the conservative nature of the basic leaky bucket shaper is overcome in the virtual loss-load model. The virtual loss-load model can therefore exploit available network capacity depending on dynamic network conditions, thereby increasing network throughput.

- *Reduced Network Overload*: There are two parameters that control network load at equilibrium in the virtual loss-load model. By extending a result of the loss-load model, the expected load in the virtual loss-load model when all senders maximise throughput is $\frac{k+1}{k}C_v = \frac{k+1}{k}vC$. The choice of $k$ in the original loss-load model determines how aggressive the model is (i.e., by how much the network capacity is exceeded). The parameter $v$ is an additional factor determining network overload in the virtual loss-load model. For a given value of $k$, $v$ can be varied to reduce or nullify network overload. Small values of $v$ can eliminate network overload completely, at the cost of possibly decreased network throughput, while accommodating dynamic network conditions. The virtual loss-load model thus allows an additional parameter that can help reduce network overload, while preserving the convergence property of the original loss-load model.

- *Reduced Sender Loss*: In the basic leaky bucket traffic shaper, cells arriving at the shaper are dropped when the buffer at the shaper becomes full. Dropping cells implies that the average cell arrival rate at the shaper exceeds the token generation rate of the shaper. In the virtual loss-load model, the token generation rate can be increased to match the cell arrival rate to take advantage of transient capacity availability in the network, while continuing the traffic shaping process. Additionally, since switches compute loss-load feedback based on the virtual capacity of a link while discarding cells based on the actual capacity, the actual loss rate of a connection will be less than that expected by the sender. In this case, the loss-load feedback provides an upper bound on the expected loss probability to a sender. Further, the use of the delayed resolution scheme at the switches can help reduce the cell loss even further.

- *Sender Control*: Senders are provided with a greater degree of control in the virtual loss-load model. First, since the colouring process is based on loss-load feedback, senders are provided with a very good guarantee that the cell priority determined by the colouring will be honoured closely by the network. In the virtual leaky bucket traffic shaper, senders can only rely on "best-effort" delivery by the network. Second, senders can operate at a higher rate, similar to the virtual leaky bucket shaper but with the additional advantage of deciding which cells they want to colour, as long as the fraction of cells coloured matches the expected cell loss probability at the current rate of operation. This allows senders to decide the priority of cells they release into the network while simultaneously allowing a higher rate of operation, a combination not possible in the virtual leaky bucket traffic shaper.

## 4.3 Experimental Background

Experiments conducted to evaluate the performance of the virtual loss-load model were based on a benchmark suite developed by Kanakia, Keshav and Mishra [19]. The scenarios forming that benchmark suite are described as being representative of typical high delay-bandwidth product networks, and attempt to capture a variety of network conditions.

Kanakia *et al.* propose this benchmark suite as a standardised test suite for the evaluation of several reactive congestion control strategies. The work conducted in this thesis uses the same benchmark scenarios, the same workload models, and the same assumptions as in the proposed benchmark suite. This approach facilitates comparison of the virtual loss-load congestion control strategy with the other congestion control strategies mentioned above.

An important factor determining the performance of a network is the workload at the senders. In contrast to general stochastic models (such as Poisson processes) used in most literature, the workload at each sender is assumed to be a finite file transfer application in [19]. According to that study, the file transfer application is a very general model of a network application: applications as diverse as network news, database access and shared whiteboards all employ the file transfer paradigm, which is the bulk transfer of data at the fastest rate possible. The study argues that a realistic evaluation of a congestion control scheme can be made only on the basis of finite file transfers, unlike many existing studies that assume infinite data sources, so as to study performance at stability. Also, according to the study, as networks become faster, file transfer durations will fall, so that it is the transient network conditions that are most relevant in studying the performance of a congestion control scheme.

## 4.4 Performance Metrics

In this thesis, the following performance metrics, used by Kanakia *et al.* are used to measure the performance of the virtual loss-load model. In the simulation studies conducted in this thesis, the performance metrics are measured for a specific connection in the network, called the *control connection.* The metrics are:

- *File Transfer Time* is the performance metric used in this study to measure network throughput. File transfer time is defined as the interval between the instant at which the first byte of data is available at the sender, and the instant at which the sender receives the acknowledgment that the destination received the last byte of the file. This metric is a sum of the time taken to transmit all the bytes of the file, plus the time taken to retransmit lost cells.

The file transfer time is the primary metric used by Kanakia *et al.*. It directly measures the effective throughput achieved by a congestion control scheme: the lower the file transfer time, the higher the effective network throughput, and vice versa. According to [19], the file transfer time allows for the simultaneous representation of several other metrics, rendering the use of those metrics unnecessary. Some of these other metrics are:

– *Bottleneck Utilisation*: A high value of the file transfer time would imply that the link was underutilised even when there was data available to be sent, indicating low bottleneck utilisation.

– *Fairness*: By comparing the file transfer times of two identical senders that need to send the same amount of data over different path lengths, a measure of the fairness of a congestion control scheme can be determined.

– *Cell Loss Probability*: If too many cells are lost, the lost packets would need to be retransmitted, causing an increase in the file transfer time.

The file transfer time is thus used as the primary performance metric in this thesis. Several additional metrics specific to the virtual loss-load model are also used, as described later.

- *Mean Cell Delay* measures the mean queueing delay encountered by a cell in the network from the time the cell is released from the leaky bucket, to the time of its arrival at the receiver. This metric is measured only for cells that are delivered across the network without retransmissions, and excludes the transmission times, propagation delays and processing delays for a cell as it travels across the network. In [19], mean cell delay is used as a secondary performance metric, as it serves to measure the response time seen by interactive users. Low values of the mean cell delay are desirable, indicating low queuing delays in the network.

- *Cell Delay Variation (C.D.V.)* is the performance metric used to measure cell delay jitter. Low values of this performance metric are desirable so that all cells arrive with a delay close to the mean cell delay. In [19], and in this thesis, the standard deviation of the mean cell delay and the coefficient of variation of delay (C.O.V. of delay) is used to measure the C.D.V.

The following additional performance metrics, specific to the virtual loss-load model, are used in this thesis:

- *Cell Loss Probability (C.L.P.)* measures the fraction of cells lost by a connection, and is the performance metric used to measure whether senders encounter reduced cell loss. This metric helps in determining the presence of network overload, and low values of this performance metric are desirable. Although this metric is not used in [19], it is used in this thesis for comparing the performance of the virtual loss-load model with the original loss-load model. The metric is useful in determining the efficiency of schemes having comparable file transfer times, but different C.L.P.'s. For example, the scheme having the lower C.L.P. may operate closer to network capacity, causing little overload in the network. On the other hand, the scheme with the higher C.L.P. may achieve a lower file transfer time by operating above network capacity, thereby maximising throughput, but with the disadvantage of causing increased congestion in the network. If the improvement in the file transfer time is marginal, then the scheme with the lower C.L.P. is the preferred scheme.

- *Green Cell Loss Probability* measures the fraction of green cells lost in the network, and helps measure the degree of control senders have in deciding cell loss priority. Since senders expect that all green cells will reach the receiver, in an ideal operating environment, this performance metric should have a value of 0. It may be noted that the green cell loss probability is more relevant to a sender than is the C.L.P. The green cell loss probability measures the cell loss probability that a sender did not expect, while the C.L.P. may reflect cell loss that the sender was willing to tolerate. The fraction of red cells that a sender loses in a network (the red cell loss probability) is not used in this thesis, as this metric is partially reflected in the C.L.P. metric itself.

## 4.5 Virtual Loss-Load Simulator

A discrete time simulator implementing the virtual loss-load model has been developed. The simulator has been designed primarily to test network scenarios as described in [19]. The simulator contains the following components:

- **Links**: Links of different capacities and propagation delays can be declared.
- **Connections**: Connections are specified by sender-receiver pairs and by the links that connect them. The workload at each sender is a finite file transfer application. The startup time of each connection and the number of bytes to be transferred are specified. Multiple connections can exist on a sender, allowing for multiplexing on a single link. Each connection in this case operates its own leaky bucket. Cells released from the leaky bucket are buffered before being released into the network. The buffer is necessary because all leaky bucket shapers that are being multiplexed may try to dispatch a cell in a given time slot simultaneously. Since only one cell can be released into the network in a given time slot, however, the additional cells must be buffered for subsequent cell release. When the buffer becomes full, operation of all the multiplexed leaky bucket shapers is shut off. The operation of the multiplexed shapers is resumed once buffer space becomes available.

  Senders undergo the following phases of operation:

  - *Connection Establishment*: Senders inform the network of the need to establish a connection by sending an ESTABLISH message. A virtual connection is established end-to-end, and senders change from the IDLE state to the ACTIVE state.
  - *Data Transfer*: In the ACTIVE state, senders transmit cells, all of which are assumed to have a fixed size of 500 bytes, similar to senders in [19]. During the course of the data transfer, senders may receive loss-load feedback (LLOAD messages) from the network. Senders use the response time

model [39] in selecting the rate of operation. If the newly selected rate differs from the earlier rate of operation, senders transmit a RATE message to the network. RATE messages are never lost in the network. Senders colour cells based on the cell loss probability associated with the current rate of operation, as determined by the most recent loss-load (LLOAD) feedback message.

– *Connection Termination*: When a sender has completed a data transfer, it sends a CLOSE request to the receiver, and enters the CLOSING state. The receiver acknowledges the number of cells it has received to the sender by sending an ACK message. If all cells are acknowledged, the sender informs the network of its decision to terminate the connection by sending a TERMINATE message, and then enters the CLOSED state. If not, the sender re-enters the ACTIVE state, from which it retransmits the unacknowledged cells. Retransmissions occur until all cells are acknowledged.

Figure 4.2 summarises the state transitions for a sender in the simulator.

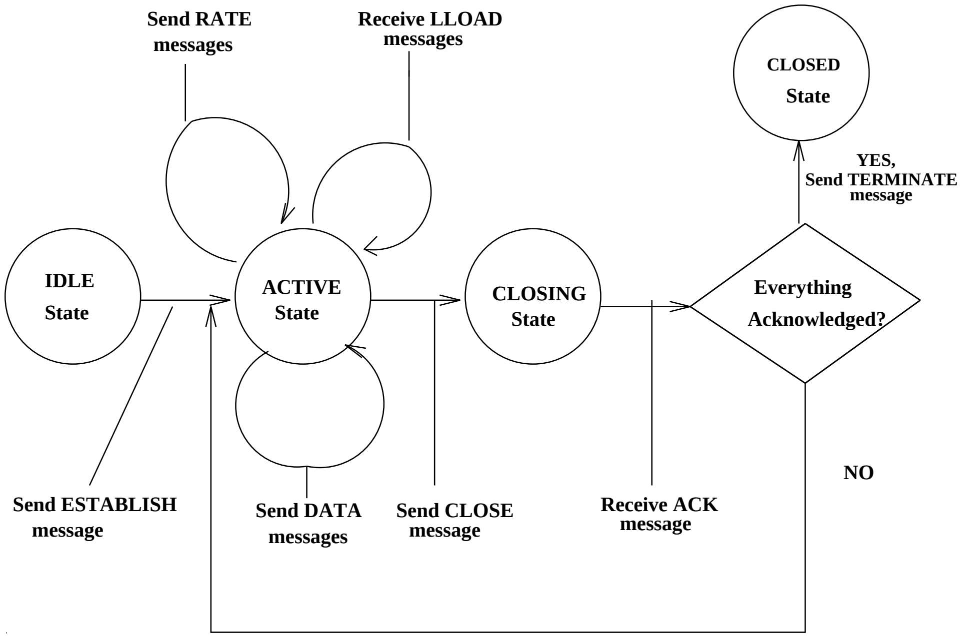



Figure 4.2: State Transition Diagram for a Sender in the Virtual Loss-Load Simulator

- **Switches**: Switches implementing the virtual loss-load model can be specified, together with the virtual capacities of all output links connected to the switch. All switches are output buffered, and each switch keeps track of all active connections by means of a table. Loss-load curves are computed whenever the switch receives an ESTABLISH, RATE or TERMINATE message from a sender. Whenever any sender changes its rate, a new LLOAD message is sent to all active senders by a switch. Switches propagate the LLOAD message to the receiver by means of the loss-load parameters $a$ and $b$. Switches that receive a LLOAD message from a previous switch, also compute the loss-load parameters $a$ and $b$ for the sender specified in the LLOAD message. If the computed value of $a$ is greater than the value of $a$ in the LLOAD message (indicating that the bottleneck link is connected to this switch, and not the previous switches), the newly computed values of $a$ and $b$ replace those in the LLOAD message. Otherwise, the received LLOAD message at a switch is transmitted unchanged. This ensures that the LLOAD message remains updated with the values associated with the bottleneck link. Receivers propagate the LLOAD message back to the sender on receipt. Loss-load feedback is thus based on the round-trip time of a connection and is used in the experiments so as not to bias the results in favour of cases where switches are closer to senders. When the discard strategy at the switch is based on the original loss-load algorithm, whenever a cell arrives from a sender, the switch determines if the cell needs to be dropped based on the computed value of loss for that connection. That is, if the loss probability associated with sender $i$ is $p_i$, then each arriving cell from sender $i$ is discarded at random, with probability $p_i$. Switches base the computation of $p_i$ on the actual capacity of the output link and always give priority to the transmission of green cells.

## 4.6 Preliminary Experiments

In order to evaluate the effectiveness of the virtual loss-load model, a preliminary experiment was conducted to study a simple two-sender case as shown in Figure 4.3.

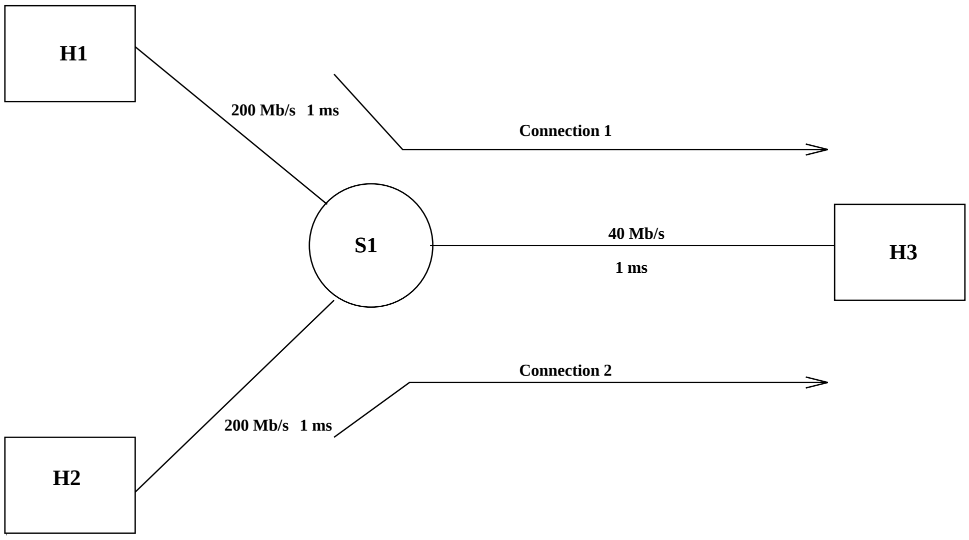



Figure 4.3: Two Sender Case

In this scenario, two hosts H1 and H2 share a bottleneck link of capacity 40 Mbps and transmit to a host H3. Each sender transmits over a link of capacity 200 Mbps. Packets transmitted are of a fixed size of 500 bytes. Both senders transmit 2 Mbytes of data and start at time 0. The leaky bucket traffic shaper in the virtual loss-load model operates with a buffer size of 10, and a token pool size of 5. Switches use the delayed resolution scheme with a buffer size of 500 packets. The value of $C$ is normalised to a capacity value of 1, and all links have a propagation delay of 1 ms. All performance metrics are measured for the control connection 1. Both senders transmit at a rate to maximise throughput, and colour the cells randomly based on the loss rate obtained from loss-load feedback.

The objective behind the experiment was to study the operation of the virtual loss-load model under a simple network scenario. Two main advantages are gained from this study. First, it provides a validation mechanism for the virtual loss-load model by allowing a comparison between the theoretical predicted performance of the virtual loss-load model with those obtained through simulation results. Second, it provides a reference point for comparison with the more complicated benchmark scenarios proposed by Kanakia *et al.*

The simulation experiments use two factors $k$ and $v$. As both senders use the raw throughput model in maximising throughput, the expected load at equilibrium is $\frac{k+1}{k}C_v = \frac{k+1}{k}vC$. When $v = \frac{k}{k+1}$, the load at equilibrium should equal 1, so that senders maximize network utilisation without causing network overload. At this value of $v$, therefore, senders should expect the best possible performance, and all the performance metrics should be optimised at this value of $v$. The motivation behind the experiment was to determine whether this indeed was the case.

For all the conducted two factor experiments, the parameter $v$ was varied from $0.5 \ldots 1.0$ in steps of 0.1. For each value of $v$, values of $k$ at 2, 5, 10 and 20 were used, similar to [38]. For each value of $k$ and $v$, the performance metrics described in Section 4.4 were used to indicate how well the virtual loss-load model performed. For the sake of simplicity, only two performance metrics are presented for this simple scenario: the file transfer time and the cell loss probability.

The following describes the experimental results obtained:

**File Transfer Time**

The file transfer time provides a measure of network throughput. The theoretical optimum transfer time for each connection is 800 ms, assuming both connections share the bottleneck link equally. The purpose of this experiment was to determine whether the file transfer time was minimised at $v = \frac{k}{k+1}$.

Figure 4.4 shows how the file transfer time changes with values of $v$ for connection 1 for different values of $k$.

For small values of $v$, the file transfer time is large. This result is expected since at small values of $v$, the entire network capacity is not exploited. As the value of $v$ increases, the file transfer time drops as the virtual capacity of the bottleneck link approaches the actual capacity. From the graph, it may be seen that the optimum file transfer time can be achieved for any value of $k$, but the value of $v$ at which this occurs depends on $k$, namely at $v = \frac{k}{k+1}$. It can be seen that file transfer time is indeed minimised at a value of $v$ close to $\frac{k}{k+1}$. Values of $v > \frac{k}{k+1}$ result in increased

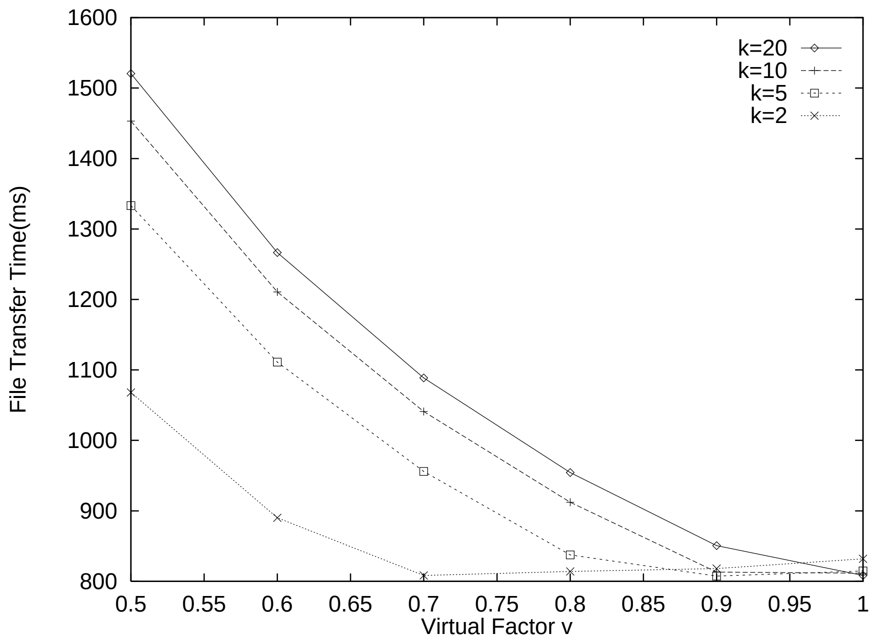



Figure 4.4: File Transfer Time for Connection 1

overload, more cell loss and retransmissions, and no improvement in the file transfer time.

**Cell Loss Probability**

The C.L.P. measures the overall cell loss for a connection, and also provides an indication of network overload. Ideally, the C.L.P. for a connection should be close to 0. Again, theoretically, at a value of $v = \frac{k}{k+1}$, there should be no overload since the load at equilibrium for this value of $v$ is 1. The C.L.P. is expected to increase beyond this value of $v$, and the purpose of the experiment was to determine whether this indeed was the case.

Figure 4.5 depicts the cell loss probability for connection 1 for different values of $k$ and $v$. The simulation results match those predicted by theory. As in the earlier cases, the C.L.P. is negligible for values of $v \leq \frac{k}{k+1}$, since network overload is negligible. Above this value, however. overload in the network results in increased cell loss. A value of $v = 1$ results in using the original loss-load model. The resulting loss at this point matches closely with that predicted theoretically (i.e, $\frac{1}{k+1}$), yielding confidence in the simulation results obtained.

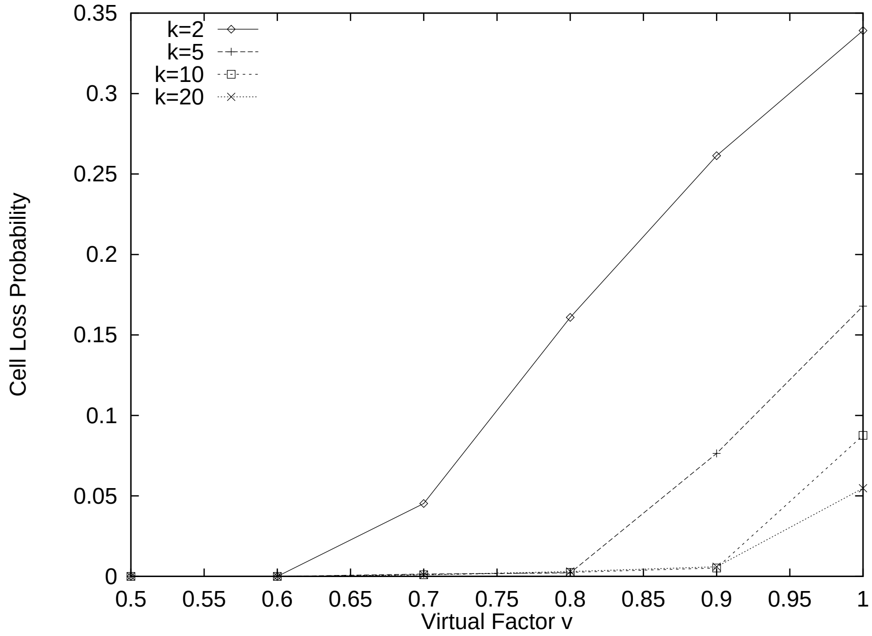



Figure 4.5: Cell Loss Probability for Connection 1

## 4.7 Summary

Experiments with the two sender case show the importance of both controlling parameters $k$ and $v$. The parameter $k$ determines the degree of aggressiveness of the virtual loss-load model by providing an indication of how much network overload is generated at equilibrium. The virtual factor $v$ sets the basis for the 'speed limit' of this overshoot. By choosing both parameters appropriately, it may be possible to obtain a controllable and predictable model that optimises the performance metrics. It is to be noted that these results hold only for this simple experiment with stable operating conditions and small feedback delays. The goal of this thesis is to examine the virtual loss-load model under network conditions reflective of emerging broadband networks. Chapter 5 does this by examining the virtual loss-load model under more dynamic network scenarios consisting of transient network conditions and large feedback delay paths, using the benchmark suite proposed by Kanakia *et al.* [19].

# Chapter 5

# Experimental Results

This chapter describes the various experiments that were performed using the virtual loss-load simulator. The experiments investigate how the controlling parameters in the virtual loss-load model affect performance. The experimental results also allow for the comparison of the virtual loss-load model with the original loss-load model, and with other feedback based congestion control strategies.

## 5.1 Experimental Objectives

As explained in Chapter 4, the experiments that were conducted to evaluate the performance of the virtual loss-load model were based on a benchmark suite developed by Kanakia *et al.* [19]. The scenarios in that suite attempt to capture dynamic network conditions and judge the performance of congestion control schemes under transient network conditions. The same benchmark suite is used for the virtual loss-load model as well, to facilitate comparison with other congestion control schemes.

Three factors influence the performance of the virtual loss-load model:

- The factor $k$ determines the aggressiveness of the virtual loss-load model. Setting $k$ determines by how much the virtual loss-load model will exceed the "speed limit" (i.e., the virtual capacity) at equilibrium.
- The virtual factor $v$ sets the "speed limit", which forms the basis for the over-

shoot generated by $k$. The virtual capacity is always less than or equal to the actual capacity (i.e., $0 \leq v \leq 1$).

- The absence or presence of the delayed resolution strategy. The absence of the delayed resolution strategy provides lower cell delay. The presence of the delayed resolution strategy causes increased cell delay while reducing the amount of cell loss. The size of the buffer at the switches plays a key role in the delayed resolution strategy. Larger values of the buffer size cause the discarding decision to be deferred, resulting in lower cell loss, although at the expense of increased cell delay.

The experiments conducted in this thesis were designed to answer the following questions:

- What role do the factors $k$, $v$, and the delayed resolution strategy play in controlling the performance of the virtual loss-load model?
- To what degree does the virtual loss-load model improve on the original loss-load model? It is to be noted that the original loss-load model is a special instance of the virtual loss-load model with $v = 1$, with $k$ being the primary factor determining performance.
- How does the performance of the original loss-load model and the virtual loss-load model compare with the performance of other congestion control strategies described in the literature?

## 5.2 Experimental Methodology

The benchmark suite used in this thesis consists of different network scenarios, and are based on those described by Kanakia *et al.* All the scenarios share certain common features, unless stated otherwise:

- All sources and switches are assumed to be infinitely fast, and bottlenecks always occur at the output buffer of the switches. In the scenarios, there are slow links that act as the bottleneck, and fast links that feed data into the switch and bottleneck links. Slow links operate at 40 Mbps while fast links have a bandwidth of 200 Mbps. The propagation delay varies from case to case.

- Senders are assumed to have finite amounts of data to send, with a constant packet size of 500 bytes. Since file transfer is the application assumed at the senders, the workload at the senders is deterministic: i.e., senders at the leaky bucket always have outstanding data to send as long as the file transfer session is active. The leaky bucket shapers at the senders have an input buffer size of 10 and a token pool size of 5.

- Switches have finite buffers that can hold up to 100 packets. In the experiments conducted using the delayed resolution scheme, however, two values were used for the switch buffer size: 440 (representing a delay-bandwidth product of packets) and 44 (representing $\frac{1}{10}$ the earlier value).

- In all the experiments, there is either one control connection (connection 1) or two control connections (connections 1 and 2). By using only one or two control connections, it is easy to study the performance metrics associated with that connection.

As mentioned in the earlier section, three factors govern the operation of the virtual loss-load model. Table 5.1 lists the factors and the levels used in the simulation experiments that were conducted:

Table 5.1: Factors and Levels used in the Experiments

| Factor | Levels | Number of Levels |
|---|---|---|
| k | 2, 5, 10, 20 | 4 |
| v | 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 | 6 |
| Delayed Resolution | absent, 440, 44 | 3 |

The four levels used for $k$ cover a broad spectrum of choice of $k$ in the virtual loss-load model. These values are chosen to match the values used in the original loss-load model [38]. The levels chosen for $v$ represent a range of operational strategies. A value of $v$ of 0.5, for example, would allocate only half the bandwidth to a connection that arrives at an unused link, and represents a conservative mode of operation of the virtual loss-load model. A value of $v = 1.0$, on the other extreme, captures the performance of the original loss-load model. Levels of $v$ greater than 1.0 are not used since the virtual loss-load model would always operate above capacity and this would not therefore be a viable congestion control strategy. There are three levels chosen for the delayed resolution strategy. In the first level, the delayed resolution strategy is not used at all, and the virtual loss-load model behaves essentially similar to the original loss-load model. The other two levels represent the values of the buffer sizes at the switch in the presence of the delayed resolution strategy. Rather than studying a range of values of the buffer size at the switches, only two values, 440 and 44, are used in studying the effect of this factor. A level of 440 is sufficient to accommodate a delay-bandwidth product of packets for the scenarios considered, while a level of 44 measures the performance of the delayed resolution strategy in the presence of much fewer buffers. As mentioned earlier, the value of the buffer size represents a tradeoff between the amount of cell delay and cell loss. By choosing two values, it is possible to study the performance of the delayed resolution strategy under two extremes. This also allows for the interpolation of performance for intermediate values of the buffer size.

Full factorial experiments were conducted for each benchmark scenario, yielding a total of 72 experiments for each. For each case, the performance metrics described in Chapter 4 were studied. The performance metrics are: the file transfer time, the mean cell delay and the cell delay variation (measured by the standard deviation and the C.O.V. of delay), the cell loss probability and the green cell loss probability.

## 5.3 Virtual Loss-Load Model Performance without Delayed Resolution

This section studies the performance of the virtual loss-load model for different values of $k$ and $v$, but without the delayed resolution strategy. The performance is studied for three different benchmark scenarios described in [19].

### 5.3.1 Scenario 1: Slow Changes in Cross Traffic

This scenario intends to measure the performance of the congestion control strategy in response to a gradual increase and decrease of traffic in the network. The configuration used for this experiment is shown in Figure 5.1.

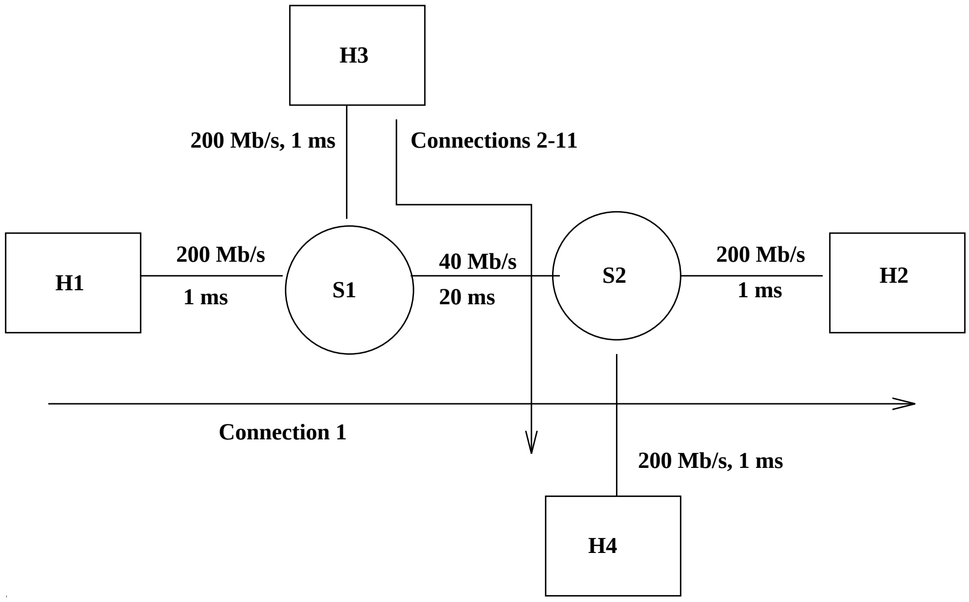



Figure 5.1: Network Configuration used for Scenarios 1 and 2

In this scenario, the cross traffic load seen by the control connection (connection 1) changes gradually. Connection 1 starts up at time 0 and has 2 Mbytes of data to send. Connection 2 commences after 100 ms. The remaining cross traffic connections, connections 3 through 11, begin at intervals of 44 ms (the round trip time for connection 1) following connection 2. Each of the ten cross traffic streams sends 500 Kbytes of data.

Before studying the performance of the virtual loss-load model for this scenario, it is instructive to study the dynamics of the virtual loss-load model, and to understand how the values of $k$ and $v$ affect performance. Although the dynamics of the virtual loss-load model are considered for this scenario alone, the behaviour of the model in other scenarios is very similar. Figures 5.2, 5.3, 5.4 and 5.5 show the aggregate offered load arriving at switch S1 (destined for the 40 Mbps output link of switch S1) as a function of time for values of $k = 2$, 5, 10 and 20 respectively. The virtual factor $v$ is fixed at a value of 1.0, so that the performance of the virtual loss-load model essentially mimics that of the original loss-load model.

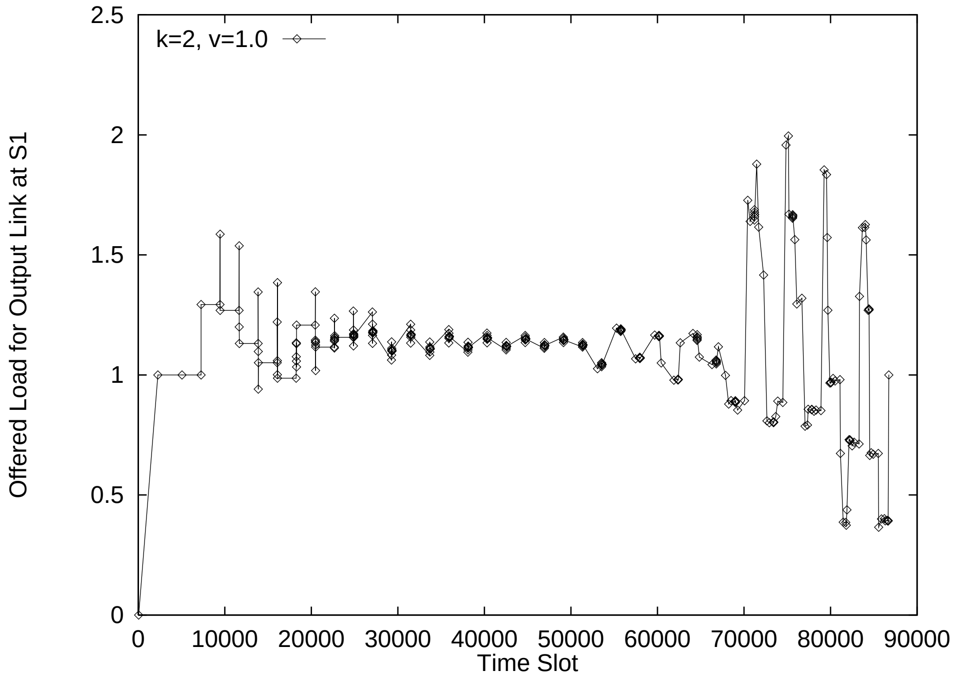



Figure 5.2: Traffic Load at Switch S1 for $k = 2$, $v = 1.0$ in Scenario 1 (Slow Changes in Cross Traffic)

As may be seen from the four graphs, the value of $k$ affects the dynamics, the responsiveness, and the level of the overload in the virtual loss-load model. In general, the value of $k$ determines the level of overshoot produced over the "speed limit" set by the value of $v$ (which is 1.0 in these four graphs). The effect of decreasing the value of $v$ (not shown in these graphs) would be to shift the entire graph vertically downward, so that the load level can be maintained closer to capacity.

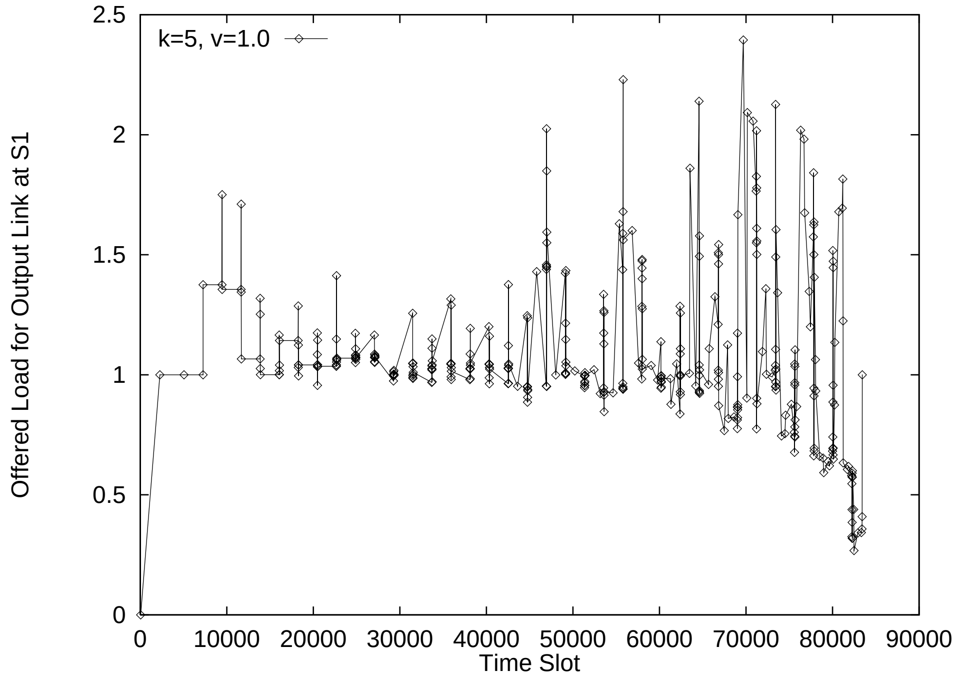



Figure 5.3: Traffic Load at Switch S1 for $k = 5$, $v = 1.0$ in Scenario 1 (Slow Changes in Cross Traffic)

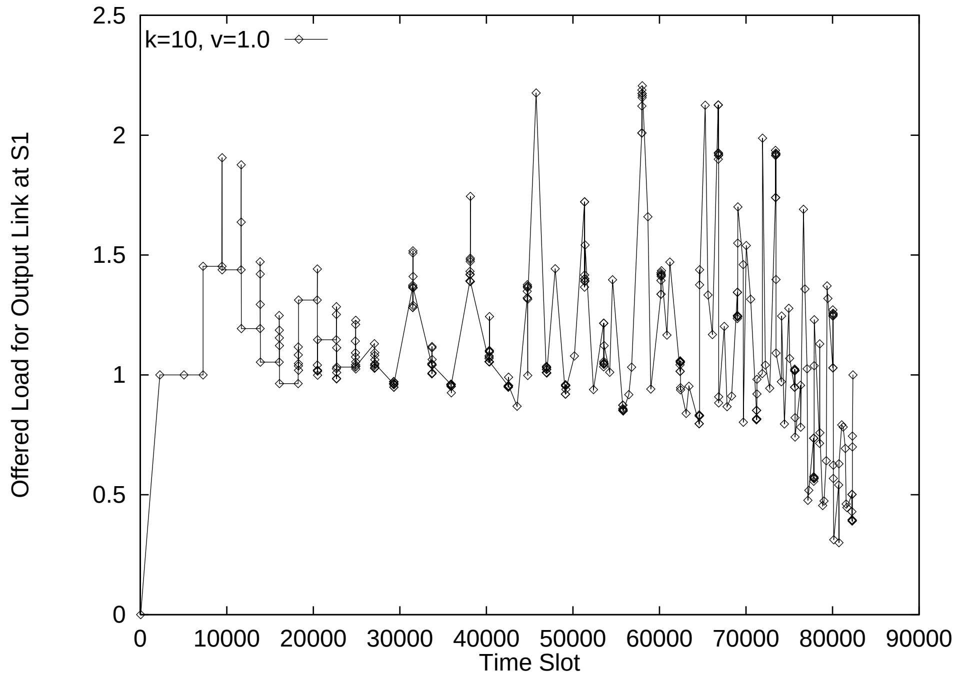



Figure 5.4: Traffic Load at Switch S1 for $k = 10$, $v = 1.0$ in Scenario 1 (Slow Changes in Cross Traffic)

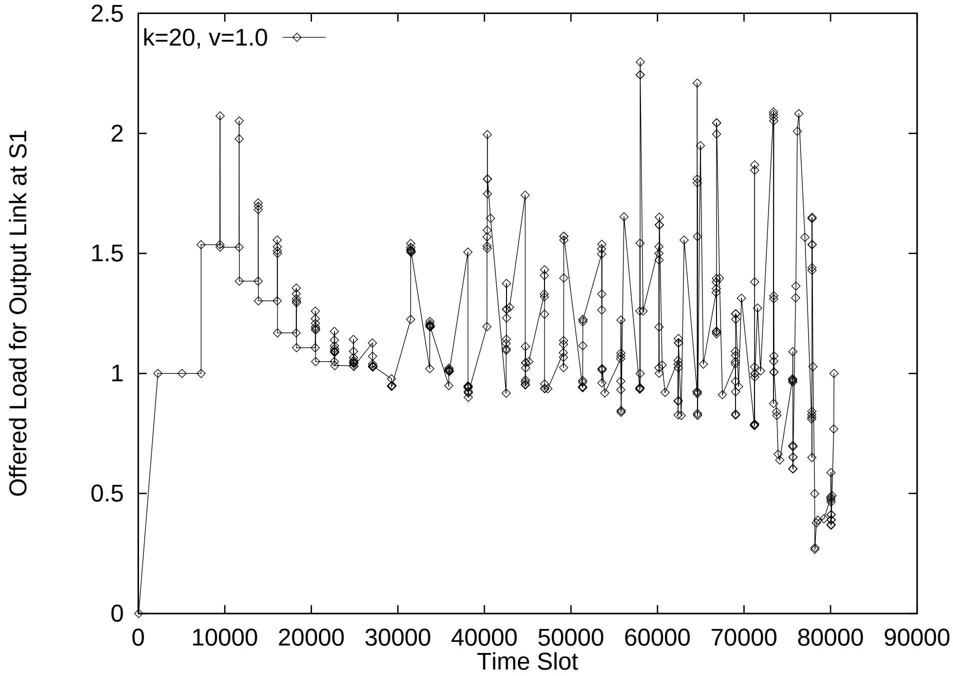



Figure 5.5: Traffic Load at Switch S1 for $k = 20$, $v = 1.0$ in Scenario 1 (Slow Changes in Cross Traffic)

In each of the four figures, there are three regions of interest:

- In the first region, approximately from 0 ms to 500 ms, connections start to arrive at the switch. It can be seen that as connection 1 begins, the load at the output link of the switch rises to 1.0, indicating that the entire bandwidth of the link is made available to that connection. As further connections arrive, the load starts to fluctuate as different connections compete for the bandwidth and adjust their load based on the response time model. Each new arrival results in a sudden increase (the vertical spike) in the offered load at S1, followed by a decrease in the load as senders react to the new load. The magnitude of the spikes and the responsiveness of the senders depends on $k$. For a value of $k = 2$, for example, the load rises to a maximum of about 1.58. For $k = 20$, the load rises to as much as 2.07.

- In the second region, approximately from 500 ms to 1000 ms, connections attempt to reach network equilibrium. The equilibrium achieved is always greater than 1.0 because of the aggressiveness of the loss-load model, and always less than $\frac{k+1}{k}$ because of the more conservative nature of the response time model [39]. It may be seen from the four graphs that true equilibrium is not completely attained, and there is always some oscillation. Larger values of $k$ take more time to reach stability, and the oscillation is much worse. A value of $k = 2$ provides the most stable operating point, as convergence to network equilibrium has been shown to be fastest for small values of $k$ [38]. The overall load for this value of $k$ varies from about 1.11 to 1.15. If the raw throughput model had been employed instead of the response time model at the senders, the expected load in this region for a value of $k = 2$ would have been $\frac{k+1}{k} = 1.5$.

- In the third region, spanning 1000 ms to the end of the simulation, connections start departing. As each connection departs, the other connections obtain the information about increased availability of network bandwidth, and try to operate at a greater rate. This causes a period of overload and oscillation once again as rate adjustments occur. Larger values of $k$ cause much greater overload and oscillation. Also, for large values of $k$, the connections start departing even before equilibrium is achieved, causing much greater overload. For $k = 20$, for example, the load reaches a value as high as about 2.35 in this region.

Larger values of $k$ are clearly undesirable in all three regions. Large values of $k$ are less responsive to dynamic load, and much more susceptible to large transient effects and oscillations, especially when there are large feedback delays in the network. In all three regions, lower values of $k$ perform better in controlling the amount of overload produced. Reducing the overload using small values of $k$ has two main implications for connections. First, as the overload is less, the overall loss for a connection for a given rate of operation is also less. Second, since lower values of $k$ indicate lesser 'punishment' at the switches, the loss rate for a given rate of operation is lower for lower values of $k$. The study of the dynamics of the model is borne out in the experiments, where lower values of $k$ have much lower cell loss rates. It would thus appear that lower values of $k$ are more suitable in networks where network conditions are dynamic.

The following are the results obtained for Scenario 1:

**File Transfer Time**

Figure 5.6 shows the file transfer time for connection 1 for various values of $k$ and $v$ for Scenario 1. The optimal file transfer time for connection 1 in this scenario is 1444 ms.

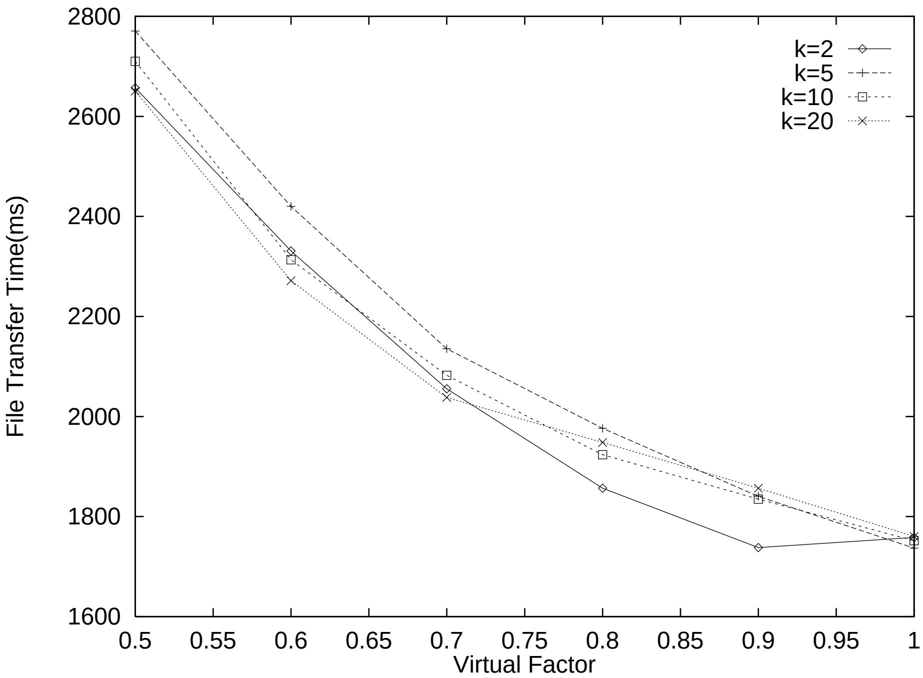



Figure 5.6: File Transfer Time for Connection 1 in Scenario 1 (Slow Changes in Cross Traffic) Without Delayed Resolution

The file transfer time falls sharply with increasing values of $v$. For example, at $k$ =2, as $v$ increases from 0.5 to 0.6, the file transfer time falls from 2656 ms to 2330 ms. This decline is understandable, since more bandwidth is made available to connections as $v$ increases. However, the decline in the file transfer time levels off with larger $v$. This occurs as the connections use all available bandwidth, and with increased rates of operation, no further gains are possible since the network has already been exploited to capacity. In fact, with increased values of $v$, the file transfer time performance actually worsens for $k = 2$ (the file transfer time in this case increases from 1737 ms at $v = 0.9$ to 1758 ms at $v = 1.0$). This is because the transfer time for a connection is composed of both transmission time as well as retransmission time. Increased rates of operation may lead to cell loss, causing retransmission time to increase, thereby increasing the file transfer time. For other values of $k$, however, increased values of $v$ result in better file transfer time, indicating that the cost of retransmitting cells is not high even though increased values of $k$ result in greater loss. Over the entire range of operation of $v$, lower values of $k$ seem better in general, which can be explained by the dynamics of the virtual loss-load model as described earlier. It can also be seen that the file transfer time behaviour in this case differs somewhat from that obtained in the simple two-sender scenario described in Chapter 4. In that case, the optimum file transfer time occurs at a value of $v = \frac{k}{k+1}$. This is not true, however, in this scenario for several reasons. First, and most important, large feedback delays cause a large difference between the loss rate perceived by a sender, and the actual cell loss rate for a sender at the switch. This difference is worsened in the presence of dynamic network conditions, where network conditions change within short intervals of time. Second, senders use the response time model in selecting their rate of operation, as opposed to the raw throughput model. The relation $v = \frac{k}{k+1}$ for optimal operation holds true only when all senders maximise throughput using the raw throughput model.

**Cell Delay**

Figures 5.7 and 5.8 show the mean cell delay and the coefficient of variation of delay for connection 1 for various values of $k$ and $v$ for Scenario 1.

As may be seen from the figure, mean cell delay increases as $v$ increases. However, the increase in delay over the entire range of $v$ is fairly low. For example, as $v$ increases from 0.5 to 1.0 and for $k = 2$, the mean cell delay increases from 0.12 ms to only 0.41 ms. The low mean cell delay occurs because the virtual loss-load model, in the absence of the delayed resolution scheme, discards cells before the buffers fill up over all ranges of $v$. The C.O.V. of delay also varies marginally over the range of $v$. From the graphs, it appears that there is no strong relationship between mean cell delay or C.O.V. of delay with different values of $k$. Also, as $v$ increases, mean cell delay may increase or decrease, but the overall change is small. For example, for $k$ =2, as $v$ increases from 0.8 to 1.0, the mean cell delay increases from 0.16 to 0.45 before falling to 0.41. The erratic increase or decrease is because the virtual loss-load

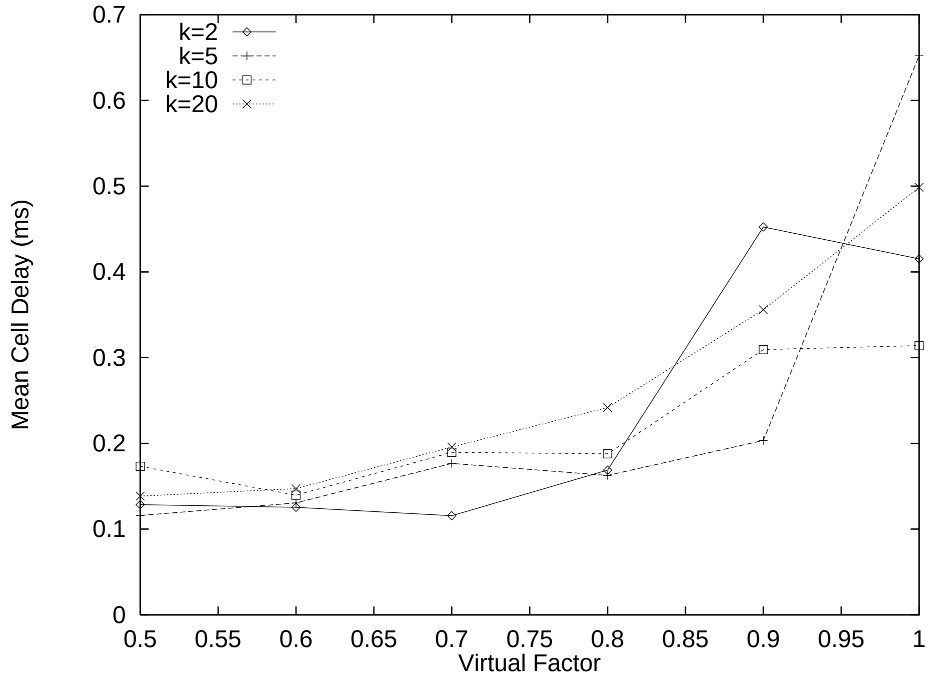



Figure 5.7: Mean Cell Delay for Connection 1 in Scenario 1 (Slow Changes in Cross Traffic) Without Delayed Resolution

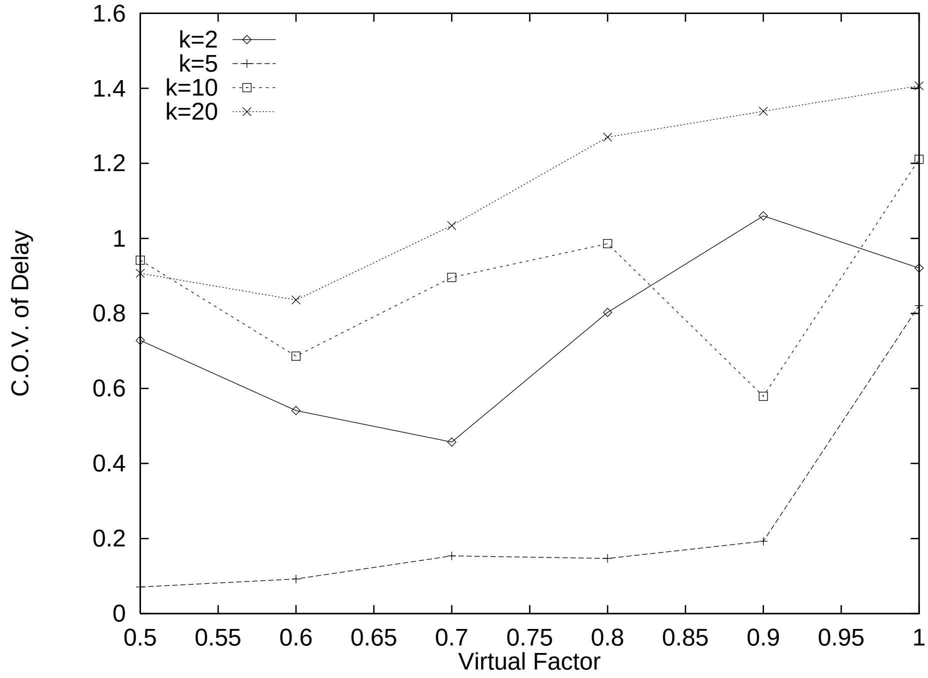



Figure 5.8: C.O.V. of Delay for Connection 1 in Scenario 1 (Slow Changes in Cross Traffic) Without Delayed Resolution

model discards arriving cells randomly, with a cell loss probability associated with the current rate of operation of the sender.

In all the experiments conducted in this section, the C.O.V. of delay shows similar trends, (i.e., variations occur without strong relation to the performance factors) and is therefore not considered in the remaining experiments in this section.

**Cell Loss Probability**

Figure 5.9 shows the overall cell loss probability for connection 1 for various values of $k$ and $v$ for Scenario 1.

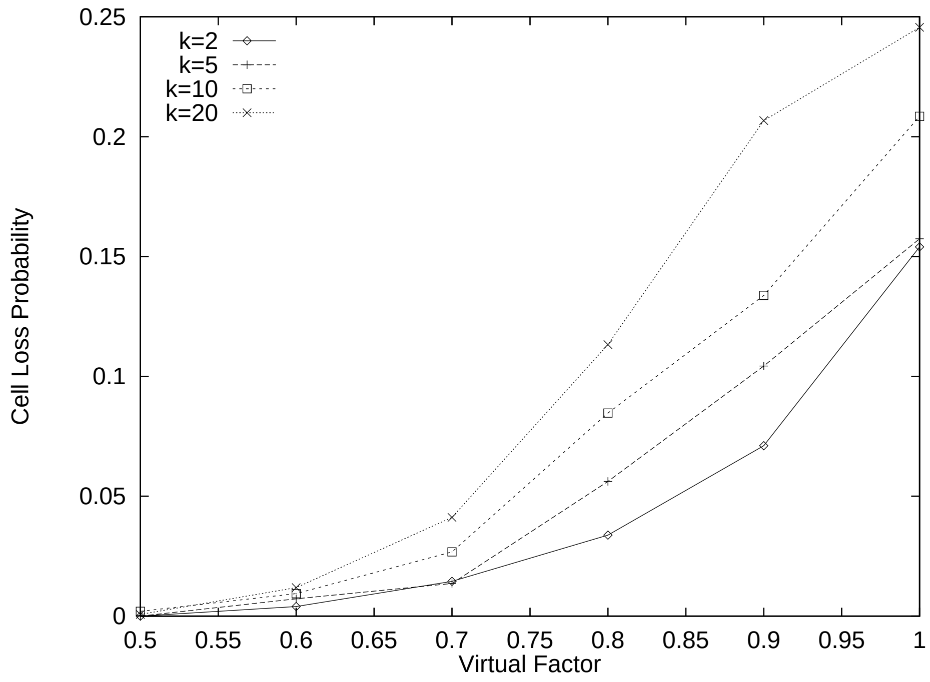



Figure 5.9: Cell Loss Probability for Connection 1 in Scenario 1 (Slow Changes in Cross Traffic) Without Delayed Resolution

Operating at low values of $v$ results in reduced cell losses as compared to operating at larger values. Also, for a given value of $v$, low values of $k$ result in lower cell loss. For example, at $v$ = 0.9, the cell loss probability is 0.10 for $k$ = 5, but is 0.20 for $k$ = 20. It may be noted that the cell loss probability at $v$ = 1.0, for a given value of $k$, differs considerably from the value of $\frac{1}{k+1}$ predicted by the raw throughput model at $v$ = 1.0. The response time model being used in the current scenario, although different from the raw throughput model, should also yield C.L.P's which are inversely proportional to $k$. However, the exact opposite behaviour is observed in the current scenario. For example, at $v = 1.0$ and for $k = 2$, the C.L.P. is 0.15, while for $k = 20$, it is 0.25. The main reason for the discrepancy is that the value of $\frac{1}{k+1}$ holds good only when the system reaches equilibrium. As was seen earlier, the period of equilibrium is small compared to the transient periods for the current scenario. It is these transient periods that dominate the cell losses encountered by a connection, and the transient periods are much worse for large values of $k$. Again, unlike the two-sender case, the cell loss probability increases sharply at a value of $v$ less than $\frac{k}{k+1}$. This is due again to the dominance of the transient periods in the network where the cell loss probability is much higher. The two-sender case, on the other hand, is characterised by the network being in a state of equilibrium for most of the time.

It is instructive to compare the file transfer times as well for a given pair of $k$ and $v$. At $v = 0.90$, the file transfer time for $k = 5$ is 1840 ms, while for $k = 20$, it is 1856 ms. This suggests that larger values of $k$ are much more aggressive, and lose many more cells than lower values of $k$; however, their aggressive nature does not cost much in terms of the file transfer time achieved. However, low values of $k$ are to be preferred as they operate closer to network capacity and cause little overhead in the network by reducing the number of cells transmitted.

Since it is the green cell loss probability that actually measures the deviation from the expected cell loss probability for a sender, and since the overall cell loss probability was found to be similar to the green cell loss probability in all the experiments, the overall cell loss probability is not considered for the other scenarios in this section.

**Green Cell Loss Probability**

Figure 5.10 shows the green cell loss probability for connection 1 for various values of $k$ and $v$ for Scenario 1.

The green cell loss probability for connection 1 is very similar to the graph for the overall cell loss probability. As can be seen from the figure, the green cell loss probability increases with increasing values of $v$. The increase is slight at first, but more rapid after $v = 0.7$ for all values of $k$. Larger values of $k$ produce far greater green cell loss. For example, at $v = 0.9$, the green cell loss probability for $k = 2$ is 0.07,

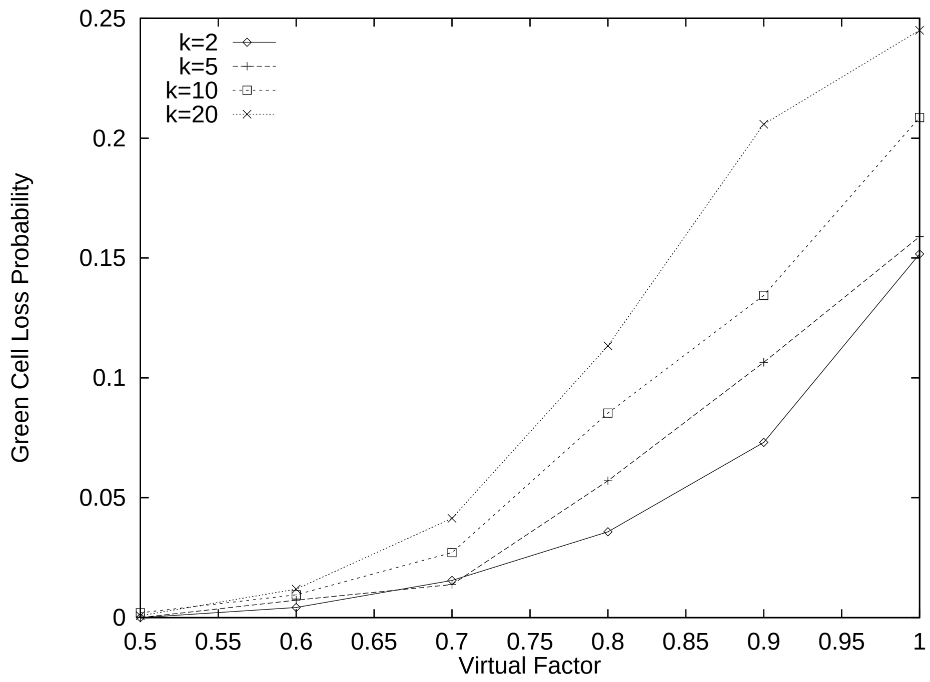



Figure 5.10: Green Cell Loss Probability for Connection 1 in Scenario 1 (Slow Changes in Cross Traffic) Without Delayed Resolution

but for $k = 20$, it is 0.20. This agrees with the analysis of the dynamics of the virtual loss-load model, where larger values of $k$ were shown to be much more aggressive in their performance and resulted in much greater overload at the switches. The experimental results also imply that to maintain cell loss at low values, the virtual factor $v$ needs to be kept at a low value. Low values of $v$, however, cause senders to operate conservatively, causing an increase in the file transfer time for senders.

### 5.3.2 Scenario 2: Sudden Jumps in Cross Traffic

Scenario 2 measures the performance of the congestion control strategy in the presence of rapid changes in bandwidth availability. The configuration shown in Figure 5.1 is used in this scenario too. However, the cross traffic load seen by the primary connection changes very rapidly. Connections 1 and 2 begin at time 0 and 100 ms respectively. Connections 3 through 11 begin every 4 ms after the commencement of connection 2. Connection 1 sends 2 Mbytes of data while connections 2 through 11 send 500 Kbytes of data. The optimal file transfer time for connection 1 is 1444 ms.

The following results were obtained for this scenario:

**File Transfer Time**

Figure 5.11 shows the file transfer time for connection 1 for various values of $k$ and $v$ for Scenario 2.

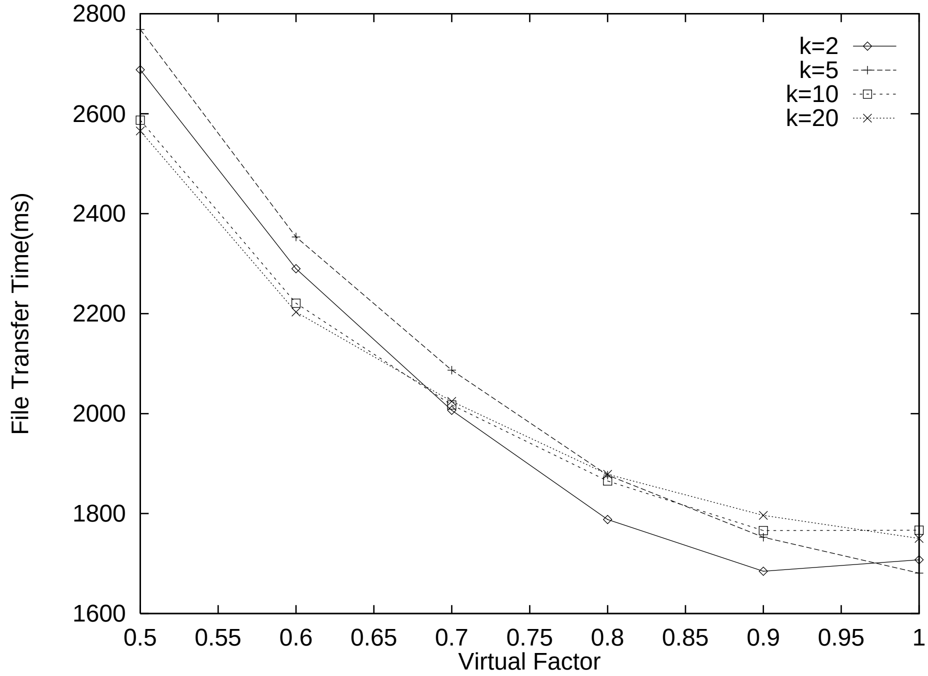



Figure 5.11: File Transfer Time for Connection 1 in Scenario 2 (Sudden Jumps in Cross Traffic) Without Delayed Resolution

The behaviour of the file transfer time is similar to that obtained for Scenario 1. The file transfer time for connection 1 falls as the value of $v$ increases, as network capacity is exploited more fully. Low values of $k$ again perform better than larger values of $k$ as in Scenario 1, for most values of $v$. For increasing values of $v$, the performance of the virtual loss-load model in this scenario is slightly better than in Scenario 1. For example, for $k = 5$ and $v = 0.9$, the file transfer time in this scenario is 1752 ms, whereas it is 1840 ms for the same values of $k$ and $v$ in Scenario 1.

**Cell Delay**

Figure 5.12 shows the mean cell delay for connection 1 for various values of $k$ and $v$ for Scenario 2.

As may be seen, there is a general increase in the mean cell delay with increased

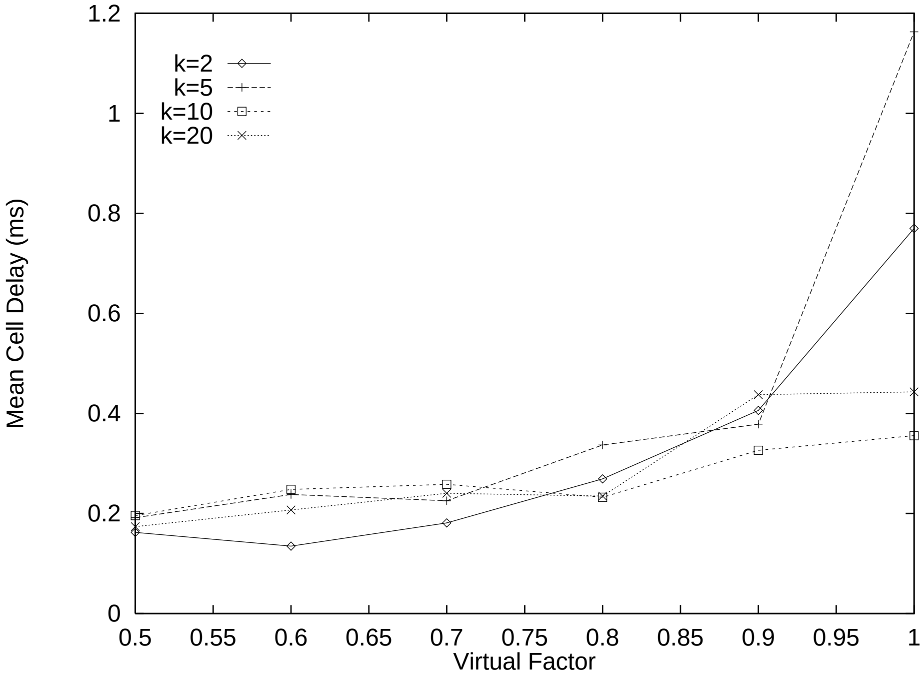



Figure 5.12: Mean Cell Delay for Connection 1 in Scenario 2 (Sudden Jumps in Cross Traffic) Without Delayed Resolution

$v$. However, as in Scenario 1, the mean cell delay is fairly small for all values of $v$, and ranges from a minimum of 0.13 ms for $k = 2$ at $v = 0.5$, to a maximum of 1.16 ms for $k = 5$ at $v = 1.0$. There is also no strong relation between the mean cell delay and $k$. The C.O.V. of delay shows behaviour similar to that described in Scenario 1.

**Green Cell Loss Probability**

Figure 5.13 shows the green cell loss probability for connection 1 for various values of $k$ and $v$ for Scenario 2.

As in Scenario 1, the green cell loss probability increases with increasing $v$. However, the green cell loss probability at low values of $v$ is higher than that for Scenario 1. For example, at $v = 0.6$ and $k = 2$, the green cell loss probability is 0.05 in Scenario 2, compared to 0.0 in Scenario 1. This occurs because the virtual loss-load model is unable to react quickly enough to sudden changes in cross traffic. This results in increased transient effects, causing greater cell loss. At higher values of $v$, there is less difference in the green cell loss probabilities between Scenario 1 and Scenario 2. For example, at $k = 2$, and $v = 1.0$, the green cell loss probabilities in both scenar-

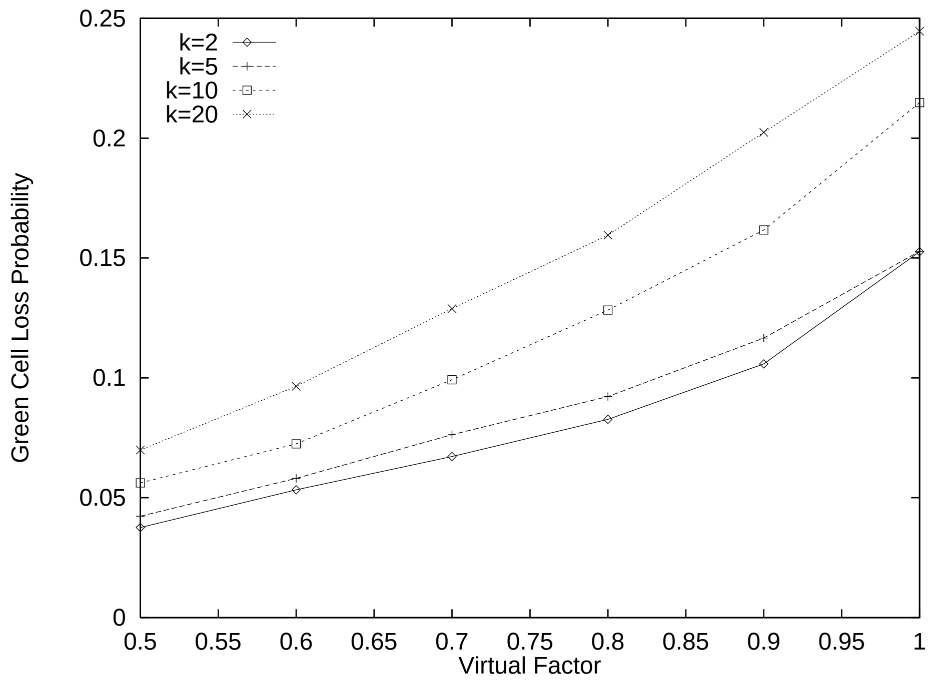



Figure 5.13: Green Cell Loss Probability for Connection 1 in Scenario 2 (Sudden Jumps in Cross Traffic) Without Delayed Resolution

ios 1 and 2 is 0.15. At higher values of $v$, network capacity is already exploited to saturation in both scenarios, causing no appreciable difference in the green cell loss probability.

### 5.3.3 Scenario 3: Migrating Bottlenecks

The scenario used in this experiment is intended to measure the performance of congestion control strategies when the bottleneck moves from one switch in the network to another.

Figure 5.14 shows the configuration used to measure congestion control performance in the presence of migrating bottlenecks. The control connection, connection 1, begins at time 0 and has 2 Mbytes of data to send. Connection 1 begins after 100 ms and sends 1 Mbyte of data. Connections 3 and 4 commence after 250 ms and send 250 Kbytes of data each. Initially, the bottleneck is the link between S1 and S2. When connections 3 and 4 start, however, the bottleneck shifts to the link between S2 and H2. When connections 3 and 4 terminate, the bottleneck shifts back to the

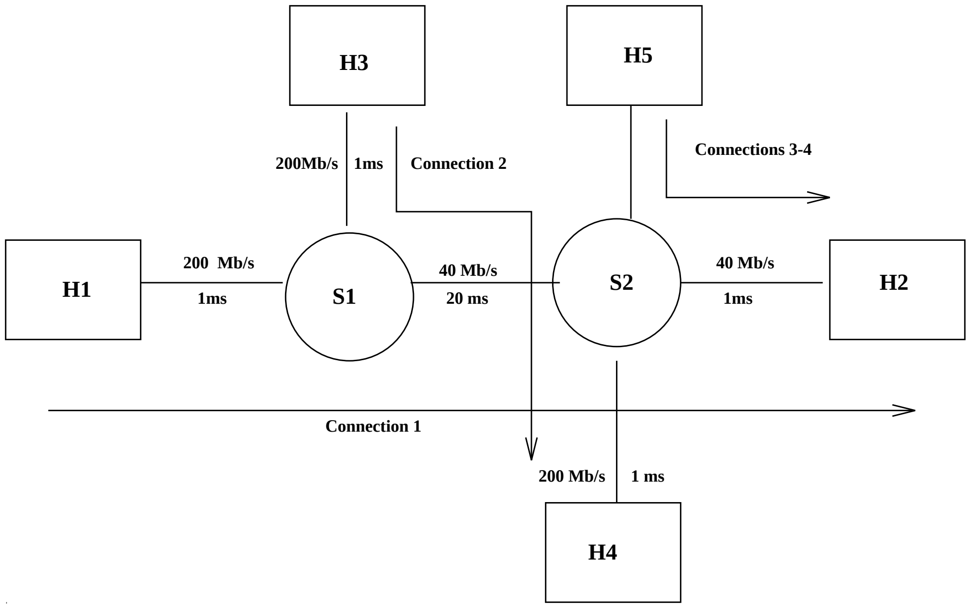



Figure 5.14: Network Configuration used for Scenario 3

link between S1 and S2.

**File Transfer Time**

Figure 5.15 shows the file transfer time for connection 1 for various values of $k$ and $v$ for Scenario 3. The optimal file transfer time for connection 1 is 644 ms.

File transfer time falls with increasing values of $v$. Unlike the earlier two scenarios, however, larger values of $k$ seem to perform slightly better in reducing file transfer time for most values of $v$. This is attributable to the fact that there are fewer cross connections than in the earlier scenarios (only 3 in this scenario as compared to 10 in scenarios 1 and 2). As was seen in the study of the dynamics of the behaviour of the virtual loss-load model, the number and speed with which connections arrive affects the degree of overload generated, with larger values of $k$ affecting the degree of overload more severely. With a reduced number of cross-connections, large values of $k$ do not cause as much overload as in the earlier scenarios, ensuring that the amount of time senders spend in retransmitting cells is reduced.

**Cell Delay**

Figure 5.16 shows the mean cell delay for connection 1 for various values of $k$ and $v$ for Scenario 3. As in the earlier scenarios, there is a general increase in the mean

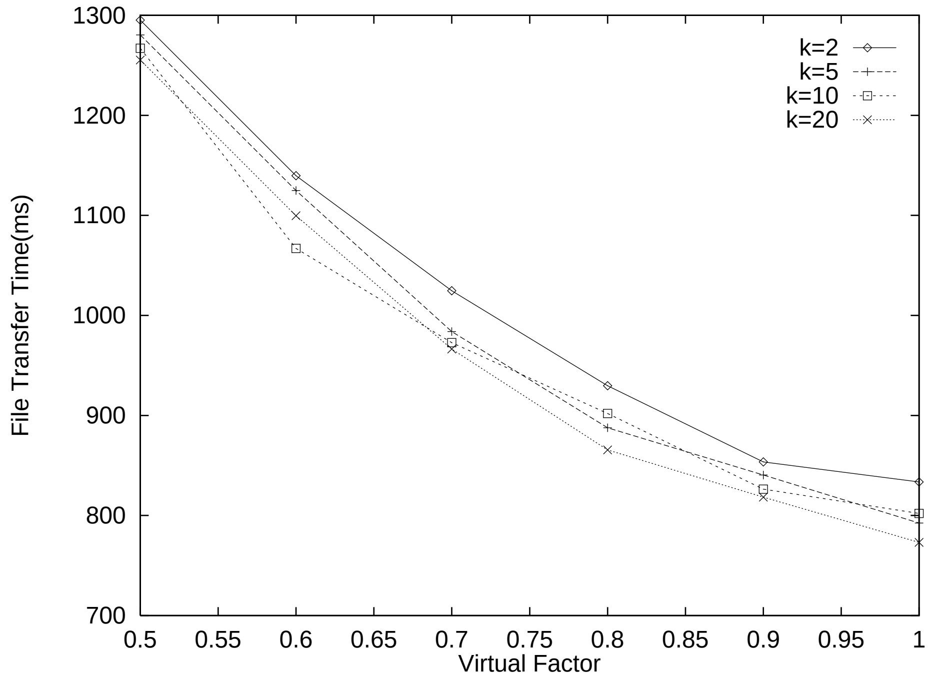



Figure 5.15: File Transfer Time for Connection 1 in Scenario 3 (Migrating Bottlenecks) Without Delayed Resolution

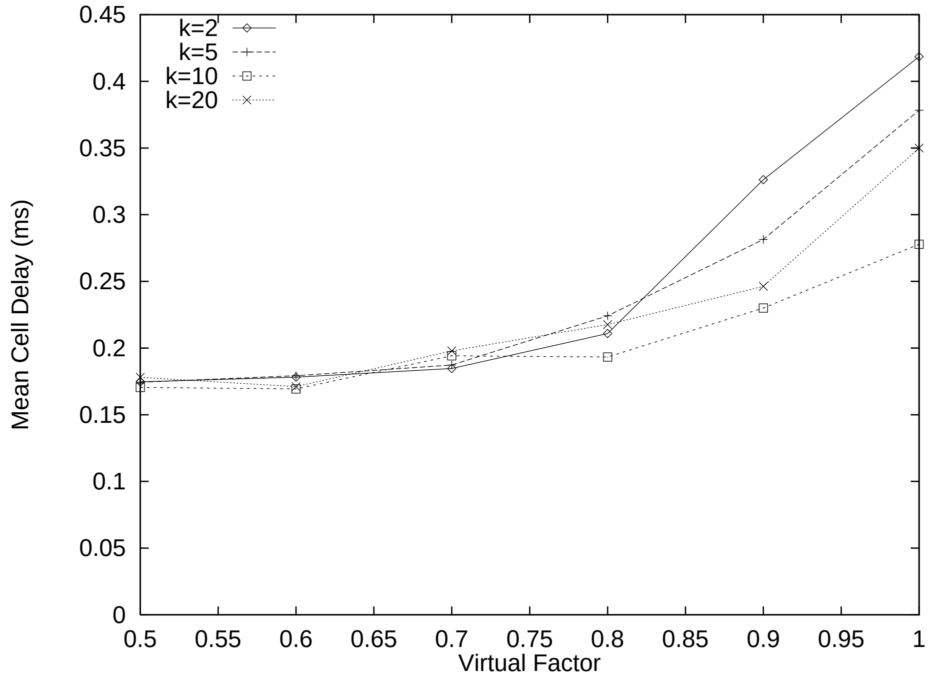



Figure 5.16: Mean Cell Delay for Connection 1 in Scenario 3 (Migrating Bottlenecks) Without Delayed Resolution

cell delay with increasing values of $v$. However, the mean cell delay is fairly small for all values of $v$, and ranges from a minimum of 0.17 ms for $k = 10$ at $v = 0.5$, to a maximum of 0.42 ms for $k = 2$, at $v = 1.0$. Again, there is no strong relation between the mean cell delay and $k$. The C.O.V. of delay shows behaviour similar to that described in Scenario 1.

**Green Cell Loss Probability**

Figure 5.17 shows the green cell loss probability for connection 1 for various values of $k$ and $v$ for Scenario 3.

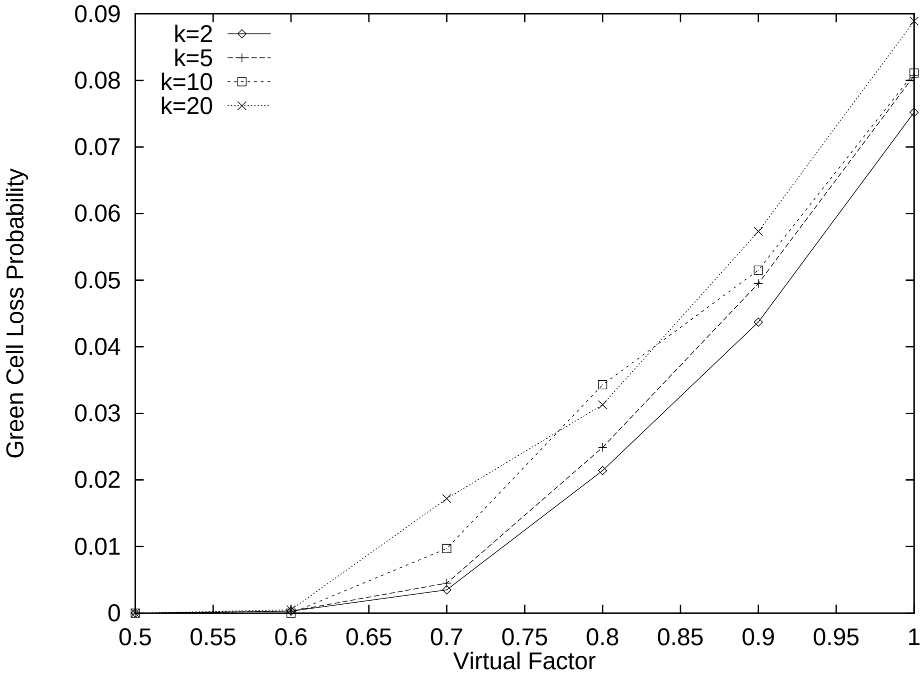



Figure 5.17: Green Cell Loss Probability for Connection 1 in Scenario 3 (Migrating Bottlenecks) Without Delayed Resolution

As in scenarios 1 and 2, the green cell loss probability increases with $v$. The green cell loss probability is low for small values of $v$, but increases sharply when $v$ is larger than 0.7. However, for a given value of $v$ and $k$, the green cell loss probability in this scenario is substantially lower than in the earlier scenarios. For example, at $k = 5$, and at $v = 1.0$, the green cell loss probability is 0.08, whereas in Scenario 1 it is as high as 0.16. The lower loss is a result of the smaller number of cross-connections in this case, as well as the migration of the bottleneck. As the bottleneck moves towards the receiver, the feedback delay decreases, and senders can react faster to load changes, thereby reducing loss. Low values of $k$ again result in reduced loss.

### 5.3.4 Summary

This subsection summarises some of the key results for the virtual loss-load model for the three scenarios considered:

- The virtual factor $v$ controls to a large extent the amount of cell loss and delay encountered by a sender. Low values of $v$ result in lower delay and loss, but do not make effective use of the bandwidth available in the network. Values of $v$ close to 1.0, on the other hand, result in considerable loss. Values of $v$ in the range 0.7 and 0.9 represent the region where the (green) cell loss probability and the mean cell delay are relatively small. For scenario 2, for example, the green cell loss probability and the mean cell delay for $k = 2$, and at $v = 0.8$ are 0.07 and 0.26 ms, which compares with 0.15 and 0.77 ms at $v = 1$, for the same value of $k$. The reduced sender loss and delay is achieved with comparatively little degradation in file transfer performance: 1788 ms for $k = 2$, and $v = 0.8$, compared to 1704 ms for $k = 2$ and $v = 1.0$. Since $v = 1.0$ represents the operation of the original loss-load model, it suggests that using a reduced value of $v$ can provide better performance in reducing cell loss and delay, with little degradation in file transfer time.

- Low values of $k$ seem more suitable in reducing overall cell loss in the network. This effect is due to the dynamics of the loss-load congestion control algorithm. Low values of $k$ produce lower network overload when connections arrive, converge to network equilibrium faster when connections stabilise, and produce lower oscillations in the network when connections depart.

- File transfer time almost always decreases with increased values of $v$. For the scenarios considered in the experiments, this implies that even though network overload increases, and there are more retransmissions, it is still worthwhile to be aggressive in the approach. Also, there is not much difference between the file transfer times for different values of $k$ for a given value of $v$. This indicates that if the file transfer time is the only performance metric of importance, the primary determinant of performance is the factor $v$, and not the factor $k$. However, lower values of $k$ achieve the transfer with largely reduced sender losses, indicating that they operate at or slightly above capacity. It may be noted that the experiments do not take into account processing overheads for transmitted packets. Since each packet consumes resources in the network, and since the total number of packets transmitted for larger values of $k$ is higher (due to the increased C.L.P.), selecting low values of $k$ becomes more attractive.

- The absence of the delayed resolution strategy results in fairly low mean cell delay and variance in the delay for all values of $k$ and $v$. The mean cell delay does not seem to have any strong relation to values of $k$ and $v$.

## 5.4 Virtual Loss-Load Model Performance with Delayed Resolution

This section studies the performance of the virtual loss-load model for different values of $k$ and $v$, with the delayed resolution strategy employed at the switches. The delayed resolution strategy allows the switch to postpone cell discard decisions until buffers are full, in the hope of making better cell discard decisions. The delayed resolution strategy is studied for two different values of buffer size: 440 (representing a delay-bandwidth product of packets, and referred to hereafter as Case 1) and a size of 44 (to represent a much lower value, and referred to hereafter as Case 2). The performance of the virtual loss-load model with delayed resolution is studied for the same three benchmark scenarios as in Section 5.3.

### 5.4.1 Scenario 1: Slow Changes in Cross Traffic

The performance of the virtual loss-load model is studied for the scenario described in Section 5.3.1. The following are the results:

**File Transfer Time**

Figure 5.18 shows the file transfer times for connection 1 for Cases 1 and 2.

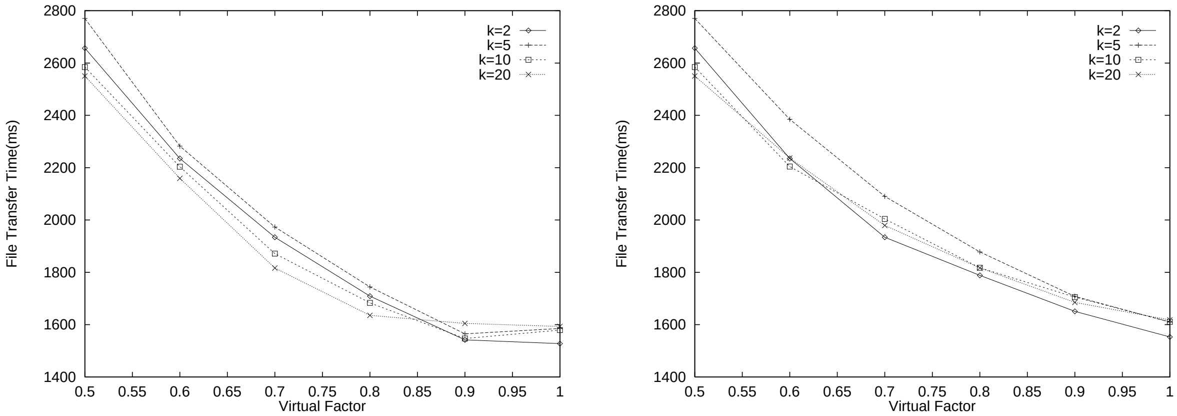



Figure 5.18: File Transfer Time for Connection 1 in Scenario 1 (Slow Changes in Cross Traffic) With Delayed Resolution. (Left: Buffer Size = 440; Right: Buffer Size = 44)

The file transfer time declines with increasing $v$, with the decline being sharper at low values of $v$. As $v$ ranges from 0.5 to 0.7, identical file transfer times are obtained in Cases 1 and 2, indicating that a much smaller buffer size of 44 is sufficient for cell discard decision making at the switch. However, as the value of $v$ increases beyond 0.7, the file transfer time for Case 1 is better than for Case 2. For example, the file transfer time for Case 1 at $k = 2$ and $v = 0.9$ is 1541 ms, whereas for Case 2, it is 1650 ms for the same values of $k$ and $v$. Case 1 performs better since fewer cells are lost as a result of the increased buffer size. However, there is not much difference between the two cases at $v = 1.0$. For $k = 2$ and $v = 1.0$, the file transfer time is 1527 ms for Case 1, and is 1553 ms for Case 2. This is because for large values of $v$, the buffer is full most of the time in both cases, and the cell loss rates in both strategies are similar.

Use of the delayed resolution strategy results in better file transfer time than use without the strategy. For example, for $k = 2$ and $v = 1.0$, the file transfer time is

1758 ms without the delayed resolution (see Figure 5.6), which is much greater than that observed for Cases 1 and 2 (1527 ms and 1533 ms respectively).

**Cell Delay**

Figure 5.19 shows the mean cell delay, while Figure 5.20 shows the C.O.V. of delay for connection 1 in Cases 1 and 2 in Scenario 1.

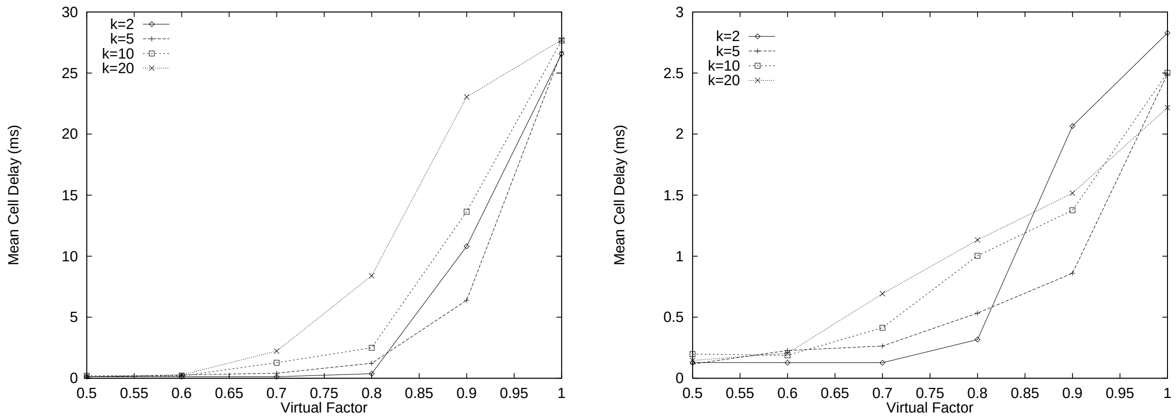



Figure 5.19: Mean Cell Delay for Connection 1 in Scenario 1 (Slow Changes in Cross Traffic) With Delayed Resolution. (Left: Buffer Size = 440; Right: Buffer Size = 44)

As can be seen from the figure, mean cell delay increases as $v$ increases for both Cases 1 and 2. This increase occurs because the buffers start filling up. For both Cases 1 and 2, the mean cell delay is identical up to $v = 0.7$. Beyond this value, however, the increase in cell delay is much greater for Case 1 than it is for Case 2. At $k = 2$ and $v = 0.9$, for example, the mean cell delay is 10.18 ms for Case 1, while it

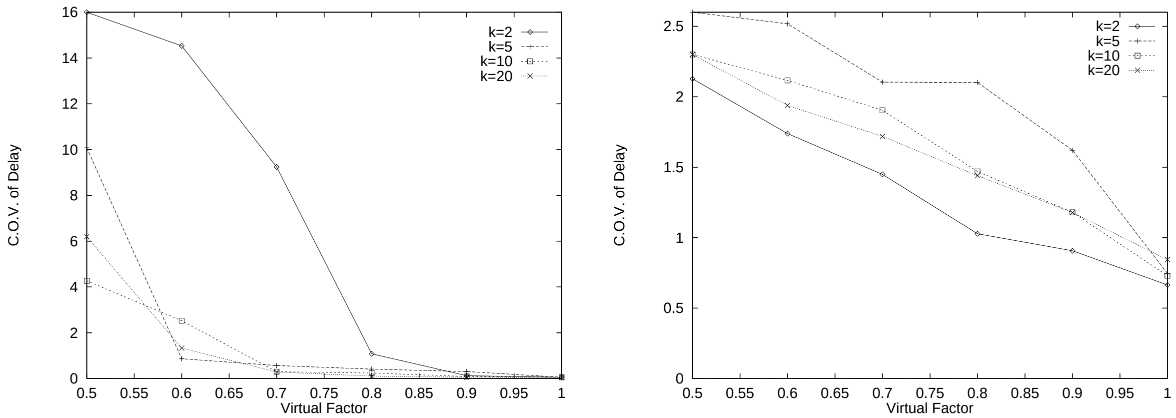



Figure 5.20: C.O.V of Delay for Connection 1 in Scenario 1 (Slow Changes in Cross Traffic) With Delayed Resolution. (Left: Buffer Size = 440; Right: Buffer Size = 44)

is 2.06 ms for Case 2. This is because the number of buffers in Case 1 is much more than in Case 2.

The difference in mean cell delay becomes worse with increasing values of $v$. At $v = 1.0$, the mean cell delay is 26.56 ms for Case 1, while it is only 2.82 ms for Case 2. This suggests that the gain in having an increased buffer is not much, since cell losses in Case 2 are not substantially more than in Case 1 (see Figure 5.21) for given values of $k$ and $v$, but cell delays in Case 2 are substantially less. Both cases, however, perform much worse when compared to the virtual loss-load model without the delayed resolution strategy, where mean cell delays are generally below 1 ms.

The C.O.V. of delay for both cases falls with increasing values of $v$ (see Figure 5.20). This is because the buffers at the switch become full, and arriving cells will almost always encounter a full buffer. As a result, there is little difference in the delay encountered by cells as they arrive at the receiver.

In all the experiments conducted in this section, the C.O.V. of delay shows similar behaviour and is therefore not presented in the remaining experiments reported in this section.

**Cell Loss Probability**

Figure 5.21 shows the cell loss probabilities for connection 1 for Cases 1 and 2 in Scenario 1.

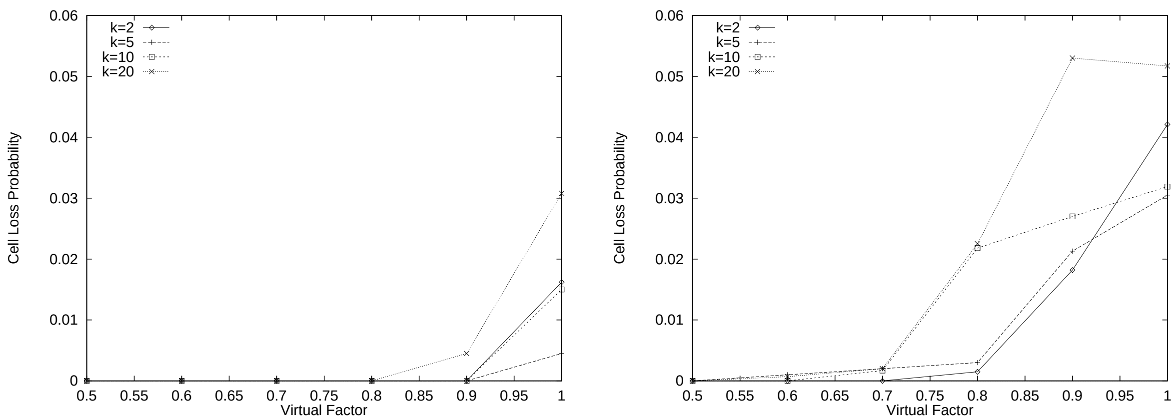



Figure 5.21: Cell Loss Probability for Connection 1 in Scenario 1 (Slow Changes in Cross Traffic) With Delayed Resolution. (Left: Buffer Size = 440; Right: Buffer Size = 44)

Operating at low values of $v$ results in reduced cell losses as compared to operating at larger values. Also, for a given value of $v$, cell loss is lower for low values of $k$. As in the case of the green cell loss probability, use of even a small buffer can reduce the cell loss substantially.

It can also be seen that the delayed resolution strategy reduces the cell loss probability significantly compared to use without the strategy (compare Figures 5.21 and 5.9). The implication of this is that use of the delayed resolution strategy allows for a more aggressive mode of operation as $v$ can be increased to a larger value to obtain a given cell loss probability. In all the experiments conducted, the overall cell loss probability was very similar to the green cell loss probability. Therefore, the rest of the experiments in this section do not consider the cell loss probability metric.

**Green Cell Loss Probability**

Figure 5.22 shows the green cell loss probabilities for connection 1 for cases 1 and 2 in Scenario 1.

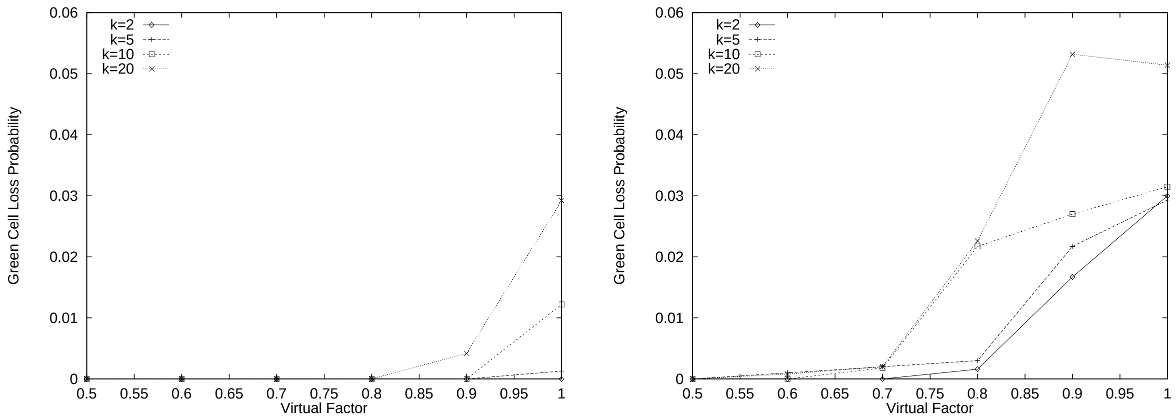



Figure 5.22: Green Cell Loss Probability for Connection 1 in Scenario 1 (Slow Changes in Cross Traffic) With Delayed Resolution. (Left: Buffer Size = 440; Right: Buffer Size = 44)

For both graphs shown, the green cell loss probability is low for $v \leq 0.7$, and increases sharply for larger values of $v$. Case 1 performs better than Case 2 in reducing the green cell loss probability for low values of $v$. For $k = 2$, for example, the green cell loss probability is 0 for all values of $v$ for Case 1. For Case 2, however, the green cell loss probability for $k = 2$ increases from 0.0 at $v = 0.7$ to 0.03 at $v = 1.0$.

Increased values of $k$ show increased cell loss, and the loss occurs at lower values of $v$. It may be seen that even though the buffer size in Case 2 is $\frac{1}{10}$th the value in Case 1, the green cell loss probability in Case 2 is not much worse than in Case 1, with the difference depending mainly on the value of $k$. This suggests that the use of even small buffer sizes may be adequate in this scenario.

The green cell loss probability reduces with the delayed resolution strategy. For $k = 2$ and $v = 1.0$, for example, the green cell loss probability is 0.15 without the delayed resolution strategy (see Figure 5.10). With delayed resolution, the green cell loss probability is 0.0 for Case 1, while it is 0.03 for Case 2.

### 5.4.2 Scenario 2: Sudden Jumps in Cross Traffic

The performance of the virtual loss-load model with delayed resolution is studied for the scenario described in Section 5.3.2. The following are the results obtained for the experiments conducted for this scenario:

**<u>File Transfer Time</u>**

Figure 5.23 shows the file transfer times for connection 1 for Cases 1 and 2.

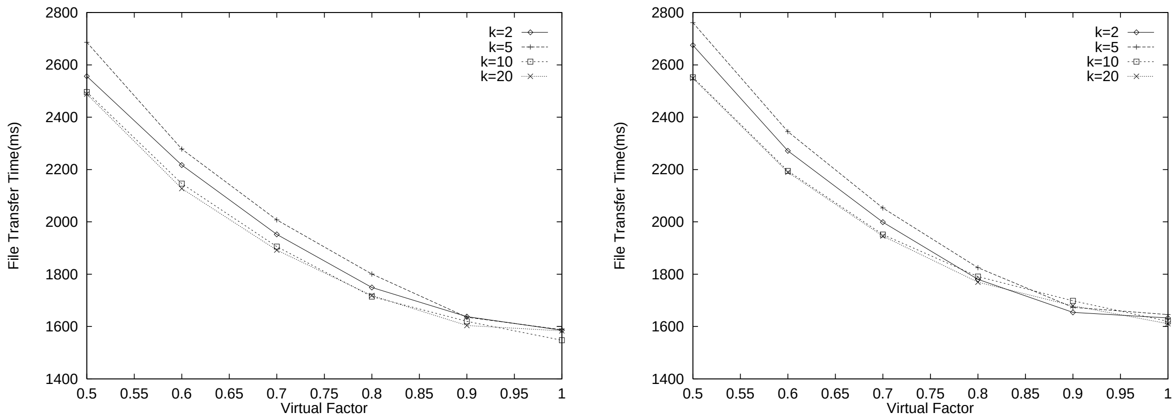



Figure 5.23: File Transfer Time for Connection 1 in Scenario 2 (Sudden Jumps in Cross Traffic) With Delayed Resolution. (Left: Buffer Size = 440; Right: Buffer Size = 44)

As in the earlier scenario, the file transfer time declines for increased values of $v$, with the decline being sharper at the initial values. In contrast to Scenario 1, the file transfer times for the two cases differ for even low values of $v$. For $k = 2$ and $v =$

0.5, the file transfer time is 2556 ms for Case 1, while it is 2675 ms for Case 2. This is because, as connections arrive rapidly, the buffers get exhausted quickly in Case 2, while the increased number of buffers in Case 1 ensures that the cell loss remains lower. At higher values of $v$, however, there is no substantial difference in the transfer time. For example, for $k = 2$ and $v = 1.0$, the file transfer time is 1585 ms for Case 1, while it is 1632 ms for Case 2. The reduced difference here is a result of increased load encountered in both cases over the entire period of the run.

The choice of the delayed resolution strategy results in better performance. For $k = 2$ and $v = 1.0$, the virtual loss-load model without the delayed resolution strategy shows a file transfer time of 1707 ms.

**Cell Delay**

Figure 5.24 shows the mean cell delay for connection 1 in Cases 1 and 2 for Scenario 2.

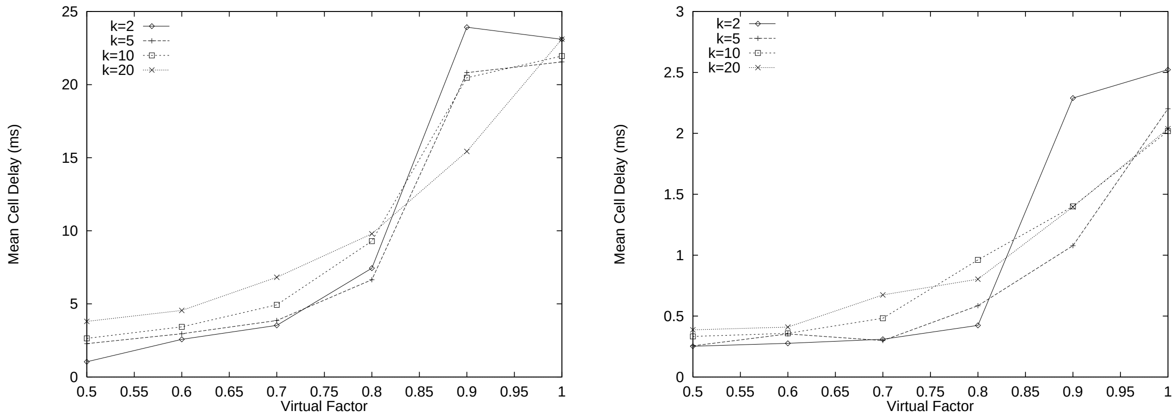


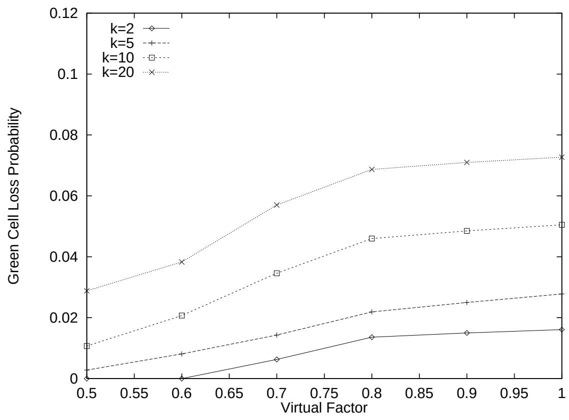

Figure 5.24: Mean Cell Delay for Connection 1 in Scenario 2 (Sudden Jumps in Cross Traffic) With Delayed Resolution. (Left: Buffer Size = 440; Right: Buffer Size = 44)

As in Scenario 1, the mean cell delay increases for increasing values of $v$. The mean cell delay is much worse for Case 1. For $k = 2$ and $v = 1.0$, the mean cell delay is 23.10 ms for Case 1, while it is only 2.52 ms for Case 2. Again, this argues for the adoption of smaller buffer sizes which results in much greater savings in terms of cell delay with minimal increase in the cell loss rate. Non-adoption of the delayed resolution strategy, however, results in the best performance in terms of mean cell delay. For $k = 2$ and $v = 1.0$, the mean cell delay in that case is only 0.77 ms (see

Figure 5.12).

**Green Cell Loss Probability**

Figure 5.25 shows the green cell loss probabilities for connection 1 for Cases 1 and 2 in Scenario 2.



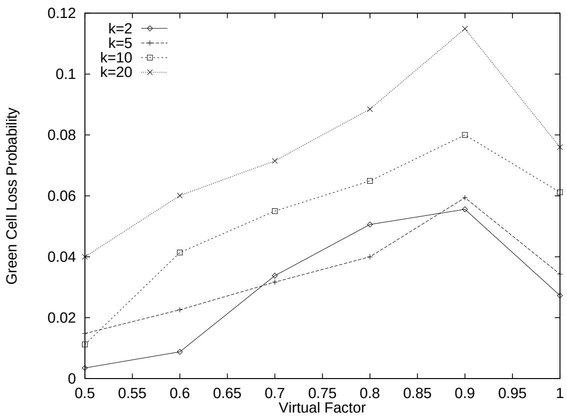




Figure 5.25: Green Cell Loss Probability for Connection 1 in Scenario 2 (Sudden Jumps in Cross Traffic) With Delayed Resolution. (Left: Buffer Size = 440; Right: Buffer Size = 44)

The green cell loss probability increases with increasing values of $v$. Increased buffering helps in reducing the green cell loss probability for given values of $k$ and $v$. The difference in the amount of green cell loss encountered between Cases 1 and 2, however, is not substantial considering the difference in the sizes of the buffer employed in each case. For example, for $k = 2$ and $v = 1.0$, the green cell loss probability is 0.01 for Case 1 while it is 0.02 for Case 2. Use of lower values of $k$ generally result in lower green cell loss probabilities for all values of $v$.

As in Scenario 1, both cases show better performance with the delayed resolution strategy. For example, without the delayed resolution strategy, for $k = 2$ and $v = 1.0$, the green cell loss probability is 0.15. which is substantially more than that of Cases 1 and 2.

### 5.4.3 Scenario 3: Migrating Bottlenecks

The performance of the virtual loss-load model with delayed resolution was also studied for the migrating bottlenecks scenario described in Section 5.3.3. The following are the results obtained for the experiments conducted for this scenario:

**File Transfer Time**

Figure 5.26 shows the file transfer times for connection 1 for Cases 1 and 2 in Scenario 3.

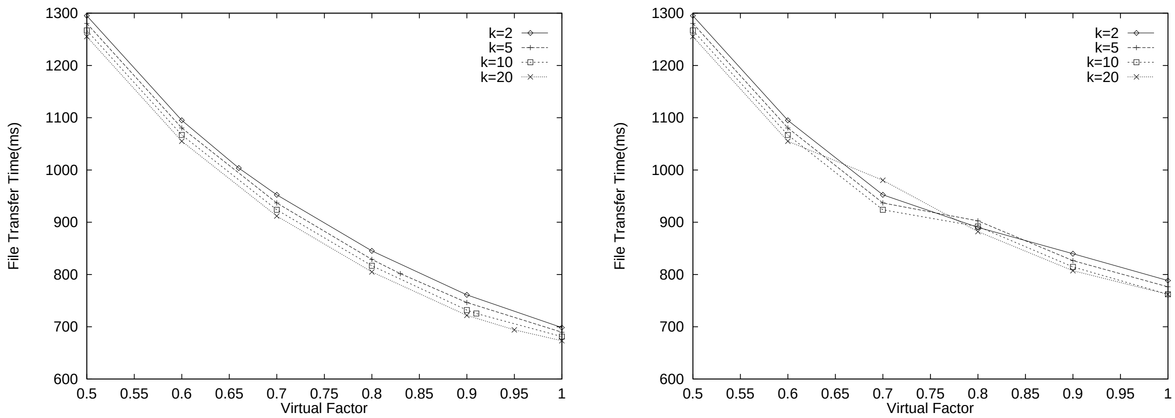



Figure 5.26: File Transfer Time for Connection 1 in Scenario 3 (Migrating Bottlenecks) With Delayed Resolution. (Left: Buffer Size = 440; Right: Buffer Size = 44)

File transfer times decrease as $v$ increases. The file transfer times for all values of $k$ are not substantially different from each other. The file transfer times for the two cases are identical for small values of $v$, (i.e., up to 0.7). Beyond this point, the differences become more pronounced. For $k = 2$ and $v = 1.0$, the file transfer time is 698 ms for Case 1 and 788 ms for Case 2.

Both cases perform better with the delayed resolution strategy. Without delayed resolution, the file transfer time for $k = 2$ and $v = 1.0$ is 833 ms (see Figure 5.15).

**Cell Delay**

Figure 5.27 show the mean cell delay for connection 1 in Cases 1 and 2 in Scenario 3.

As in the earlier scenarios, the mean cell delay increases for increasing values of $v$. The mean cell delay is much worse for Case 1. For $k = 2$ and for $v = 1.0$, the mean cell delay is 10.12 ms for Case 1, while it is only 2.47 ms for Case 2. Non-adoption of the delayed resolution strategy, again, results in the best performance in terms of mean cell delay. For $k = 2$ and $v = 1.0$, the mean cell delay is only 0.41 ms without

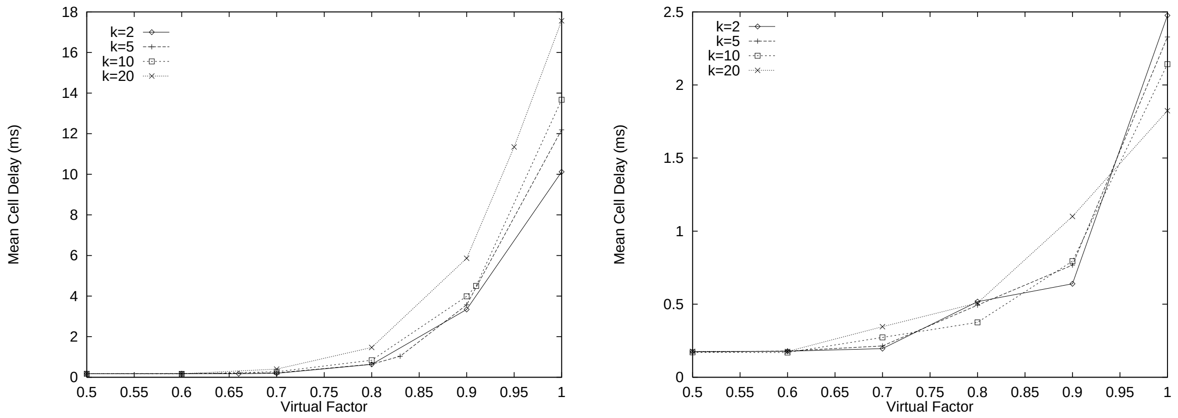



Figure 5.27: Mean Cell Delay for Connection 1 in Scenario 3 (Migrating Bottlenecks) With Delayed Resolution. (Left: Buffer Size = 440; Right: Buffer Size = 44)

the delayed resolution strategy (see Figure 5.16).

**<u>Green Cell Loss Probability</u>**

Figure 5.28 shows the green cell loss probabilities for connection 1 for Cases 1 and 2 in Scenario 3.

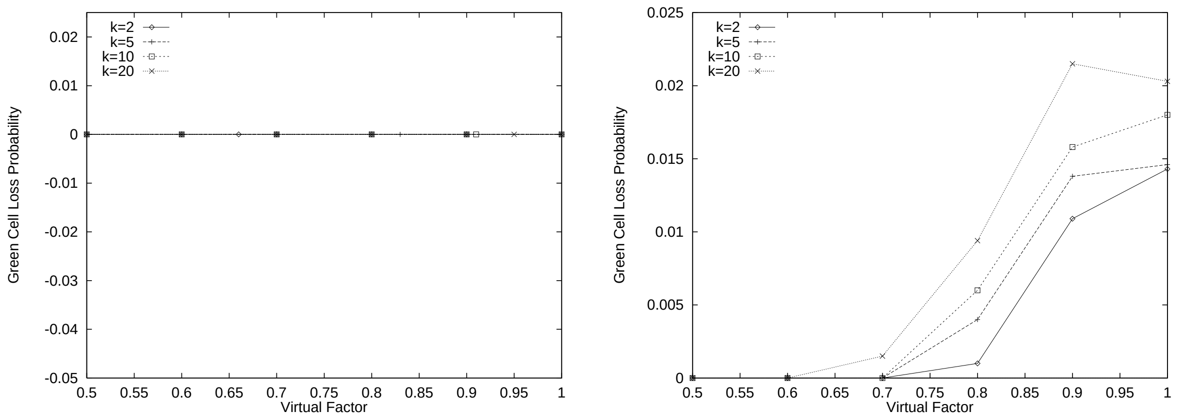



Figure 5.28: Green Cell Loss Probability for Connection 1 in Scenario 3 (Migrating Bottlenecks) With Delayed Resolution. (Left: Buffer Size = 440; Right: Buffer Size = 44)

For Case 1, the green cell loss probability was observed to be 0 for all values of $k$ and $v$. For Case 2, increasing the value of $v$ results in the green cell loss probability increasing, for a given value of $k$.

Increased buffering clearly results in reducing the green cell loss probability for a given value of $k$ and $v$. However, the difference in the amount of green cell loss encountered between Cases 1 and 2 is again not very large considering the difference in the sizes of the buffer employed in each case. For Case 1, the green cell loss probability is 0 for all values of $k$ and $v$. For $k = 2$ and $v = 1.0$, the green cell loss probability is 0.01 for Case 2. Both cases perform better with the delayed resolution strategy. The green cell loss probability for $k = 2$ and $v = 1.0$ is 0.08 without the delayed resolution strategy. As in Scenarios 1 and 2, increasing the value of $k$ results in increasing the green cell loss probability for a given value of $v$.

### 5.4.4 Summary

This subsection summarises some of the key results for the virtual loss-load model for all the three scenarios considered, in the presence of the delayed resolution strategy.

- As in the scenarios without the delayed resolution strategy, the virtual factor $v$ controls the amount of cell loss and delay encountered by a sender to a considerable extent. Increasing the value of $v$ results in greater cell loss and delay.

- In the scenarios considered, the effect of $k$ is similar with or without the delayed resolution strategy. Low values of $k$ seem more able to reduce overall loss in the network.

- File transfer times again decrease with increasing $v$, with the delayed resolution strategy. The delayed resolution strategy reduces the file transfer time as compared to use without the strategy. Use of the buffers helps in reducing the amount of cell loss and therefore cuts down the time spent retransmitting cells.

- The use of the delayed resolution strategy reduces cell loss considerably, but the use of large buffer sizes in the delayed resolution strategy results in large mean cell delays. However, the decrease in the cell loss probability is not as substantial as expected when the buffer sizes are large. The results show that even a small buffer can provide substantial reduction in mean cell delay without increasing the cell loss appreciably.

## 5.5 Additional Experiments

While previous sections considered the performance of the virtual loss-load model in dynamic network conditions, the scenarios in this section measure other aspects of congestion control performance.

### 5.5.1 Scenario 4: Transfer Time for Short Files

Transfer time is a critical performance measure for small file transfers. The purpose of this experiment is to study the performance of different congestion control strategies based on the way senders adjust their rate of operation as they commence a file transfer, and to determine what effect this has on file transfer performance.

The configuration shown in Figure 5.29 is used to evaluate the congestion control mechanisms for short file transfers of 4, 20 and 200 packets [19].

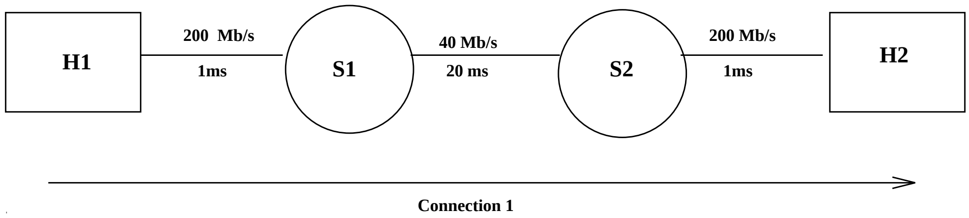



Figure 5.29: Network Configuration used for Scenario 4

As can be seen, there is no cross-traffic in this scheme, since it is aimed at measuring only rate changes as senders adapt to the network. The only performance metric used in this experiment is the file transfer time. The optimal file transfer time for files of sizes 4, 20 and 200 packets is 44.4 ms, 46 ms and 64 ms respectively.

Figure 5.30 shows the file transfer time for connection 1 for various values of $k$ and $v$ for the transfer of 20 packets.

The results presented are from experiments without the delayed resolution strategy, but experiments with delayed resolution produced identical results. The transfer of files of sizes 4 and 200 packets produced similar results which are not shown.

In this scenario, there are no changes in the rate of operation of the connection since there is no cross-traffic. The file transfer time therefore approaches the theo-

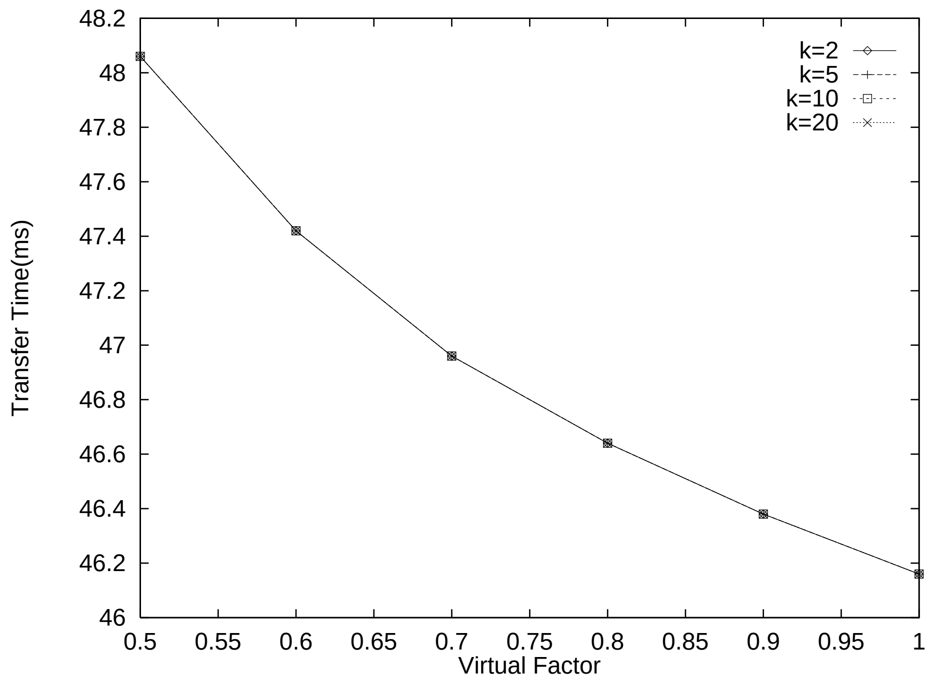



Figure 5.30: File Transfer Time for Connection 1 in Scenario 4 (Transfer Time for Short File of Size 20 Packets) Without Delayed Resolution

retical optimum value, as the value $v$ is increased in the virtual loss-load model. The factor $k$ does not play any role, since the maximum bandwidth possible for a given value of $v$ is made available to a connection regardless of the value of $k$. Thus, the lines corresponding to different values of $k$ coincide in the graph.

### 5.5.2 Scenario 5: Fairness

Earlier studies of TCP's performance have shown that connections operating over a longer feedback delay path get a smaller share of the bandwidth at the bottleneck as a result of the reactive congestion control policy employed ('slow start'). This "unfairness" results in increased file transfer time for traffic streams operating over longer delay paths.

The purpose of this final benchmark scenario is to judge the fairness of a given congestion control strategy by comparing the file transfer times of two different control connections, operating over different feedback delay paths. A scheme is judged to be "fair" if the file transfer times are comparable to one another.

Figure 5.31 shows the configuration used in evaluating fairness for the various proposed schemes. The two control connections, 1 and 2, begin at time 0 and each have 2 Mbytes of data to send. The cross traffic stream, connection 3, begins after 100 ms while connections 4 through 12 begin every 44 ms later. All cross traffic streams send a file of size 500 Kbytes. As can be seen, connection 1 has a round trip of only 44 ms, while connection 2 has a feedback delay of 82 ms. The optimal file transfer time is 1844 ms for connection 1 in this scenario, while it is 1882 ms for connection 2.

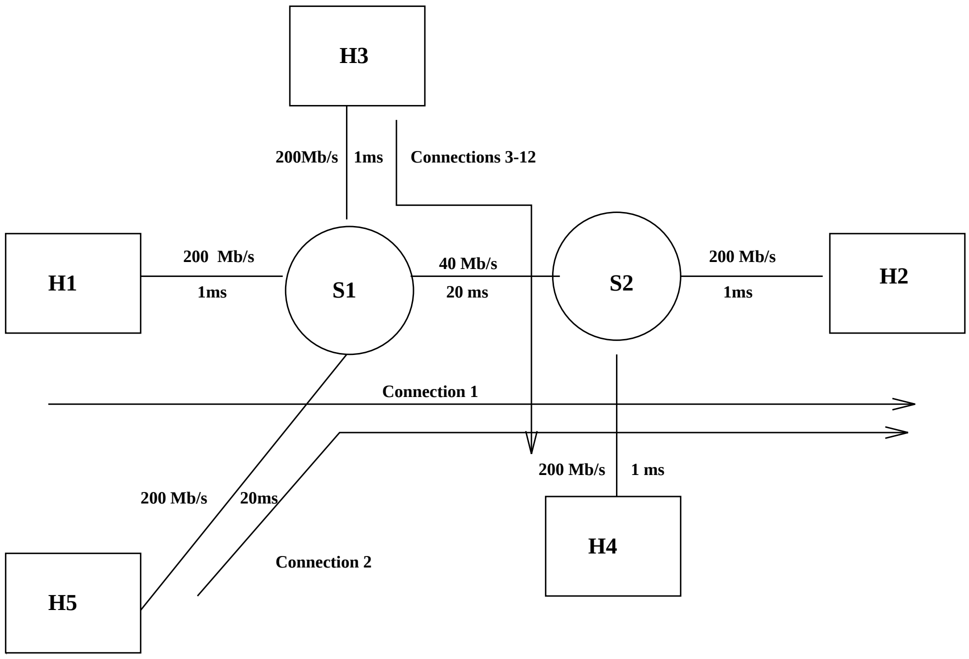



Figure 5.31: Network Configuration used for Scenario 5

Figure 5.32 shows the file transfer times of connections 1 and 2 for different values of $v$, for a value of $k = 2$, without and with the delayed resolution strategy. Values of $k$ other than 2 produced similar results, and so are not shown. The main purpose of the experiment was to compare the fairness of the virtual loss-load model for different values of $k$, $v$ and the delayed resolution strategy.

From Figure 5.32, connection 1 is observed to perform better than connection 2 for all values of $v$, both without and with the delayed resolution strategy. The difference in file transfer time for the two connections is mostly constant for all values

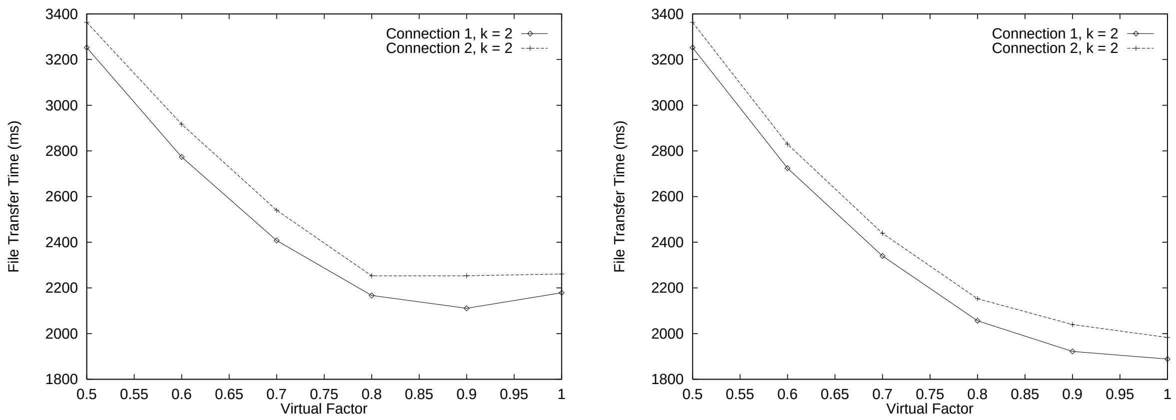



Figure 5.32: File Transfer Time for Connections 1 and 2 in Scenario 4 (Fairness). (Left: Without Delayed Resolution, Right: With Delayed Resolution, Buffer Size = 440)

of $v$ in both cases. For example, when the delayed resolution strategy is used, the file transfer time for connection 1 is 1922 ms for connection 1 at $v = 0.9$, while it is 2039 ms for connection 2. At $v = 1.0$, the file transfer time decreases to 1888 ms for connection 1, while it falls to 1982 ms for connection 2. However, since the file transfer times are comparable to one another, the scheme appears to be reasonably fair. It appears that the factors $k$, $v$ and the delayed resolution strategy do not play a major role in the fairness of the virtual loss-load model. Rather, the feedback delay is the factor contributing to the unfairness.

## 5.6 Comparison with other Congestion Control Strategies

Performance results are available for the following congestion control strategies in the test suite described by Kanakia *et al.*: the TCP slow-start scheme (tcp) implemented in the 4.3 BSD Tahoe operating system [16], the DECbit scheme [33], the Packet Pair scheme (pp) [20], the explicit end-to-end feedback scheme (e2e-ef) and the hop-by-hop (hbh) congestion control scheme described in [29].

This section compares the performance of the virtual loss-load model with that of the original loss-load model and also with those of the above reactive congestion control strategies for each of the benchmark scenarios described earlier. Although a range of combinations of $k$, $v$, and the delayed resolution strategy could be considered for the virtual loss-load model in making this comparison, only four choices are presented. This includes a value of $k = 2$ (since this provided the best overall performance of the virtual loss-load model), values of $v$ at 1.0 and 0.9 (so as to indicate the performance of the original loss-load model and the effect of the virtual factor in the virtual loss-load model), and the absence/presence of the delayed resolution strategy. A switch buffer size of 44 is used when making the comparison using the delayed resolution strategy. This yields four combinations of the factors of the virtual loss-load model:

- The first combination of the virtual loss-load model, with $k = 2$, $v = 1.0$, and without delayed resolution, provides performance characteristics representative of the original loss-load model.

- The second combination, with $k = 2$, $v = 1.0$, and with the delayed resolution strategy (buffer size = 44), allows for comparing the virtual loss-load model with the original loss-load model in the presence of the delayed resolution strategy alone.

- Using $k = 2$, $v = 0.9$, and without the delayed resolution strategy, the third combination provides information on the impact of the virtual factor in the virtual loss-load model.

- Finally, the fourth combination, with $k = 2$, $v = 0.9$, and the delayed resolution strategy (buffer size = 44), allows for comparison of the virtual loss-load model with the original loss-load when using both the delayed resolution strategy, as well as a lower value of the virtual factor.

### 5.6.1 Comparison for Scenario 1: Slow Changes in Cross Traffic

Table 5.2 presents the performance of the virtual loss-load model in Scenario 1. Table 5.3 summarises the performance of other congestion control strategies, as reported in [19].

Table 5.2: Virtual Loss-Load Model Performance for Scenario 1 (Slow Changes in Cross Traffic)

| Scheme | Conn 1 Transfer Time (ms) | Mean Cell Delay (ms) | Std. Dev. | Green C.L.P. |
|---|---|---|---|---|
| $k = 2$, $v = 1.0$ without delayed resolution | 1758 | 0.41 | 0.09 | 0.15 |
| $k = 2$, $v = 1.0$ with delayed resolution, buffer size = 44 | 1553 | 2.82 | 1.87 | 0.03 |
| $k = 2$, $v = 0.9$ without delayed resolution | 1737 | 0.45 | 0.48 | 0.07 |
| $k = 2$, $v = 0.9$ with delayed resolution, buffer size = 44 | 1650 | 2.06 | 0.90 | 0.01 |

Table 5.3: Congestion Control Scheme Performance for Scenario 1 (Slow Changes in Cross Traffic) [19]

| Scheme | Conn 1 Transfer Time (ms) | Mean Cell Delay (ms) | Std. Dev. |
|---|---|---|---|
| Optimal | 1444 | 0 | 0 |
| hbh | 1534 | 0.99 | 0.41 |
| pp | 1569 | 0.31 | 0.17 |
| e2e-ef | 1621 | 0.85 | 0.42 |
| tcp | 4430 | na | na |
| decbit | 5566 | na | na |

From Table 5.2, it may be seen that the original loss-load model shows the worst performance among the four virtual loss-load model combinations both in terms of file transfer time and in terms of the green cell loss probability. The original loss-load model, however, shows good mean cell delay characteristics. The addition of the delayed resolution strategy to the original loss-load model improves file transfer time, while decreasing the green cell loss probability considerably. By decreasing the value of the virtual factor to 0.9 from a value of 1.0 in the original loss-load model, the green cell loss probability can be reduced to a large extent, while simultaneously improving the file transfer time. By incorporating both a low value of the virtual factor, and the delayed resolution strategy, the green cell loss probability can be reduced to its smallest value, while improving the file transfer time performance of the original loss-load model, though at the expense of increasing the mean cell delay moderately.

The virtual loss-load model performs far better than the tcp scheme and the decbit scheme in terms of file transfer time, for all four combinations used for the comparison. All combinations of the virtual loss-load model are comparable in performance to those of the other reactive congestion control strategies. The second combination of the virtual loss-load model (using $k = 2$, $v = 1.0$, and the delayed resolution strategy), shows the best file transfer time performance among the four combinations. Its file transfer time performance is better than all the other reactive congestion schemes with the exception of the hbh scheme.

### 5.6.2 Comparison for Scenario 2: Sudden Jumps in Cross Traffic

Table 5.4 presents the performance of the virtual loss-load model in Scenario 2. Table 5.5 enumerates the performance of other reactive congestion control strategies for the same scenario [19].

The performance results for the four virtual loss-load model combinations in Table 5.4 are similar to those of Scenario 1 in Table 5.2. The original loss-load model once again shows the worst file transfer time performance among the four combinations. The delayed resolution strategy reduces the green cell loss probability considerably,

Table 5.4: Virtual Loss-Load Model Performance for Scenario 2 (Sudden Jumps in Cross Traffic)

| Scheme | Conn 1 Transfer Time (ms) | Mean Cell Delay (ms) | Std. Dev. | Green C.L.P. |
|---|---|---|---|---|
| $k = 2$, $v = 1.0$ without delayed resolution | 1707 | 0.77 | 0.89 | 0.15 |
| $k = 2$, $v = 1.0$ with delayed resolution, buffer size = 44 | 1632 | 2.52 | 1.94 | 0.03 |
| $k = 2$, $v = 0.9$ without delayed resolution | 1684 | 0.40 | 0.35 | 0.10 |
| $k = 2$, $v = 0.9$ with delayed resolution, buffer size = 44 | 1653 | 2.28 | 2.00 | 0.05 |

Table 5.5: Congestion Control Scheme Performance for Scenario 2 (Sudden Jumps in Cross Traffic) [19]

| Scheme | Conn 1 Transfer Time (ms) | Mean Cell Delay (ms) | Std. Dev. |
|---|---|---|---|
| Optimal | 1444 | 0 | 0 |
| hbh | 1543 | 1.09 | 0.51 |
| pp | 1594 | 0.36 | 0.23 |
| e2e-ef | 1605 | 1.01 | 0.49 |
| tcp | 3775 | na | na |
| decbit | 5568 | na | na |

while simultaneously improving file transfer time performance. By using a value of $v = 0.9$, a reduction in the file transfer time and mean cell delay can be achieved, while decreasing the green cell loss probability to a large extent. Finally, by reducing $v$ and using the delayed resolution strategy, the virtual loss-load model can be made to improve on the performance of the original loss-load model both in terms of file transfer time, as well as in terms of the green cell loss probability, though at the expense of increasing the mean cell delay marginally.

Again, all four combinations of the virtual loss-load model show better performance than the tcp and decbit schemes. All four combinations show comparable performance to the other reactive congestion control strategies. The virtual loss-load model performs slightly worse than the hbh, pp, and e2e-ef schemes in this scenario.

### 5.6.3 Comparison for Scenario 3: Migrating Bottlenecks

Table 5.6 summarises the performance of the four combinations of the virtual loss-load model in Scenario 3, while Table 5.7 presents the performance of other congestion control strategies [19] for Scenario 3.

In contrast to the earlier two scenarios, the original loss-load model does not perform worse than all the other virtual loss-load model combinations in terms of file transfer time performance. However, it shows the worst green cell loss probability when compared to the other combinations. Other results for the combinations are similar to Scenarios 1 and 2.

All four combinations of the virtual loss-load model perform better than the tcp and decbit schemes, as in the earlier scenarios. However, all the other reactive congestion control strategies (hbh, pp, e2e-ef) show moderately better file transfer time performance when compared to the four combinations of the virtual loss-load model.

Table 5.6: Virtual Loss-Load Model Performance for Scenario 3 (Migrating Bottlenecks)

| Scheme | Conn 1 Transfer Time (ms) | Mean Cell Delay (ms) | Std. Dev. | Green C.L.P. |
|---|---|---|---|---|
| $k = 2$, $v = 1.0$ without delayed resolution | 833 | 0.41 | 0.50 | 0.07 |
| $k = 2$, $v = 1.0$ with delayed resolution, buffer size = 44 | 788 | 2.47 | 2.69 | 0.01 |
| $k = 2$, $v = 0.9$ without delayed resolution | 853 | 0.32 | 0.42 | 0.04 |
| $k = 2$, $v = 0.9$ with delayed resolution, buffer size = 44 | 839 | 0.64 | 1.08 | 0.01 |

Table 5.7: Congestion Control Scheme Performance for Scenario 3 (Migrating Bottlenecks) [19]

| Scheme | Conn 1 Transfer Time (ms) | Mean Cell Delay (ms) | Std. Dev. |
|---|---|---|---|
| Optimal | 644 | 0 | 0 |
| hbh | 714 | 0.49 | 0.07 |
| pp | 771 | 0.082 | 0.041 |
| e2e-ef | 769 | 0.37 | 0.07 |
| tcp | 3237 | na | na |
| decbit | 5846 | na | na |

Table 5.8: Virtual Loss-Load Model Performance for Scenario 4 (Transfer Times for Short Files)

| Scheme | Trans Time (ms) 4 pkts | Trans Time (ms) 20 pkts | Trans Time (ms) 200 pkts |
|---|---|---|---|
| $k = 2$, $v = 1.0$ without delayed resolution | 44 | 46 | 64 |
| $k = 2$, $v = 1.0$ with delayed resolution, buffer size = 44 | 44 | 46 | 64 |
| $k = 2$, $v = 0.9$ without delayed resolution | 44 | 46 | 64 |
| $k = 2$, $v = 0.9$ with delayed resolution, buffer size = 44 | 44 | 46 | 64 |

### 5.6.4 Comparison for Scenario 4: Transfer Times for Short Files

Table 5.8 presents the performance of the virtual loss-load model in Scenario 4, while Table 5.9 summarises the performance of other congestion control strategies [19].

The virtual loss-load model provides optimal performances for all four choices in this scenario. For this scenario, all four combinations of the virtual loss-load model perform better than the other congestion control strategies.

Table 5.9: Congestion Control Scheme Performance for Scenario 4 (Transfer Times for Short Files) [19]

| Scheme | Trans Time (ms) 4 pkts | Trans Time (ms) 20 pkts | Trans Time (ms) 200 pkts |
|---|---|---|---|
| Optimal | 44.4 | 46 | 64 |
| hbh | 53 | 73 | 122 |
| pp | 84 | 92 | 146 |
| e2e-ef | 47 | 63 | 150 |
| tcp | 120 | 200 | 327 |
| decbit | 120 | 321 | 1203 |

### 5.6.5 Comparison for Scenario 5: Fairness

Table 5.10 presents the performance of the four combinations of the virtual loss-load model in Scenario 5. Table 5.11 presents the performance of other congestion control strategies [19].

Table 5.10: Virtual Loss-Load Model Performance for Scenario 5 (Fairness)

| Scheme | Transfer Time (ms) Conn. 1 | Transfer Time (ms) Conn. 2 |
|---|---|---|
| $k = 2$, $v = 1.0$ without delayed resolution | 2178 | 2260 |
| $k = 2$, $v = 1.0$ with delayed resolution, buffer size = 44 | 1888 | 1983 |
| $k = 2$, $v = 0.9$ without delayed resolution | 2111 | 2253 |
| $k = 2$, $v = 0.9$ with delayed resolution, buffer size = 44 | 1922 | 1983 |

Table 5.11: Congestion Control Scheme Performance for Scenario 5 (Fairness) [19]

| Scheme | Transfer Time (ms) Conn. 1 | Transfer Time (ms) Conn. 2 |
|---|---|---|
| Optimal | 1844 | 1882 |
| hbh | 1859 | 1927 |
| pp | 1859 | 2022 |
| e2e-ef | 1877 | 2004 |
| tcp | 4030 | 8643 |
| decbit | 6316 | 12010 |

Among the four combinations of the virtual loss-load model, the delayed resolution strategy at $v = 1.0$ and $v = 0.9$ provides the least divergence from the optimal file transfer times for control connections 1 and 2. The difference in file transfer times between the two connections is not large, indicating good fairness properties of the virtual loss-load model. The absence of the delayed resolution strategy in the virtual loss-load model causes a large divergence from the optimal file transfer times for both connections, although the relative difference between the file transfer times of the two connections is not large.

## 5.7 Summary

This chapter examined the performance of the virtual loss-load model under various network scenarios, and compared it to other reactive congestion control strategies. Feedback delays and the effect of transient network conditions produce a major impact on the performance of the virtual loss-load model, and cause its behaviour to differ considerably from that of the simple two-sender scenario considered in Chapter 4.

For the first three benchmark scenarios, small values of $k$, together with the delayed resolution strategy with few buffers, produced the best overall performance. Small values of $v$, in the region of 0.5 to 0.6, gave good results in terms of cell loss probability and mean cell delay. Large values of $v$, in the range 0.9 to 1.0, on the other hand, showed reduced file transfer time at the expense of increasing cell loss and mean cell delay. Values of $v$ in the range 0.7 to 0.9 gave good results in terms of cell loss and mean cell delay, without degrading file transfer time performance substantially.

For the final two benchmark scenarios, the virtual loss-load model performed optimally for short file transfers, and showed properties of being a reasonably fair congestion control algorithm.

The virtual loss-load model provides performance comparable to that of other reactive congestion control strategies, depending on how the parameters affecting its performance are tuned. Overall, the virtual loss-load model performs comparably to other congestion control strategies when $k$ is small (e.g., $k = 2$) and the delayed resolution strategy with limited buffers is in effect.

Chapter 6 provides a more complete overview of the experiments that were conducted and the experimental results that were obtained.

# Chapter 6

# Summary and Conclusions

This chapter summarises the objectives of the thesis, the experiments conducted using the virtual loss-load model, and the results and conclusions arising from those experiments. Several areas for future research are identified.

## 6.1 Thesis Summary

The virtual loss-load model is a hybrid congestion control approach, combining the leaky bucket traffic shaper (a preventive congestion control scheme) with the loss-load congestion control approach (a reactive congestion control scheme). The main motive for combining these two approaches is to exploit the advantages, and to mitigate the disadvantages, that each scheme possesses in isolation.

The main objective of this thesis was to evaluate the performance of the virtual loss-load model. The simulation experiments conducted in this thesis were designed to answer the following questions:

- How do the factors $k$, $v$ and the use of the delayed resolution strategy affect the performance of the virtual loss-load model?
- To what degree does the virtual loss-load model improve on the originally proposed loss-load model?

- How does the performance of the original loss-load model and the virtual loss-load model compare with the performance of other congestion control strategies described in the literature?

To answer these questions, simulation experiments were conducted based on a set of benchmark scenarios proposed by Kanakia, Keshav and Mishra [19]. The set of scenarios attempts to capture dynamic network conditions characteristic of broadband networks and judge the performance of congestion control algorithms under transient network conditions. The workload in the experiments was based on the file transfer paradigm: a very general model of network workload that captures the characteristics of a diverse range of applications. The benchmark scenarios also allow for comparing the performance of the virtual loss-load model with the performance results published for other reactive congestion control strategies: the TCP slow-start scheme implemented in the BSD Tahoe operating system, the DECbit scheme, the Packet-Pair scheme, and the Explicit End-to-End Feedback and the Hop-by-Hop scheme.

Five benchmark scenarios were considered. The first three are used to evaluate the performance of a congestion control scheme in reaction to (i) slow changes in cross-traffic, (ii) sudden changes in cross-traffic, and (iii) migrating bottlenecks in the network. The fourth is used to evaluate performance for short file transfers, and the fifth is used to judge fairness.

Three factors control the performance of the virtual loss-load model. These are the factor $k$, the factor $v$, and the presence/absence of the delayed resolution strategy. For each experiment, combinations of various levels of the above factors were considered, yielding a total of 72 simulation experiments for each benchmark scenario.

The following summarises some of the key observations from the experiments conducted for the first three benchmark scenarios:

- Low values of $k$ are more effective in reducing file transfer time and cell loss in the network. This is due to the dynamics of the loss-load congestion control algorithm where low values of $k$ were found to reduce overload and oscillation in the presence of transient network conditions.

- The virtual factor $v$ controls to a large extent the amount of cell loss and delay encountered by a sender. Low values of $v$ result in lower cell loss and cell delay, but do not effectively utilise the available bandwidth in the network. High values of $v$, on the other hand, result in considerable losses for a sender. Values of $v$ between 0.7 and 0.9 provide reasonable cell delay and cell loss, along with reasonably good performance in terms of file transfer time. The file transfer time almost always decreases with increasing $v$. The implication of this result is that even though higher cell losses are encountered for increasing $v$, it may still be worthwhile to operate at larger values of $v$ if reducing file transfer time is the only criterion of importance to the sender.

- The presence of the delayed resolution strategy helps in reducing the cell loss considerably when compared to use without the strategy. The results showed that the use of even a small buffer with the delayed resolution strategy could reduce mean cell delay substantially, while not increasing file transfer time and cell loss appreciably.

The following are the results for the final two benchmark scenarios that were considered:

- The virtual loss-load model performed well for short file transfers, with its performance approaching the theoretical optimum.

- The virtual loss-load model is a reasonably fair congestion control strategy. The factors $k$, $v$ and the delayed resolution strategy do not appear to play a major role in affecting the fairness of the virtual loss-load model. The feedback delay seems to be the major factor affecting fairness of the virtual loss-load model.

## 6.2 Conclusions

Based on the experiments that were conducted, the following conclusions may be drawn:

- *Effect of factors in the virtual loss-load model*: Low values of $k$ provided the best overall performance for the benchmark scenarios considered in this thesis. Larger values of $v$ improve file transfer time, but do so at the expense of increased cell loss and delay. Values of $v$ in the region 0.7 to 0.9 provide low cell loss and delay, with comparable file transfer time performance. Use of the delayed resolution strategy with few buffers was found to improve performance considerably in terms of reducing file transfer time, cell loss and mean cell delay.

- *Comparison with the loss-load model*: The virtual loss-load model improves on the original loss-load model by substantially reducing cell loss. The virtual loss-load model differs from the original loss-load model primarily through the incorporation of the virtual factor $v$, and the use of the delayed resolution strategy. Use of the delayed resolution strategy can reduce cell loss considerably, simultaneously improving file transfer time. The factor $v$ is another important controlling parameter in the virtual loss-load model. By setting $v$ to small values, cell loss can be decreased considerably.

- *Comparison with other congestion control strategies*: The performance of both the original loss-load model and the virtual loss-load model is comparable to that of several other reactive congestion control strategies. The virtual loss-load model using the delayed resolution strategy (with a small number of buffers) showed the most promising overall performance, competitive with the performance of the other reactive congestion control algorithms considered in this thesis.

## 6.3 Future Work

The performance of the congestion control strategies in this thesis was considered in the framework of broadband networks. One example of an emerging broadband network is ATM networks. ATM networks will probably require more than one type of congestion control strategy, as it seems unlikely that a single congestion control strategy would be sufficient to satisfy the QOS requirements of all types of traffic. In the ATM environment, for example, best-effort traffic is distinct from other types of traffic. Best-effort traffic, known as Available Bit Rate (ABR) traffic in the ATM Forum, represents traffic that does not require stringent real-time requirements. The virtual loss-load model is intended to satisfy the QOS requirements of traffic similar to ABR traffic. An important research question that needs to be answered is how feasible, and what changes, if any, would be necessary to incorporate the virtual loss-load model in an ATM environment? Can the virtual loss-load model coexist effectively with other congestion control strategies in an ATM environment?

The experiments that were conducted for this thesis show the importance of characterising and understanding the dynamics of congestion control strategies in transient network conditions, with dynamic cross traffic flows and large feedback delays. Although these conditions are aimed specifically at modelling future broadband networks, transient traffic conditions may be a valid model even in the existing LAN and WAN environment. The performance of many current congestion control strategies, on the other hand, have been evaluated under simple traffic scenarios and/or under stable operating conditions. Further research needs to be carried out to see if these performance studies are realistic enough, and whether the performance results of these strategies provide a true characterisation of the effectiveness of these congestion control approaches. Also, more research needs to be carried out to design more and better benchmark scenarios to sufficiently capture the performance of different congestion control strategies.

# Bibliography


[1] H. Armbruster and K. Wimner, "Applications using ATM Networks: High Performance Computing, High Capacity Storage and High Speed Communications", *IEEE Journal on Selected Areas in Communication*, Vol. 10, No. 9, pp. 1382-1402, December 1992.

[2] K. Bala, I. Cidon, and K. Sohraby, "Congestion Control for High Speed Packet Switched Networks", *Proceedings of INFOCOMM 1990*, San Francisco, pp. 520-525, June 1990.

[3] A. Bhargava, *Integrated Broadband Networks*, Artech House, Boston, 1991.

[4] M. Butto, E. Cavallero, and A. Tonietti, "Effectiveness of the Leaky Bucket Policy Mechanism in ATM Networks", *IEEE Journal on Selected Areas in Communication*, Vol. 9, No. 3, pp. 335-342, April 1991.

[5] I. Cidon, I. Gopal, and R. Guerin, "Bandwidth Management and Congestion Control in plaNET", *IEEE Communications Magazine*, Vol. 30, No. 10, pp. 54-64, October 1991.

[6] L. Cuthbert and J. Sapanel, *ATM: The Broadband Telecommunications Solution*, Institution of Electrical Engineers, London, 1993.

[7] M. DePrycker, *Asynchronous Transfer Mode*, Ellis Horwood, London, 1993.

[8] L. Dittman, S. Jacobsen, and M. Moth, "Flow Enforcement Algorithms for ATM Networks", *IEEE Journal on Selected Areas in Communication*, Vol. 9, No. 3, pp. 343-350, April 1991.

[9] B. Doshi and H. Nguyen, "Congestion Control in ISDN Frame Relay Networks", *AT & T Technical Journal*, Vol 67, pp. 35-46, November-December 1988.

[10] A. Eckberg, D. Luan and D. Lucantoni, "Meeting the Challenge - Congestion Control and Flow Control Strategies for Broadband Information Transport", *Proceedings of IEEE GLOBECOM 1989*, pp. 49.3.1-49.3.5, November 1989.

[11] A. Eckberg, "B-ISDN/ATM Traffic Control", *IEEE Network*, Vol. 6, No. 5, pp. 28-37, September 1992.

[12] H. Gilbert, O. Aboul-Magd, and V. Phung, "Developing a Cohesive Traffic Management Strategy for ATM Networks", *IEEE Communications Magazine*, Vol. 30, No. 10, pp. 36-45, October 1991.

[13] I. Habib and T. Saadawi, "Controlling Flow and Avoiding Congestion in Broadband Networks", *IEEE Communications Magazine*, Vol. 30, No. 10, pp. 46-53, October 1991.

[14] D. Hong and T. Suda, "Congestion Control and Prevention in ATM Networks", *IEEE Network Magazine*, Vol. 5, No. 4, pp. 10-15, July 1991.

[15] T. Hoshi, K. Mori, T. Takahashi, and T. Ishizaki, "B-ISDN Multimedia Communications and Collaboration Platform using Advanced Video Workstations to Support Cooperative Work", *IEEE Journal on Selected Areas in Communication*, Vol. 10, No. 9, pp. 1402-1412, December 1992.

[16] V. Jacobson, "Congestion Avoidance and Control", *Proceedings of ACM SIGCOMM 1988*, pp. 158-181, August 1988.

[17] R. Jain, "Congestion Control in Computer Networks: Issues and Trends", *IEEE Network Magazine*, pp. 24-30, May 1990.

[18] R. Jain, "Myths about Congestion Control in High Speed Networks", DEC Technical Report: DEC TR-726, January 1991.

[19] H. Kanakia, S. Keshav and P. Mishra, "A Comparison of Reactive Host Based Flow Control Schemes in Reservationless Networks", Draft Report, AT & T Bell Laboratories, August 1993.

[20] S. Keshav, "A Control Theoretic Approach to Flow Control", *Proceedings of ACM SIGCOMM 1991*, pp. 3-15, September 1991.

[21] G. Kessley, "IEEE 802.6 MAN", *LAN Magazine*, pp. 102-116, April 1990.

[22] L. Kleinrock, "The Latency/Bandwidth Tradeoff in Gigabit Networks", *IEEE Communications Magazine*, Vol. 30, No. 4, pp. 36-40, April 1992.

[23] H.T. Kung, "Gigabit Local Area Networks: A Systems Perspective", *IEEE Communications Magazine*, Vol. 30, No. 4, pp. 79-89, April 1992.

[24] H. Liu, "Traffic Shaping for Congestion Control in High Speed ATM Networks", Research Report 92-6 (M.Sc. Thesis), Department of Computational Science, University of Saskatchewan, August 1992.

[25] W. Liu, I. Cidon, I. Gopal, and M. Sidi, "Congestion Control Through Input Rate Regulation", *Proceedings of IEEE GLOBECOM '89*, pp. 1764-1768, November 1989.

[26] S. Lowe, "Plugging into Frame Relay for Speed and Flexibility", *Data Communications Magazine*, pp. 54-62, April 1990.

[27] E. Mier, "New Signs of Life for Packet Switching", *Data Communications Magazine*, pp. 90-106, December 1989.

[28] S. Minzer, "Broadband ISDN and Asynchronous Transfer Mode (ATM)", *IEEE Communications Magazine*, Vol. 27, No. 9, pp. 17-24, September, 1989.

[29] P. Mishra and H. Kanakia, "A Hop-by-Hop Rate Based Congestion Control Scheme", *Proceedings of ACM SIGCOMM 1992*, pp. 112-123, August 1992.

[30] H. Ohnishi, T. Okada, and T. Noguchi, "Flow Control Schemes and Delay/Loss Tradeoffs in ATM Networks", *IEEE Journal on Selected Areas in Communication*, pp. 1606-1616, December 1988.

[31] C. Partridge, *Gigabit Networking*, Addison-Wesley, Reading MA, 1993.

[32] D. Petr and V. Frost, "Optimal Packet Discarding: An ATM-Oriented Analysis Model and Initial Results", *Proceedings of INFOCOMM 1990*, San Francisco, pp. 537-542, June 1990.

[33] K.K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks", *ACM Transactions on Computer Systems*, Vol. 8, No. 2, pp. 158-181, 1990.

[34] E. Rathgeb, "Modeling and Performance Comparison of Policing Mechanisms for ATM Networks", *IEEE Journal on Selected Areas in Communication*, Vol. 9, No. 3, pp. 325-332, April 1991.

[35] J. Turner, "New Directions in Communications (or Which Way to the Information Age?)", *IEEE Communications Magazine*, Vol. 24, No. 10, pp. 8-15, October 1986.

[36] J. Turner, "Design of an Integrated Services Packet Network", *Proceedings of ACM SIGCOMM 1985*, pp. 124-133, 1985.

[37] J. Turner, "Managing Bandwidth in ATM Networks with Bursty Traffic", *IEEE Network*, Vol. 6, No. 5, pp. 50-58, September 1992.

[38] C. Williamson and D. Cheriton, "Loss-Load Curves: Support for Rate-Based Congestion Control in Datagram Networks", *Proceedings of ACM SIGCOMM '91*, pp. 17-28, 1991.

[39] C. Williamson, "Optimizing File Transfer Response Using the Loss Load Curve Congestion Control Mechanism", *Proceedings of ACM SIGCOMM '93*, pp. 117-126, August 1993.

[40] G. Woodruff, R. Rogers and P. Richards, "A Congestion Control Framework for High-Speed Integrated Packetized Transport", *Proceedings of IEEE GLOBE-COM 1988*, pp. 203-207, November 1988.

[41] G. Woodruff, "Towards Standardization of ATM Traffic Control for B-ISDN", Symposium on Traffic Control Architectures for Integrated Networks, Columbia University, New York, April 1989.

[42] L. Zhang, "Some Thoughts on the Packet Network Architecture", *ACM Computer Communication Review*, Vol. 17, No. 1, pp. 3-15, 1987.

[43] "Broadband Aspects of ISDN", CCITT Recommendation I.121, Geneva, Switzerland, 1989.

[44] "Generic System Requirements in Support of Switched Multimegabit Data Service", Bellcore Technical Advisory TA-TSY-000772, Issue 3, October 1989.

[45] "High Performance Parallel Interface - Mechanical, Electrical and Signalling Protocol Specification (HIPPI-PH)", ANSI x3.183-1991, 1991.