Integrated Dynamic Bandwidth Allocation and Congestion Control
in Satellite Frame Relay Networks

by

Chee-Heng Lee

Submitted to the Department of
Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the Degrees of

Bachelor of Science
and
Master of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1994

© 1994 Chee-Heng Lee
All rights reserved

The author hereby grants to MIT permission to reproduce and to distribute publicly
paper and electronic copies of this thesis document in whole or in part.

Signature of Author .................................................................
Department of Electrical Engineering and Computer Science
January 27, 1994

Certified by .................................................................
Dr. Steven G. Finn
Thesis Supervisor (Academic)

Certified by .................................................................
Dilip S. Gokhale
Company Supervisor (COMSAT Laboratories)

Accepted by .................................................................
Frederic R. Morgenthaler, Chairman
Committee on Graduate Students
Integrated Dynamic Bandwidth Allocation and Congestion Control in Satellite Frame Relay Networks

by
Chee-Heng Lee

Submitted to the Department of Electrical Engineering and Computer Science on January 14, 1994, in partial fulfillment of the requirements for the degrees of Bachelor of Science and Master of Science in Electrical Engineering

Abstract

A study was conducted to evaluate the performance of various congestion control mechanisms in satellite frame relay networks. In particular, the advantage of applying the dynamic bandwidth allocation capability of some satellite systems to frame relay, as part of the congestion control mechanism, was investigated. Two bandwidth allocation algorithms were developed to achieve this end. The performance of a satellite frame relay network with dynamic bandwidth allocation capability was evaluated experimentally vs that of another satellite frame relay network without such capability. Various end-user congestion control algorithms were simulated to study their performance when used with the bandwidth allocation algorithms in the network having the dynamic bandwidth allocation capability. Results showed that the network with such capability demonstrated better overall performance with the appropriate end-user congestion control algorithms and parameters, than the network without such capability. Further, this superior network performance was generally consistent across a wide variety of network loads, traffic types, and Network Control Center signaling delays.

Thesis Supervisors: Dr. Steven G. Finn
Lecturer
Department of Electrical Engineering and Computer Science

Dilip S. Gokhale
Manager, ISDN/Data Communications
COMSAT Laboratories
Acknowledgments

I would like to thank all the people who made this thesis possible. They include Dr. Steven Finn, who made some valuable suggestions at various stages of the development of this work; Dilip Gokhale, who carefully and patiently guided me through this work with his insights and resourcefulness; my colleagues, Sumant Hattikudur, Ajai Kaul, Faris Faris, Anukka Piironen, Anil Agarwal, and Michael Wolff, who rendered their help and shared their expertise with me in various ways; and Barbara Wassell of the Editing Department at COMSAT Laboratories, who helped me prepare the final copy of this thesis document. My appreciation also goes to my family, who provided me with their love and moral support. Lastly, I would like to thank my Lord, Jesus Christ, for without Him I would not have come this far.
Table of Contents

Abstract ...................................................................................................................................... 2
Acknowledgments .................................................................................................................. 3
Table of Contents ..................................................................................................................... 4
List of Figures ............................................................................................................................. 6
List of Tables ............................................................................................................................... 7
1. Introduction .......................................................................................................................... 8
   1.1 Frame Relay and Its History ............................................................................... 8
   1.2 Overview of Thesis ........................................................................................... 9
2. Background........................................................................................................................... 10
   2.1 Chapter Organization ...................................................................................... 10
   2.2 Frame Relaying ............................................................................................... 10
   2.3 Congestion Control .......................................................................................... 12
   2.4 Satellite Networks vs Terrestrial Networks .................................................. 14
   2.5 COMSAT's Bandwidth-on-Demand (BoD) Satellite System ...................... 15
3. Goals of the Project .......................................................................................................... 18
   3.1 Chapter Organization ...................................................................................... 18
   3.2 Bandwidth Availability on Demand .............................................................. 18
   3.3 Efficient Bandwidth Utilization ...................................................................... 20
   3.4 Congestion Notification to End-Users ............................................................. 21
   3.5 Fairness in Sharing Network Resources ......................................................... 23
4. Description of the Project ................................................................................................. 25
   4.1 Chapter Organization ...................................................................................... 25
   4.2 Description of Network Models ...................................................................... 25
      4.2.1 Network Architectures ........................................................................... 25
      4.2.2 Node Model ............................................................................................. 28
      4.2.3 End-User Terminal Model ..................................................................... 30
      4.2.4 Network Control Center (NCC) Model ............................................... 32
   4.3 Description of Control Algorithms .................................................................... 33
      4.3.1 Comparative Studies of Other Window Congestion Control Algorithms ........................................ 33
      4.3.2 Enhanced Congestion Control Algorithm ............................................. 40
      4.3.3 Distributed Bandwidth Allocation Algorithm ....................................... 42
      4.3.4 Integration of Bandwidth Allocation and Congestion Control Algorithm ........................................ 46
4.4 Description of Simulation .................................................................................................... 47

5. Experimental Results .............................................................................................................. 49
5.1 Chapter Organization .............................................................................................................. 49
5.2 Definitions of Simulation Input and Output Statistics .......................................................... 49
5.3 Comparison of Networks with Different Control Schemes and Parameters .......................... 51
  5.3.1 Performance of a BoD Network vs a Non-BoD Network ................................................. 51
  5.3.2 "Aggressive" Parameters vs Q.922 Default Parameters .................................................. 56
  5.3.3 TCP on the BoD Network and the Non-BoD Network .................................................... 61
  5.3.4 Exponential Window Increase Strategy on the BoD Network ........................................ 64
  5.3.5 Centralized Control on the BoD Network ......................................................................... 67
5.4 Parameter Sensitivity Analysis ............................................................................................. 71
  5.4.1 Utilization Threshold in Bandwidth Control Algorithm .................................................. 71
  5.4.2 Maximum Link Capacity ................................................................................................. 73
  5.4.3 Transmission Queue Buffer Size ..................................................................................... 76
  5.4.4 Percentage of Traffic Mix (File Transfer and Interactive) ............................................. 78
  5.4.5 NCC Signaling Delay ....................................................................................................... 82
5.5 Statistical Accuracy .............................................................................................................. 84

6. Conclusions ........................................................................................................................... 86

Appendix .................................................................................................................................... 90
  A.1 Description of the Control Algorithms in Finite State Machine ..................................... 90
    A.1.1 Network Algorithm for Congestion Control (as described in 4.3.2) ......................... 90
    A.1.2 Network Algorithm for Integrated Bandwidth Allocation and Congestion Control (as described in 4.3.3. and 4.3.4) ................................................................. 91
  A.2 Method of Biasing the File Transfer Traffic to the Specified CIR by the Law of Large Numbers ............................................................... 93
  A.3 Numerical Verification of Some Simulation Results ....................................................... 95

References .................................................................................................................................. 100
List of Figures

Figure 1. Comparison between frame relay and X.25 ............................................................... 11
Figure 2. A satellite frame relay network using ACTS or BoD system .................................. 16
Figure 3. A frame relay network model .................................................................................. 27
Figure 4. A node model showing internal queue architecture and congestion control scheme .......................................................... 29
Figure 5. A node model showing internal queue architecture and integrated bandwidth allocation and congestion control scheme .......................................................... 30
Figure 6. An end-user terminal model .................................................................................... 32
Figure 7. Network Control Center model ............................................................................ 32
Figure 8. Network performance as a function of load .......................................................... 34
Figure 9. Distributed bandwidth allocation model ............................................................... 45
Figure 10. Initial configuration of Network I and permanent configuration of Network II ...... 53
Figure 11. Performance of Network I vs Network II .............................................................. 55
Figure 12. Performance of Network I vs Network II using Q.922 default parameters ......... 56
Figure 12. Performance of Network I vs Network II using Q.922 default parameters ......... 57
Figure 13a. Comparison of "aggressive" and Q.922 default parameters in Network I ........ 59
Figure 13b. Comparison of "aggressive" and Q.922 default parameters in Network II ......... 60
Figure 14. Performance of TCP in Network I and Network II .............................................. 63
Figure 15. Performance of exponential window increase strategy in Network I .......... 65
Figure 15. Performance of exponential window increase strategy in Network I .......... 66
Figure 16. Performance of Network I with centralized and distributed-control algorithms ........................................................................ 70
Figure 17. Performance of Network I with different utilization thresholds ......................... 72
Figure 17. Performance of Network I with different utilization thresholds ......................... 73
Figure 18. Performance of Network I with different maximum link capacity limits ............ 75
Figure 19. Performance of Network I with different queue buffer sizes ......................... 77
Figure 19. Performance of Network I with different queue buffer sizes ......................... 78
Figure 20. Performance of Network I and Network II with various traffic mixes at 0.85 load ........................................................................ 80
Figure 21. Performance of Network I and Network II with various traffic mixes at 0.5 load ........................................................................ 81
Figure 22. Performance of Network I with different NCC signaling delays ...................... 83
Figure 22. Performance of Network I with different NCC signaling delays ...................... 84
Figure 23. Performance of Network I with different random seed numbers ..................... 85
Figure 24. Finite state machine of the network congestion control algorithm ................... 90
Figure 25. Finite state machine of the integrated bandwidth allocation and congestion control algorithm ........................................................................ 91
Figure 26. File transfer traffic simulator .............................................................................. 93
List of Tables

Table 1. Frame loss ratio corresponding to Figure 15 (iii) .............................................. 67
Table 2. Window size and amount of data sent as a function of round-trip time ............................................................ 97
Table 3. Comparison of calculation and simulation results ...................................................... 99
circuits. However, this service was found to be unsuitable for new applications such as LAN (Local Area Network) interconnection. Frame relay is an additional packet mode bearer service created to fill such needs. This service is specified in CCITT1 Recommendation I.122 (1988) and ANSI T1.606 (1990). To date, frame relay services offered as PVCs (Permanent Virtual Circuits) are available from many commercial network service providers (AT&T, Compuserve, Wiltel, Sprint, etc.) both as a stand-alone service or as part of their ISDN services. While these networks are all based on terrestrial transmission facilities (primarily over fiber optic cables), it is believed that frame relay services can also be effectively deployed over satellite links.

1.2 Overview of Thesis

The remainder of this thesis is divided into five chapters. Chapter 2 provides the background materials and major concepts involved in this work. The objectives of the project, from which this thesis is derived, are described in chapter 3. Chapter 4 gives a detailed description of various simulation models and the control mechanisms/algorithms used in the simulations. Chapter 5 presents the results of a series of simulations, with the relevant discussion of the results and their implications. Finally, conclusions and future research are discussed in chapter 6.

---

1 In 1993 the International Telecommunications Union (ITU) was reorganized. The CCITT is now known as International Telecommunications Union - Telecommunications Standardization Section (ITU - TS). The old designation "CCITT" is used in this thesis to avoid confusion.
1. Introduction

This thesis presents the results of a study on the congestion control aspects of a frame relay network that uses satellite transmission facilities. In particular, the suitability of applying dynamic bandwidth allocation capability provided by some satellite networks, such as COMSAT Laboratories' Bandwidth on Demand (BoD) system, to frame relay as part of the congestion control mechanism was investigated in the study.

1.1 Frame Relay and Its History

For the past several decades we have witnessed a dramatic evolution in the global communication scene. The advances in computer and communication technology have led to a change in the nature of communication services demanded by users. For more than a century, voice traffic has been the predominant type of communication and it has largely shaped the communication infrastructure we find in most countries today. This system was designed for analog voice transmission and is proving inadequate for many modern communication needs, such as data transmission, facsimile, and video. ISDN (Integrated Services Digital Network) is one aspect of the evolution from analog communication to digital communication, driven by the need of the users. Its basic tenet is to provide a variety of services using common access arrangements with a limited set of access interfaces.

One access interface is defined for packet data transmission over ISDN. Until recently, the only packet mode bearer service offered by ISDN was X.25 virtual
2. Background

2.1 Chapter Organization

This chapter provides an overview of the network concepts used in the rest of the thesis. Section 2 describes the mechanism of frame relaying, and how it compares to other packet-switched technologies. In the following section, we examine various congestion control alternatives in a frame relay network that have been proposed by national and international standard organizations. Some generic aspects of implementing frame relay networks using a satellite backbone are discussed in section 4. Finally, COMSAT Laboratories' Bandwidth-on-Demand (BoD) system, from which the dynamic bandwidth allocation capability is derived, is described in section 5.

2.2 Frame Relaying

Frame Relaying is the technical term for the simplified switching and transfer of data units at layer 2 of the Open Systems Interconnection (OSI) model for information transfer. Conventional packet-switched network technologies like X.25 were invented during the era when the transmission facility was characterized by low transmission rate and high bit error probability [1, 3, 4]. Therefore, such networks typically have to employ complicated layer-2 and layer-3 protocols where error detection, error correction and flow control, among other functions, are to be performed both within the network as well as at the Data Terminal Equipment - Data Communications Equipment (DTE-DCE) interface [2-4]. With the advent of
high speed and reliable digital transmission facilities, these technologies are no longer suitable as the complexity of such protocols lead to transmission bottlenecks at the network nodes. The streamlining of layer-2 and layer-3 protocols to avoid these bottlenecks results in the so-called fast packet switching concept. Frame Relay is one of the packet-switched technologies that apply this concept.

A frame relay network performs logical channel multiplexing, frame transparency, error detection and frame discard within its layer-2 protocol [16]. Error correction and flow control responsibilities have been assigned to the end systems. Figure 1 compares the protocol architecture between a traditional X.25 network and a frame relay network. With this reduction in protocol processing at the network nodes, high speed data transfer can be achieved. However, unlike X.25, frame relay has no explicit protocol mechanism for flow control of user traffic. While X.25 provides both sliding window and stop-go flow control in both its level-2 and level-3 protocols, frame relay has neither. What frame relay does provide is the option to perform congestion control\(^2\). Using words from one critic: "everything is optional and nothing is mandatory" [19].

\(\text{Figure 1. Comparison between frame relay and X.25}\)

\(^2\)Congestion control in frame relay stays as an option (rather than a requirement) primarily to allow for more flexibility to vendor equipment. When the network is under heavy load, particularly with bursty data traffic, some form of congestion control mechanism is necessary to maintain network performance.
2.3 Congestion Control

In [20], congestion is said to occur in a network if for any time interval, the total sum of demands on a resource is more than its available capacity, i.e., mathematically speaking:

\[
\int_{t}^{t+\delta t} \sum \text{demand} > \int_{t}^{t+\delta t} \text{available resources} \quad \text{(EI)}
\]

In a packet-switched network, network resources consist of one or more of the following: buffers, link bandwidths, processor speeds, etc. A node may become congested if it cannot transmit incoming packets faster than they arrive. If this situation is allowed to persist, then within a finite time interval packets will have to be discarded because of lack of buffer space, regardless of the size of the buffer. For data networks, discarded or delayed packets may trigger retransmissions, which puts even more load on the node resources. Eventually, a situation termed as "congestion collapse" by Nagle [23] is likely to happen where each packet is retransmitted several times on average due to excessive delay or buffer overflow.

CCITT and ANSI have proposed several optional procedures and features for managing congestion in a frame relay network. For example, admission control can be enforced at the network entry points by requiring users to conform to the traffic parameters negotiated at the PVC setup time. There are three standard traffic parameters currently specified in ANSI document [16]:

- **Committed Information Rate (CIR)** - defined as the rate at which the network agrees to transfer data under normal conditions.

- **Committed Burst Size (Bc)** - defined as the maximum number of bits, during time interval T, the networks agrees to accept under normal conditions.
• Excess Burst Size (Be) - defined as the maximum number of bits, during time interval T, the network agrees to accept above the Committed Burst Size, Bc.

The above parameters are usually used in conjunction with the Discard Eligibility (DE) bit in the frame relay header to control congestion by selectively discarding frames within the network. The DE bits are usually set to 1 at the network ingress points on frames which exceed the Bc parameter. If frame discard becomes necessary in the network due to congestion, these marked frames can be given a higher discard priority. Alternatively, when users attempt to transmit data above their CIRs, some frames can be discarded at the network ingress nodes instead of being admitted into the network. Some algorithms proposed to enforce these parameters on the users can be found in [19, 24].

In addition to admission control features, CCITT and ANSI have also defined Forward Explicit Congestion Notification (FECN), Backward Explicit Congestion Notification (BECN) and Consolidated Link Layer Messages (CLLM) [5, 16]. Both FECN and BECN are a single bit carried by the frame relay I (Information) frame header. The use of BECN, FECN and CLLM for congestion control is described below [25]:

• BECN - this bit may be set to 1 by a congested network to notify the user that congestion avoidance procedures should be initiated, where applicable, for traffic in the opposite direction of the transmitted frame.

• FECN - this bit may be set to 1 by a congested network to notify the user that congestion avoidance procedures should be initiated, where applicable, for traffic in the direction of the transmitted frame.

3 It is however important to keep in mind that such traffic enforcement schemes attempt to enforce fairness and uniformity on traffic by tagging and dropping packets. Data traffic is known to be extremely bursty and non-uniform, and such mechanisms may severely degrade the performance of a data network. Due to this reason, we do not focus our attention on these congestion control options.
• CLLM - this is a variation of BECN flow control protocol using a "broadcast type message" containing a list of congested users.

2.4 Satellite Networks vs Terrestrial Networks

Up until now, frame relay networks have been typically realized via the use of terrestrial links, particularly via fiber optic cables. Given the requirements for high bandwidth and low bit error rate, we believe that satellite links can also be used in a frame relay network with appropriately developed network procedures and algorithms. As usual, there are issues of trade-off involved in deciding which realization is better [15]. In view of this, some unique characteristics and capabilities of a satellite network that differentiate it from a fiber-based network are listed below. They are:

• large propagation delay, in the order of 250 - 300 ms for a one way node-to-node transmission.
• medium bit error rate, in the order of $1 \times 10^{-6}$ to $1 \times 10^{-8}$.
• possibility of real time, dynamic switching of circuits among network nodes.
• 1-hop-only and distance-insensitive transmission across the network.

Propagation delay is determined by the distance between two communicating points and the speed of light. Terrestrial links are often short enough that they can be assumed to have negligible propagation delay. The use of fiber optics as a transmission medium in a terrestrial network also makes the assumption of very low bit-error rates valid. However, both of these assumptions are not true for a
satellite network, as shown by the above numbers. These factors directly influence various aspects of satellite network design and implementation.

The dynamic circuit-switching capability is made possible by the basic concepts of satellite communications. Each satellite consists of several transponders which can support up to several hundred circuits by using multiple access technologies such as TDMA (Time Division Multiple Access). These circuits can be switched from one network node to another at a low overhead. This capability will be discussed in more detail when we describe the BoD system in a later section.

Though it is generally expected that optical fiber is going to become the dominant mode of transmission in the future [26], there are reasons to believe that satellite communication will continue to sustain its popularity. One of the reason is simply that even though fiber offers a much higher bandwidth compared to satellite, much of this bandwidth is usually not accessible to the users [15]. This stems from the fact that there is still a high infrastructure cost to make the bandwidth accessible to everyone. Another reason for the continuing popularity of satellite communication is its use for broadcast and mobile communication services. Satellite networks also possess the additional ability to dynamically reconfigure intra-network trunks to meet changing user requirements.

2.5 COMSAT's Bandwidth-on-Demand (BoD) Satellite System

The bandwidth on demand capability in satellite networks is provided by at least two available systems: COMSAT Laboratories' Bandwidth-on-Demand (BoD) and NASA's Advanced Communications Technologies Satellite (ACTS). Using the service provided by these systems, link capacity between any two nodes in a satellite
network can be dynamically adjusted to meet traffic needs. The ACTS or BoD systems will set up or release circuits of various types (simplex, multipoint, full duplex) at variable rates (64 kbps to 2.048 Mbps) on users' requests. In the context of this thesis, a frame relay node poses as a user to this service, which constitutes the basis for our dynamic bandwidth allocation scheme. A conceptual view of a satellite frame relay network utilizing the ACTS or BoD systems is shown in Figure 2.

![A satellite frame relay network using the ACTS or BoD system](image)

**Figure 2. A satellite frame relay network using the ACTS or BoD system**

COMSAT Laboratories' BoD system is based on Time Division Multiple Access (TDMA) technology. The system consists of a master station, the Network Control Center (NCC), which provides time-slot assignment and management to the overall network. Users of the system are assigned periodic time-slots in which data or voice is transmitted in a short burst at the designated time. The periodic time-slots are usually in increments of 64 kbps or multiples of 64 kbps, and these time-slots are
assigned dynamically to the users via standard ISDN switched-circuit call set-up procedures. While this thesis primarily addresses the frame relay network as shown in figure 2, a second method of providing dynamic-bandwidth frame relay services is also examined. In this method, the functionality of a frame relay node is moved into the BoD system and the assignment of network resources (i.e. link bandwidth in terms of time-slots) is performed in a centralized manner by the NCC. Conceptually, the frame relay nodes as shown in figure 2 would appear inside the ACTS/BoD "cloud" in this method.
3. Goals of the Project

3.1 Chapter Organization

The motivations for this research project are presented in this chapter. The dynamic bandwidth assignment capability of COMSAT's BoD system prompts us to study its applicability in providing more effective frame relay services, particularly with respect to fixed-bandwidth satellite networks. This study also aims to aid in current and future implementations of satellite frame relay networks within COMSAT Laboratories in particular and other research organizations in general.

In each of the following sections, we discuss in detail several issues that are important from a network user's and/or a network administrator's perspective:

- bandwidth availability to meet dynamically varying user demand.
- efficient utilization of overall network bandwidth.
- congestion notification to end-users when network resources are depleted.
- fairness in sharing network resources.

3.2 Bandwidth Availability on Demand

Making bandwidth available on demand is high on the list of goals, especially from a user's perspective. Data traffic is typically characterized by a high degree of randomness. This randomness is generally caused by the random number of active
users and the bursty nature of each application. A packet-switched network, like frame relay, overcomes this problem by having multiple users share the network resources on a demand basis. However, such statistical sharing can potentially lead to congestion in the network when several users demand the network resources simultaneously.

In most literature that we have come across, statistical multiplexing is achieved within a single physical link with fixed bandwidth (e.g. multiple virtual circuits traversing a 64 kbps trunk between 2 nodes) [10-11]. It has been shown that when several users attempt to transmit data across the link simultaneously, an effective congestion control mechanism will maintain the total throughput of that link. Nevertheless, throughput per individual unavoidably suffers as the link capacity has to be shared by a greater number of users when congestion occurs.

We would like to investigate the effect on throughput when the bandwidth between any two nodes in a network can be dynamically adjusted\(^4\) according to traffic load, under the constraint of a constant total network bandwidth. We suspect that such capability will not only improve delay-throughput performance per individual, but total network throughput will also improve as a result of statistical multiplexing in a larger context. An over-simplistic\(^5\) but conceptually correct way to view this is that the total network bandwidth is treated as a single entity, and statistical multiplexing among all network users is achieved within this entity. With respect to this idea, one of the objectives of the project is to design a bandwidth allocation algorithm that can be applied to a satellite-only frame relay network based on the

---

\(^4\) As in COMSAT's BoD or NASA's ACTS network.

\(^5\) This view is over-simplistic for two reasons. First, the bandwidth allocation is discrete (e.g. multiple 64 kbps), so some inefficiency due to this inflexibility is bound to occur. Second, there is delay involved in setting up/tearing down a circuit. These factors are going to degrade the ability of BoD systems to function as network-wide statistical multiplexers.
BoD system, in which the capability of dynamic bandwidth switching is provided. Performance of a network implementing both a bandwidth allocation scheme and a congestion control scheme is compared with another network which implements the same congestion control scheme but with fixed link capacities in the network. The two cases compared will have the same total network capacity.

3.3 Efficient Bandwidth Utilization

From the network's perspective, the ability to dynamically allocate bandwidth to adapt to a user's demand also improves the efficiency of bandwidth utilization. To achieve perfect utilization of resources in a network, the network has to conform to the work-conserving principle. In queuing theory terms, a server is considered to be work-conserving if it is busy for as long as the queue is not empty. In view of the dynamic bandwidth allocation capability, the idea is to treat the entire satellite network as a server, with its service rate equal to the total network capacity. If the server is work-conserving and can be switched from one user to another without any delay, perfect utilization of network bandwidth can be achieved.

However, in a real satellite network, switching of bandwidth can only be performed with non-zero propagation and processing delay. Nevertheless, if bandwidth can be dynamically assigned to network nodes, it will be desirable for a node to get more bandwidth when it is congested (its current link capacity is highly or over-utilized), and to release some of its bandwidth to another congested node when it is under-utilizing its current link capacity. Even though the work-conserving principle is not achieved in the perfect sense, one can immediately see that in getting closer to that principle, bandwidth utilization can be improved. Therefore, even from a theoretical view point, the dynamic bandwidth switching
capability can be used to approach the work-conserving principle and hence more efficient utilization of bandwidth can be achieved in the network.

An issue related to the efficiency in bandwidth utilization is the congestion control protocols used by the end-systems of a network. Ideally, it shall be the aim of a congestion control protocol to maintain continuous transmission on a link if it is not congested (this is called "filling" the pipe in the networking literature). In doing so, work-conservation on a per link basis can be achieved. However, continuous transmission of data is often limited by the window size of the transmitter for congestion control purposes. Large propagation delay inherent in a satellite link forces us to rethink the feasibility of a closed-loop window congestion control [9, 10], which has been frequently proposed as a solution to the congestion problem in frame relay networks [3, 6-8]. This is because in order to extract high bandwidth utilization from a satellite network, large window size has to be used. However, the use of a large window size, when coupled with delayed feedback, often leads to large amplitude oscillations in bandwidth utilization. The advantage of an open-loop feedback mechanism over a closed-loop feedback mechanism in a satellite network has been shown in [22]. The open-loop feedback mechanism is able to achieve a much better bandwidth utilization compared to the closed-loop mechanism.

3.4 Congestion Notification to End-Users

In frame relay, as well as in some other reliable data communication protocols, end users can be informed of congestion by two methods. Implicit notification of congestion is said to have occurred if end-users receive multiple acknowledgments for the same packet or time-out on a particular packet. These two events usually result from a packet being dropped in the network because of buffer
overflow or transmission errors. On the other hand, the network may explicitly notify the end-users of congestion by monitoring the network resource utilization. For example, the utilization of a link or the buffer occupancy level can be used as an indication to congestion. Explicit notification can be conveyed to a user either by setting a bit carried by the packet header, or by creating a control packet to carry the information.

In a satellite network, there are several factors that weaken the effectiveness of implicit congestion notification. An implicit notification takes a round trip time interval to get to a user, i.e. between the event of congestion (i.e. a packet dropped by the network) and the event of the user getting the notification (by a time out or multiple acknowledgment event), there is an elapsed time of a satellite round trip delay. The satellite delay-bandwidth product is so large that implicit congestion feedback information usually becomes obsolete by the time the user gets it. The implicit notification mechanism has been shown to be ineffective in a satellite environment in [10], in which the throughput and delay performance of a hybrid network consisting of one satellite link degenerates with increasing load.

A second, though less significant reason, is the presence of satellite bit errors. Thus a packet dropped by the network because of transmission error may be mistakenly treated as a congestion indication (the end-user has no way of telling the difference between the two). Also, depending on the end-user protocols, every frame loss in the network may cause one or more round-trip delays contributing to bandwidth waste due to "draining" of the bit pipe. In terms of satellite delay-bandwidth product, this waste is relatively expensive.
To ensure that a network can maintain its performance in the face of congestion, an effective congestion notification scheme is necessary. One such study by COMSAT Laboratories has shown that in a satellite environment, an explicit backward notification scheme performs considerably better than an implicit or forward-explicit scheme, particularly if congestion occurs at the network ingress node [22, 30]. This result leads us to believe that in order to achieve the goal of effective congestion notification in a frame relay network, the BECN protocol in frame relay must be employed.

3.5 Fairness in Sharing Network Resources

An effective congestion control scheme must also be able to maintain fairness among users of a network. Fairness can be defined in a variety of ways [12], and it is therefore necessary to adopt one view of fairness before one even attempts to achieve it. In the context of this thesis, we can look at the fairness issue at two levels. On the DLCI (Data Link Connection Identifier) level, our objective is to identify and throttle the source of congestion such that the "innocent" users shall not notice a significant degradation in network performance. It has been shown [13] that a multiplicative decrease/additive increase dynamic window algorithm converges to this fairness goal.

On the network level, our aim is to ensure that the bandwidth is distributed in proportion to traffic load on each link. More often than not, a compromise will have to be reached among achieving various objectives such as maximizing throughput, maintaining fairness and reducing implementation overhead. As described in [12], the exact balance in determining "who-should-get-how-much" is
often reached by trial-and-error. This issue will be further discussed when we present
the adaptive bandwidth allocation algorithm in the next chapter.
4. Description of the Project

4.1 Chapter Organization

This chapter describes, in detail, various aspects of the project comprising this thesis. In the next section, network models used in the simulations are presented. In section 4.3, we first study several window congestion control algorithms that have been proposed in the literature. Then we present the enhanced congestion control algorithm (based on CCITT Recommendation Q.922 [31]) and the integrated bandwidth allocation and congestion control algorithm. Finally, the simulation methodology is presented in section 4.4.

4.2 Description of Network Models

4.2.1 Network Architectures

In view of the potential complexity of this project, we have made the following simplifying assumptions regarding our network models:

- Number of nodes in the subnet is limited to less than 10.
- Network nodes are connected to each other by satellite links only.
- Total network bandwidth is \( N \times 64 \) kbps, where \( N \) is a constant integer.
- Satellite Bit Error Rate (BER) is assumed to be zero.
• End-user terminals are directly connected to the network nodes via terrestrial links\(^6\).

The network shown in Figure 3 is an example of the first type of network that we are going to simulate. As an illustration, we may assume a full-mesh network topology with full-duplex links of 128 kbps. The capacity of each link is fixed. Each link also has a one way propagation delay of 300 ms. Total bandwidth of such a network can be calculated according to the following:

Given that there are \( n \) nodes in the network, then

- number of full duplex links = \( n \times (n-1) + 2 \)
- capacity of each link = 128 kbps \( \times 2 \)

Therefore,

- total network bandwidth
  - \( N \times 64 \text{ kbps} \)
  - \( N \times 64 \text{ kbps} \)
  - \( N \)

\[ \text{total network bandwidth} = \text{number of full duplex links} \times \text{link capacity} \]
\[ = n \times (n-1) + 2 \times 128 \times 2 \text{ kbps} \]
\[ = n \times (n-1) \times 128 \text{ kbps} \quad \text{(E2)} \]

The second type of network that we are simulating is similar to the first type with the following exceptions:

• A static, full-mesh connectivity with full-duplex links of \( S_{\text{init}} \) kbps is maintained by the network (\( S_{\text{init}} < 128 \)).

• a pool of \( M \times 64 \) kbps point-to-point simplex links is dynamically shared among the satellite nodes.

• the shared bandwidth, when not being used, is maintained by the Network Control Center, which is modeled as an independent node accessible to all frame relay nodes via low bandwidth signaling channels.

\(^6\)The terminal emulates a bridge or router, which is connected to one or several LANs. Each terminal consists of multiple DLCIs.
Figure 3. A frame relay network model

The parameter M can be determined by simple calculation, assuming a value for $S_{init}$ and the same total bandwidth across the two networks. For example:

Assuming

- $S_{init} = 64$ kbps
- total network bandwidth = $N \times 64$ kbps
- $= n \times (n-1) \times 128$ kbps from (E2)

Then,

- total static bandwidth = $n \times (n-1) \times S_{init}$ kbps
- = $n \times (n-1) \times 64$ kbps
- common pool bandwidth = total network bandwidth - total static bandwidth
- = $n \times (n-1) \times (2-1) \times 64$ kbps
- $M \times 64$ kbps
- $= n \times (n-1) \times 64$ kbps
- $M = n \times (n-1)$

The common pool of bandwidth is dynamically shared among the nodes according to the algorithms we have developed. The total network bandwidth is kept
the same across the two networks to make the comparisons of their simulation results valid. For the fixed-bandwidth network, each link will be assigned a constant capacity equivalent to the sum of CIRs (Committed Information Rates) of all virtual circuit connections on that link. Therefore the total network bandwidth is equal to the sum total of CIRs of all virtual circuit connections in the network. The parameter \( S_{\text{init}} \) in the BoD network will be set to a fraction of the link capacity in the fixed-bandwidth network, with the difference contributing to the shared bandwidth accessible to all nodes.

4.2.2 Node Model

The node model we have developed for the fixed-bandwidth network is shown in Figure 47. All inbound\(^8\) frames go into their corresponding transmission (or network) queues, of which one of them is shown in the figure. Processing delay is not modeled here, which is a reasonable assumption in the context of frame relay. Therefore, a frame arriving in the node will experience queuing delay, transmission delay and propagation delay before reaching another node. Each transmission link is modeled as a server, with the mean service rate equal to the mean frame size divided by the link bandwidth. Frames arriving from other frame relay nodes (i.e. outbound frames) are routed into the corresponding users' queues as shown in the lower left corner of the figure. There is one such queue corresponding to each access link. We again assume that the processing delay is negligible. As the terminals are connected to the network nodes via terrestrial access links, the outbound frames will only experience queuing delay and transmission delay. We further make the

\(^7\)Access control and traffic policing can be exercised at the network ingress point, as shown in Figures 4 and 5. However, this option is not invoked in our simulation. With its primary thrust being the ability to ensure fairness, we decided to leave out this option so that we can focus more on the congestion control and dynamic bandwidth switching aspects of the network.

\(^8\)The term "inbound" and "outbound" are used here in the same way as they are used by the subway/metro systems in big cities. Therefore, an inbound frame (train) is a frame (train) that is going into the network (city) and an outbound frame (train) is a frame (train) that is leaving the network (city).
assumption that all queues (network and user) have finite buffer sizes and frames arriving at a full queue are to be discarded. The congestion control scheme monitors the traffic level of all the queues, and it generates congestion feedback messages to end-users when necessary.

The node model for the second type of network we are simulating is similar to the one above with the following exceptions:

- The service rate for the inbound network queue is variable.
- The congestion control scheme is replaced by an integrated bandwidth allocation and congestion control scheme.

Figure 5 shows a node model with variable outgoing link rate and the integrated bandwidth allocation and congestion control scheme.

Figure 4. A node model showing internal queue architecture and congestion control scheme
4.2.3 End-User Terminal Model

As shown in Figure 6, each terminal consists of one or more traffic generators. The traffic generators are independent entities as each of them models a virtual circuit connection, represented by their Data Link Connection Identifier (DLCI). Traffic in this module is modeled as either a bursty ON-OFF source, to emulate a file transfer connection, or a Poisson arrival source, to emulate an interactive data connection. For the file transfer traffic generator model, each file is generated at an exponential interarrival time, $X$ with uniformly distributed file size, $L$. When the entire file is generated, the traffic source goes to an OFF period in which it does not generate any traffic. The next ON period starts when the next file is generated. Interactive traffic is simulated by a Poisson packet arrival stream. The average traffic generation rate for both types of traffic is set to the Committed Information Rate (CIR) of a DLCI specified at simulation time (cf. Appendix A.2).
The amount of traffic generated by each DLCI during a simulation is determined by its CIR and a load parameter. For example, if a virtual connection has 64 kbps of CIR, and the load parameter is equal to 0.5, then the average traffic generation rate for that connection is 0.5 x 64 kbps = 32 kbps. However, the actual amount of traffic that goes into the network is controlled by its corresponding transport protocol. The transport module is able to buffer a fixed number of frames from the traffic generator, beyond which the traffic generator will stop generating any frame until the transport buffer opens up. The transport protocol is designed to perform reliable communication with its peer at the opposite end of the connection, i.e. the destination application layer always receives a stream of in-sequence frames from its transport layer. The congestion control mechanism we use in the simulations is incorporated within the transport layer protocol. We also make the assumption that all end-users are well-behaved, in that they reduce their traffic loads correspondingly when they receive congestion notifications.

Inbound traffic streams from different DLCIs are multiplexed into the access link queue. Similarly, outbound traffic streams have to be demultiplexed into their corresponding destination DLCIs. Therefore, a multiplexer/demultiplexer module is incorporated into the terminal model. A complete picture of the terminal architecture is shown in the Figure 6.
4.2.4 Network Control Center (NCC) Model

For simulation purposes, a simple model for the Network Control Center (NCC) was designed. The NCC is modeled as a process (finite state machine) with a single-server-single-queue FCFS discipline as shown in the picture below:

When a network node attempts to acquire or release some capacity, it sends a message to the NCC. In our simulations, this message is sent via a low-speed signaling channel. Because the message size is very small, transmission delay relative to the propagation delay can be assumed to be negligible. However, the message shall incur propagation delay associated with traversing the signaling channel to the NCC. When the message arrives at the NCC, it goes into a processor.
queue as shown in the diagram above. The server shall process all messages in the queue in a FCFS order. A constant service time of 10 ms is assumed for each message.

A counter variable is used to model the number of circuits available at the NCC at any time. Thus, when a frame relay node acquires a circuit from the NCC, the counter variable is decreased by 1. Similarly, the counter variable is increased by 1 when a circuit is released back to the NCC. When the counter variable is 0, the entire resource pool at the NCC is depleted and no circuit can be allocated at this time. There is always a round-trip delay between the time the signal is sent by the frame relay node and the actual allocation of the circuit by the NCC. Therefore, the entire process of acquiring or releasing a circuit via a satellite signaling channel takes 300ms + 10ms + 300ms = 610ms. This NCC model, though much simplified, is an adequate representation of the real system for our simulations.

4.3 Description of Control Algorithms

4.3.1 Comparative Studies of Other Window Congestion Control Algorithms

One way of categorizing any congestion control mechanism is by the region in which the mechanism operates. Referring to Figure 8, congestion avoidance generally operates within region II (between knee and cliff), i.e. before congestion actually occurs. However, in region III, where congestion has occurred, (i.e. degraded throughput and delay performance), a separate set of mechanisms have to be invoked to recover the network from congestion, i.e. back to region II. These mechanisms are usually termed as congestion recovery mechanisms. In this section, we will discuss some algorithms proposed in the literature for each region.
The first scheme that we are going to look at is first proposed by Ramakrishnan and Jain [13] for connectionless data networks and later by Goldstein [7] specifically for frame relay. It is a mechanism that has been suitably termed "explicit binary feedback" by both papers. In this congestion avoidance scheme, every packet carries one bit in its header which can be set by a network node if the node queue length exceeds a certain threshold. The end-user continuously computes the ratio of "marked" packets to the total number of packets received during a round-trip interval. If this ratio exceeds 50%, the network is considered as about to be congested. Depending on this ratio, a user then adjusts its window size according to a multiplicative decrease/additive increase algorithm. This scheme aims to prevent packet loss due to buffer overflow by constantly monitoring the time-averaged queue length (to filter out transient congestion) and conveying this information to the end-users via a bit on each packet.

A user's window size is adjusted once every two round trips, as the congestion bits are set on packets going in the direction of congestion. These bits are then piggybacked on the acknowledgments going back to the source of congestion. The user takes another round-trip to filter the congestion bits before updating its window. The primary difference of this scheme compared to a source-quench mechanism is that first, it achieves congestion avoidance rather than congestion recovery, and second, it does not generate additional packets to achieve its goal. It is not clear from the paper how this scheme will scale with the delay-bandwidth product of the networks, which is the primary concern here. Jacobson has also pointed out that this scheme requires a new bit in the packet headers and a modification to all existing gateways to set this bit [14]. The end-users also need to
Within each congestion control mechanism there are two distinct components: feedback and control. As Jain has pointed out in his paper [20], control theory teaches that the control frequency should be equal to the feedback frequency. Control theory also says that no control policy can solve congestion that is shorter than its feedback delay. This later condition is especially important in designing a congestion control scheme for satellite networks because of the large propagation delay inherent in the system.

There have been many proposals and suggestions for controlling congestion in a frame relay network, but few of them are directly applicable when used over satellite links. Mitra has pointed out in his paper [27] that the effect of large delay-bandwidth product demands new analysis and design of congestion control mechanisms. However, it will still be helpful to look at some of these schemes because we can often derive insights from them and make appropriate modifications to suit the conditions and needs of a satellite network.
implement a filtering algorithm on which the decision to increase or decrease their window size is based.

Another well known dynamic window congestion control, which has received much attention since its publication and has since become a requirement in TCP implementation is elaborated in Van Jacobson's *Congestion Avoidance and Control* [14]. This scheme is really more similar to a congestion recovery scheme, because it operates just to the left of the cliff (see figure 8) as Jacobson has pointed out himself. The feedback mechanism used in this scheme is implicit, in that a packet dropped by a gateway due to buffer overflow will cause the corresponding user to time-out. Whenever a user time-outs, it always sets its window size to 1 packet before incrementing it according to the algorithm described below. With such a feedback scheme, the buffer occupancy level will almost certainly oscillate between nearly full and nearly empty, thus contributing to large delay and delay variability [7]. The rationale for this algorithm is that it does not need any modification to the existing gateway. As mentioned in section 3.4 of this thesis, there are some inherent disadvantages of such an implicit feedback mechanism when applied to satellite networks.

The end-user's response to the congestion feedback is described by the following algorithm:

- On time-out, record half the current window size in a variable `ssthresh` and set current window size, `cwnd`, to 1. *Ssthresh* is defined as the optimum window size at the time of congestion.

- On receiving acknowledgment for new data, sender does

  ```
  if (cwnd < ssthresh)
      /* if we're still doing slow start,
      open window exponentially */
      cwnd += 1 /* add 1 to cwnd */
  ```
else

    /* otherwise do congestion avoidance 
       by increment by 1 */
    cwnd += 1/cwnd  /* add 1/cwnd to cwnd */

The user thus opens its window exponentially (with respect to the round-trip delay) to what the algorithm thinks is a safe operating point (ssthresh), then congestion avoidance takes over and slowly increases the window to probe for more bandwidth as it becomes available on the path.

Much research and investigation has been done on this dynamic window algorithm, one of which calls for our attention. In [17], it is found that one peculiar effect of the slow-start mechanism is the synchronization of packet loss due to gateway buffer overflow. In other words, all transmitting virtual circuits on the same gateway lose exactly one packet each whenever the buffer is full. This results in all connections shutting down their window sizes to one packet at the same time. The paper also demonstrated through simulations that the buffer oscillates between empty and full in cyclical manner, as mentioned above. We believe that the phenomena of synchronized window shut-down combined with extreme buffer oscillation is detrimental to a satellite network because of the large delay-bandwidth product. Specifically, shutting down all window sizes to 1 packet will "drain" the pipe, causing link bandwidth to be underutilized. For a satellite link with very large pipe size, this is very inefficient. It is also believed that the extreme buffer oscillation may contribute to a large queueing delay in the network.

It is very important for us to look at the congestion control mechanisms proposed by CCITT and ANSI, as they define standards and guidelines for current and future implementations of frame relay networks. The recommended use of
FECN, BECN and CLLM by a frame relay network can be found in the document Addendum to T1.606 [5] and has been described above. In addition to the explicit notifications above, user's response to implicit notification is also suggested. The use of FECN is not considered in this thesis as it has been shown by some previous work [22, 30] that it is considerably ineffective compared to BECN. The relevant material for dynamic window congestion control proposed by CCITT can be found in section I.2.2 of the document Appendix I to Recommendation Q.922 [31] and is summarized below:

- on receiving a BECN, reduce window size to 0.625 of current window size. If $S$ consecutive frames are received with the BECN bit set, this reduction is repeated.
- on receiving any $S/2$ consecutive frames with the BECN bit clear, increase window size by 0.125 of user throughput.
- on detecting a frame loss, reduce window size to 0.25 of current window size.

In addition, the window size shall not be reduced to less than its throughput (the window equivalence of CIR) so as to maintain the guaranteed rate as negotiated at the PVC set up time. As we can see, the dynamic window algorithm proposed above is one instance from the set of multiplicative decrease/additive increase window algorithms. It has been shown in [13] that this type of algorithm leads to stabilization at fair window sizes for multiple connections. However, it is not clear how the precise values of the increase/decrease parameters in this algorithm were reached. Some effects of the decrease parameter are nevertheless discussed and demonstrated by Ramakrishnan and Jain in the same paper [13]. For example, by using a larger decrease parameter, fairness may be achieved more rapidly, but it also

---

9 $S$ is defined in [31] as the number of frames that can be transmitted in a round-trip time interval (i.e., one window turn).

10 Throughput is defined as $(CIR / frame size) \times round\text{-}trip\ propagation\ delay$. It is also being referred to as "pipe" size in the literature.
leads to larger amplitude oscillation around the maximally efficient window size. On the other hand, a smaller decrease parameter reduces oscillation around the optimal window size, but also prolongs the time it takes to achieve fairness. It is therefore, clearly, an issue of compromise and of relative importance between achieving fairness and reducing throughput degradation.

Several other papers we have studied proposed and discussed various alternatives for controlling congestion in a frame relay network. For example, in [29], a selective discard strategy is recommended, particularly under the condition of non-cooperating end users. This strategy can be implemented by utilizing the DE bit of the frame relay packet header and by placing access controllers at the network entry points to set these bits if necessary. However, it is our objective to avoid any frame discard altogether by using an efficient congestion avoidance scheme since all discarded frames, whether they are marked or not, will have to be retransmitted by the transport protocol.

In several related papers primarily published by Debasis Mitra [11, 27, 28], he proposed an innovative adaptive window congestion control based on round-trip delay time estimation. Both analytical and simulation results show that this innovative scheme works extremely well for data networks with high delay-bandwidth product. However, the algorithm implementation calls for a revolutionary design on the end-user protocol and it has not been implemented on any existing networks. As frame relay is intended to use currently available technologies and to support current applications, a decision against simulating this protocol in this project was reached. Some other alternatives of frame relay congestion control can also be found in references [4, 6].
4.3.2 Enhanced Congestion Control Algorithm

The congestion control algorithm proposed in this thesis consists of two distinct components: a feedback mechanism from the network and an end-user dynamic window adjustment algorithm. On the network side an utilization sampling algorithm coupled with the congestion notification mechanism is implemented on each link. A source-controlled, adaptive window algorithm is incorporated to the transport mechanism of each DLCI connection to dynamically control its traffic load based on the network conditions.

The utilization sampling algorithm itself is fairly simple. The link is scheduled to compute its utilization for a past duration equivalent to the satellite round-trip propagation delay every T_{util} sec (T_{util} < satellite round-trip propagation delay). In other words, the algorithm slides the sampling "window" along the positive time axis to measure link utilization at discrete time intervals. It is important to note that the choice of the utilization duration is not arbitrary; it essentially captures the effect of the previous window size on the link utilization. When the link utilization is computed, it is compared to a threshold value, U_{th}. If the computed value exceeds this threshold, the link is considered as congested or about to be congested. In this situation, a BECN message is created and sent to each traffic source that traverses that link (in effect, it is as though the network generates a CLLM message). The utilization threshold value is a simulation parameter. We consider 0.9 (or 90%) to be an appropriate value for U_{th}.

It is of vital importance to note here that the network algorithm mentioned above is only implemented on the satellite ingress node, i.e. the node adjacent to the source of traffic. In most situations, this is the only type of congestion that will
occur\textsuperscript{11}. However, when bandwidth can be dynamically allocated as in the BoD system, congestion may also occur at the egress node, i.e. the node adjacent to the destination of traffic\textsuperscript{12}. Congestion control at the egress node is a fairly complicated issue in a satellite environment, and clearly the same algorithm that works well enough for the ingress node congestion is not necessarily going to be effective for the egress node congestion.

For our simulations, when congestion occurs at the egress node, a BECN message is sent to the source of the frame that has most recently arrived at the node. By the law of probability, the source that is using a larger fraction of the bandwidth has a higher chance of getting "picked" as the source of congestion. After sending a BECN message, the algorithm waits for one satellite round-trip duration before it sends the next congestion message, if necessary (this is because it takes half a round-trip for the BECN message to get to the source and another half a round trip for the effect of the new window size to reach the egress node). We are, by no means, justifying that this is in fact the best mean of handling congestion at the egress node. This mechanism does seem to work reasonably well within the context of our simulations.

The transport protocol on the user side implements the dynamic window algorithm as described below:

- Each connection starts with a window size of $W = W_{\text{min}}$, which is the satellite round-trip "pipe" size assuming a transfer rate of the connection's CIR.

\textsuperscript{11}This is based on the assumption that the terrestrial access link capacity is much larger than the internodal satellite link capacity - a common scenario found in WANs (Wide Area Networks).

\textsuperscript{12}Our simulation results have shown this phenomena in the dynamic-bandwidth case, although the frame loss ratio is negligible.
• Upon receipt of an acknowledgment, increase current window size, \( W \), by \( \frac{k}{W} \), in which \( k \) is a constant, up to a maximum of \( W_{\text{max}} \), i.e., in pseudo-code: \( W += \frac{k}{W}, W < W_{\text{max}} \).

• Upon receipt of a BECN, reduce window size to a fraction \( Q (Q < 1) \) of current window size but not beyond \( W_{\text{min}} \), i.e.: \( W = Q \times W, W > W_{\text{min}} \).

• Upon a retransmission time-out, reduce window size to a fraction \( q (q < Q < 1) \) of current window size but not beyond \( W_{\text{min}} \), i.e.: \( W = q \times W, W > W_{\text{min}} \).

This algorithm is effectively similar to the one proposed by Appendix I to Recommendation Q.922 [31], except for some minor modifications. First of all, users do not have to implement the filtering algorithm for the BECNs they receive every round-trip to increase or decrease window size. The filtering function has been shifted to the network component of the congestion control by measuring the link utilization for every round-trip duration (hence there is no requirement for a user to maintain the counter variable \( S \) as described in [31]). Secondly, instead of increasing the window size by \( k \) packets after receipt of \( W \) acknowledgments, it is increased by \( \frac{k}{W} \) on receipt of every acknowledgment. Window size is still increased by \( k \), but the increase is fragmented into smaller size steps\(^{13}\). This modification has the effect of first, reducing the impulse load on the networks, and second, reducing the clustering of frames into individual connections, which has some adverse effect as discussed in [17]. The specific values for the parameters in this algorithm are presented and discussed in the simulation chapter.

4.3.3 Distributed Bandwidth Allocation Algorithm

When the capacity of a transmission link can be dynamically adjusted, congestion control can be viewed in a different manner. From queueing theory, we

\(^{13}\) A window of size \( W \) frames will generate at most \( W \) acknowledgments in one round-trip. Thus an increment of \( \frac{k}{W} \) per acknowledgment will increase the window by at most \( k \) frames in each round-trip.
know that if the average arrival rate to a queueing system is larger than the average
departure rate, the system is unstable. In data networks, an unstable queue often
causes its buffer to overflow as the steady state queue length is infinite (the extent of
the overflow of course depends on how long the queue remains unstable).

To prevent a queue from building up unboundedly, the conventional
method is to reduce the arrival rate to that queue. In practice, the traffic arrival rate
can be reduced by using window size or rate adjustment mechanisms. The service
rate of the queue cannot be increased to meet the stability condition because the link
capacity is fixed in a terrestrial network. However, if the link capacity can be
dynamically increased to keep up with the traffic arrival rate to the link, as in the
BoD system, the stability condition may be achieved without penalizing the users
(i.e. reducing their traffic rates). From the point of view of one link, the ability to
increase its link bandwidth dynamically creates a new possibility for performing
congestion control.

The bandwidth allocation algorithm we have designed for our simulations is
a simple FCFS (First Come First Served) distributed algorithm implemented on each
link in the network. Using the same utilization sampling algorithm as in our
congestion control scheme, the measured link utilization is compared to two
thresholds, Uthhigh and Uthlow (Uthhigh > Uthlow). If the measured utilization is
greater than Uthhigh, the link is considered as a qualified candidate to request more
capacity. The algorithm then sends a request signal to the NCC for an additional
capacity of $S_{inc}$ kbps.

However, when the measured utilization is found to be less than Uthlow, the
link capacity is considered underutilized and a capacity of $S_{inc}$ kbps is returned to the
NCC. Therefore, bandwidth is always acquired and released in terms of \( \text{Sinc} \) kbps simplex circuits. Each link maintains a static capacity of \( \text{Sinit} \) kbps throughout the simulations. Whenever the link capacity changes to a new value, its utilization for the next \( T_{\text{wait}} \) sec is measured before the next change can occur. The \( T_{\text{wait}} \) duration allows the algorithm to measure the link utilization based on the new capacity.

In this distributed bandwidth allocation algorithm, there is no inter-nodal communications. Nor does the NCC have any information on the other nodes. Each entity (links, NCC) knows only its own state (for the link, the state is the current link capacity and its utilization; for the NCC, it is the current amount of capacity that can be distributed). It is therefore conceivable that two or more requests may "collide", i.e. several links simultaneously needing more capacity, but only one of them will be successful in getting it. The "collision" scenario is especially probable when the network traffic is so heavy that there is not enough bandwidth to go around.

An important criteria in the design of our bandwidth allocation algorithm is that it should attempt to achieve the work-conserving principle on a network-wide basis. However, to adhere fully to this principle is not possible in practice because of the signaling delay involved in switching satellite circuits from one connection to another. Nevertheless, this algorithm does allow a greater degree of work-conservation compared to a fixed-bandwidth network, in that the network capacity can be distributed dynamically according to the traffic load on each link. It is this relative adherence to the work-conserving principle that the BoD network can be made to take advantage of by the distributed bandwidth allocation algorithm.

To further illustrate the concept of the distributed bandwidth allocation algorithm, we refer to Figure 9. Traffic streams from user 1 to 5 are multiplexed into
the corresponding inbound links. The algorithm is implemented on each of these node-to-node links. It monitors the link utilization, interacts with the NCC and adjusts the link capacity accordingly.

![Diagram of distributed bandwidth allocation model](image)

**Figure 9. Distributed bandwidth allocation model**

One question that has yet to be answered is whether the algorithm above distributes bandwidth fairly among the congested nodes. The answer depends on what the fairness criteria is. As our algorithm is such that there is no one entity in the network that has the state information of all the other entities, fairness as described in section 3.5 of this thesis cannot be achieved in a strict sense. In other words, the algorithm does allow two equally congested links to possess unequal capacities in steady state, though this occurrence is highly unlikely unless in extreme heavy load conditions.

Nevertheless, the algorithm can be considered fair using a less stringent criteria, that is, it distributes bandwidth in proportion to the link congestion level on
a FCFS basis. Given the simplicity of the distributed bandwidth allocation algorithm, we are satisfied with this fairness criteria\textsuperscript{14}.

### 4.3.4 Integration of Bandwidth Allocation and Congestion Control Algorithm

The integration of the distributed bandwidth allocation algorithm and the enhanced congestion control algorithm described in section 4.3.2 is a synthesis of two general classes of congestion control schemes: resource creation and demand reduction. Both schemes, while trying to achieve the same overall objective of upholding network performance in the face of congestion, can have disastrous effects if they are allowed to function independent of each other. Consider a situation where demand on a network is reduced and resources are added to the network in the same time interval. Newly added resources will most likely go underutilized because the initial increase in demand, which calls for the resource creation, has already been eliminated by the demand reduction scheme. The integration or coordination problem is further complicated by the large response time (between 600 to 1200 msec of round-trip signaling delay) of bandwidth switching in a BoD system. Nevertheless, we believe that the two apparently dichotomous algorithms we have developed can be integrated in such a way that they complement each other's effectiveness.

The problem posed by the large response time of bandwidth switching in a BoD system is this: if the congestion control algorithm is invoked during the signaling delay (a period of uncertainty; the distributed bandwidth allocation algorithm may or may not be able to obtain more capacity) and more capacity is

\textsuperscript{14}We were able to design a centralized control algorithm at a much later stage of this project, after much time had been spent on the integrated algorithm. As it is not within the original intention of this thesis, the centralized algorithm is not presented and discussed in detail in this chapter. It is included in the simulation chapter as an alternative to the distributed bandwidth allocation scheme.
available later, underutilization of capacity may occur as the window size of the users may have been shut down. On the other hand, if the users are allowed to increase their traffic freely within the round-trip response time, they may overflow the queue quickly. To avoid falling into either situation, a queue control algorithm is implemented in this state of uncertainty. If the queue size grows beyond a certain threshold, Qth, during this interval, BECN messages will be sent to the users to reduce their window sizes. Note that this mechanism is very different from the congestion control mechanism; it is based on queue length, instead of link utilization. Link utilization during this period may grow to 100% and yet users will not be throttled unless the queue threshold is violated. By allowing the queue to build up but not allowing it to overflow, additional bandwidth will be utilized more efficiently when it becomes available at the end of the signaling delay.

One problem with the stand-alone distributed bandwidth allocation algorithm is that it does not provide any remedy if the link capacity cannot be increased to meet its traffic load. To prevent the onset of congestion in this situation, traffic load has to be reduced by our congestion control scheme. This is where the integration of the two algorithms comes in. Our point of view is that the congestion control algorithm complements the distributed bandwidth allocation algorithm. When bandwidth allocation fails (i.e. when capacity cannot be increased on a link), congestion control is executed (i.e. demand on the link is reduced). Therefore, the integration of the two algorithms can be summarized as such: *when bandwidth allocation fails, exercise congestion control.*

4.4 Description of Simulation
The simulation program was developed using OPNET Modeler, a special-purpose communication network simulation package. OPNET Modeler is also a comprehensive package because it can be used throughout all phases of a simulation project, from development of the program to analysis of the simulation results. Briefly, we highlight several important and useful features of this package that justify its use in this thesis:

- All code is written as finite state machines and in a hierarchical, object-based environment. The object-based modeling structure not only facilitates distributed algorithm development, but also makes the task of debugging a large simulation program relatively easier.

- It allows multiple simulations to be run concurrently on several machines.

- It provides a user-friendly graphical interface which makes the initial learning curve of this package relatively fast. It also provides a direct interface to UNIX, which is more useful and convenient when the user becomes familiar with the package.

- It allows for re-usability of simulation software modules across different projects.

The simulation program was designed to take input describing network topology, traffic source characteristics, and all conceivable simulation parameters. Most input methods are either in the form of ASCII files or UNIX command lines. All simulation parameters can be changed without altering the source codes. This results in a more testable and reliable simulation.
5. Experimental Results

5.1 Chapter Organization

Results for some of the simulations we have performed are presented in this chapter. In each simulation, statistics such as normalized network throughput, average end-to-end frame delay, average file transfer rate and frame loss ratio are collected. In section 5.2, the definitions of simulation input and output statistics and the procedures used to measure them are presented. In section 5.3, we first compare the performance of a non-BoD satellite network with the performance of a BoD satellite network. The non-BoD network exercises the congestion control algorithm, while the BoD network exercises the integrated control algorithm as described above. The performance of various end-user congestion control schemes are also examined, with emphasis on their suitability to the BoD network. A simple centralized control algorithm for bandwidth allocation in the BoD network is then described and its performance is compared to that of the distributed algorithm. Some parameter sensitivity issues are studied in section 5.4. Finally, the statistical accuracy of our simulation results is verified in section 5.5.

5.2 Definitions of Simulation Input and Output Statistics

This section presents the definitions of the input and output statistics used in our simulations. In most cases, the input (or independent) statistics are the network loads. Output statistics such as the average file transfer rate, frame loss ratio and
average end-to-end frame delay are collected as the input statistics are varied in each set of simulations.

**Load** - The load of a DLCI is the average rate at which it would transmit data when unconstrained by network delay and congestion control. Network load is the sum total of the individual DLCI's load. In the simulation results, the load of a network is shown as normalized to the network capacity.

**Network capacity** - This is the maximum aggregate rate at which data can flow between all pairs of edge nodes in a network. In our simulations, it is set to the sum total of all the individual DLCI's CIR.

**File transfer duration** - The time interval between the arrival of the first frame of a file at the destination DLCI and the arrival of the last frame of the file at the destination DLCI.

**Average file transfer rate** - For each file transmitted in a simulation, its file size and transfer duration are recorded. At the end of the simulation, the average file transfer rate is obtained via dividing the sum total of the file sizes by the sum total of the file transfer durations.

**Average End-to-end frame delay** - This delay is measured from the moment a frame leaves the sending transport layer (for the network) to the moment the frame arrives at the receiving transport layer. Retransmission delay is not included in this measure.
Network Output - This is a normalized measure of how much data gets through a network in a simulation. It is therefore obtained via summing up the total amount of data received at the end of a simulation and normalizing this value to the maximum amount of data that can be transferred by the network in that simulation. The maximum amount of data that can be transferred by a network is defined as the product of the network capacity and the simulation duration.

Frame loss ratio - this is obtained via dividing the number of frames discarded by the network by the total number of frames received in a simulation.

5.3 Comparison of Networks with Different Control Schemes and Parameters

5.3.1 Performance of a BoD Network vs a Non-BoD Network

A series of simulation runs were conducted to examine the performance of a BoD satellite network with the integrated control scheme relative to the performance of a non-BoD satellite network with the congestion control scheme. The independent parameter in these simulations is network load, which we vary from 0.2 to 0.95. Network performance in terms of average file transfer rate, average end-to-end frame delay and normalized network output (throughput) are examined.

The networks that we simulated are pictured in Figure 10. Network II is a full-mesh network with 1.544 Mbps terrestrial access link and 576 kbps inter-nodal satellite links. All capacities are fixed in this network. Network I, on the other hand, has the same fixed access link speed except that the inter-nodal links have an initial capacity, $S_{\text{init}}$, of 192 kbps. The Network Control Center in Network I manages a bandwidth pool of $72 \times 64$ kbps ($= 4.608$ Mbps), which makes the total capacity of the two networks equal. Throughout the simulations, the link capacities in Network I
change dynamically according to the traffic load, but the integrated control algorithm
ensures that they do not fall below the initial link capacity, $S_{init}$, at any time. Both
networks share the same traffic configuration. Each network node is connected to 3
terminals. Within each terminal, there are 9 PVCs. Therefore, there are altogether
108 traffic sources. Each source generates frames with CIR of load level $\times 64$ kbps.
The rest of the parameter values are set to the following values:

**traffic parameters:**
- file size = uniformly dist. between $[1.0, 2.0]$ Mbytes
- frame size = deterministic, 128 kbytes
- traffic mix = 100 % file transfer

**window parameters**$^{15}$:
- initial window size, $W_{min}$ = 37.5
- max window size, $W_{max}$ = 1024.0
- window increase, $k$ = 18.75
- window decrease, $q$ = 0.875

**network parameters:**
- queue size = 64 kbytes / 500 frames
- queue threshold$^{16}$, $Q_{th}$ = 32 kbytes / 250 frames
- utilization period, $T_{util}$ = 100 msec
- $T_{wait}$ (for BoD) = 600 msec
- additional capacity, $S_{inc}$ = 64 kbps

**link utilization thresholds:**
- congestion, $U_{th}$ = 0.9
- bandwidth inc., $U_{thhigh}$ = 0.8
- bandwidth dec., $U_{thlow}$ = 0.7

---

$^{15}$ We feel that some justification of our choice for the window parameter values is necessary. $W_{min}$ is set
to 37.5, which is the pipe size for a user CIR of 64 kbps. The value for $k$ is double of what is being suggested
by Q.922 [31], for the obvious reason that a more rapid increase strategy is required to take advantage of
the huge satellite pipe size [18]. We choose a $q$ value of 0.875 for the same reason. This value of $q$ also
helps to reduce the large amplitude swing in the window size, of which we have observed in our
simulations when the Q.922 default value of 0.625 was used.

$^{16}$ Note that the threshold value is half of the queue size, which, as Nagle pointed out in [23], appears to be
a reasonable engineering decision. The value of the queue size itself is somewhat arbitrary.
The average file transfer rate is shown in Figure 11(i). This result is consistent with our intuition. Network capacity is better utilized when it can be dynamically distributed according to traffic load. This is particularly true when the load is low. For files of size 1.5 Mbytes on average, bandwidth from an idle link can be switched to a busy link and consequently yields better overall throughput performance. However, as the idle period scales down with the increasing load, so does the file transfer performance. This is supported by the diminishing vertical distance between the two curves.

End-to-end frame delay is maintained at a low value throughout the load range in Network II. This is consistent with the fact that there is very little queueing at the node buffers because whenever the link utilizations reach 0.9 (90%), congestion messages are broadcasted to end-users to reduce window sizes. There is a larger queueing delay for Network I. This queueing delay is incurred mainly because the queue control algorithm allows the buffer to build up to the specified threshold.
Qth during the signaling delay period. This delay, however, is not significantly higher than that of Network II considering that we are simulating file transfer application, which presents itself as a type of highly bursty traffic to the network. The delay can also be reduced if a frame-tagging leaky-bucket scheme is used in conjunction with a priority service discipline.\(^\text{17}\)

Figure 11(iii) shows the normalized output of the two networks in comparison. Both networks are able to achieve an output level very close to the input load throughout the load range. It is interesting to note that for Network II, the output level tops at 0.9, even though the input load is 0.95. This is consistent with the fact that the congestion control algorithm always maintains the link utilization at or below the threshold, 0.9 and this result serves as a quick check to the correctness of the algorithm.

---

\(^{17}\)File transfer typically will exceed its CIR due to its burstiness. Some of its frames can then be tagged—the DE bit can be set to 1. These tagged frames can be given a lower priority by the queue server, as long as their sequential order is preserved.
Figure 11 (i, ii). Performance of Network I vs Network II.
5.3.2 "Aggressive" Parameters vs Q.922 Default Parameters

The window parameters as described in Appendix I of Recommendation Q.922 are used in this set of simulations. As shown by the results in Figure 12, the BoD network (Network I) also yields better performance in terms of the average file transfer rate compared to the non-BoD network (Network II). Average end-to-end frame delay is, as expected, slightly higher for the BoD network (especially at high load). This is again the result of a larger queueing delay in the node buffers.

Figure 13 combines the simulation results from Figure 11 and Figure 12, so that the effects of the window parameters on the network performance can be compared. As expected, by using more "aggressive" window parameters (i.e. smaller decrease factor and larger increase factor), we are able to achieve better network-wide throughput performance (Figure 13a(iii) and 13b(iii)) as well as average file transfer...
rate (figure 13a(i) and 13b(i)). It is important to note that while overall throughput performance is improved in both networks, the average frame delay does not scale up significantly with the "aggressive" window parameters. Even though figure 13a(ii) appears to be the contrary, the magnitude of difference is only in the order of 0.1 ~ 0.15 seconds, which is hardly noticeable from the user's perspective.

It can also be observed from figure 13a(iii) and 13b(iii) that at a very high load, the "aggressive" window parameters are able to "extract" more throughput out of the two networks compared to the default window parameters. This result could be due to the fact that with the default parameters, each window size is increased more slowly and reduced by a larger fraction every time a BECN is received. Even at such a high load the combined window size of the active connections is still less than the large satellite pipe size. In addition, the larger amplitude swing of the window size due to the decrease factor (0.625) could very possibly contribute to this throughput degradation.
Average End-to-end Frame Delay

![Graph showing average end-to-end frame delay for Network I and II with BoD and Non-BoD parameters.]

Network Output versus Load

![Graph showing network output versus load for Network I and II with BoD and Non-BoD parameters.]

Figure 12 (ii, iii). Performance of Network I vs Network II using Q.922 default parameters
Figure 13a (i-iii). Comparison of "aggressive" and Q.922 default parameters for Network I
Figure 13b (i-iii). Comparison of "aggressive" and Q.922 default parameters for Network II
5.3.3 TCP on the BoD Network and the Non-BoD Network

In this section, the performance of the Transmission Control Protocol (TCP) on the BoD network and the non-BoD network is examined. The TCP code that is used in the simulations corresponds very closely to the 4.4 BSD TCP code. Specifically, the fast retransmit algorithm, the slow-start algorithm and the congestion avoidance algorithm are implemented. The slow-start algorithm and the congestion avoidance algorithm are presented and discussed in section 4.3.1 of this thesis. The code related to TCP connection set-up, keep-alive, and close is not implemented here as it is not relevant to our simulations.

Unlike the previous simulations, the integrated bandwidth allocation and congestion control algorithm is not implemented in the BoD network. This is because the TCP code implemented here does not respond to BECN. Instead, the stand-alone distributed bandwidth allocation algorithm is used, and it works independently with the congestion control algorithm in TCP.

As shown by figure 14 (i)-(iii), the BoD network is superior in terms of all three performance measures. Particularly, the end-to-end delay demonstrated by the BoD network is shorter than that of the non-BoD network. This may be due to the fact that as capacity of each transmission link can be dynamically increased, arriving frames are able to be "flushed" out of the queues more rapidly. The frame loss ratio depicted in figure 14 (iii) consists of frames that are dropped both at the ingress nodes and the egress nodes. Frame loss cannot be prevented as it is the only congestion feedback mechanism that is available in TCP.

Note that even though the TCP congestion control algorithm is not coupled with the bandwidth allocation algorithm, it is still able to detect the change in the
link bandwidth because of the "clocking" mechanism. As link bandwidth increases, frames are "clocked" out of the system by the acknowledgments more rapidly. Since each acknowledgment enlarges a TCP window (either exponentially or linearly), the excess bandwidth can be detected and utilized. In this sense, the TCP congestion control algorithm and the distributed bandwidth allocation algorithm do behave somewhat symbiotically; to what extent this symbiotic relationship goes is an interesting question to be answered by future research.

A cross comparison of the simulation results presented here and those presented in section 5.3.1 and 5.3.2 also yields some interesting general observations. It seems that some form of open-looped explicit congestion notification scheme (i.e. BECN in our context) is necessary for a satellite network to operate effectively. As discussed earlier, the shutting down of window size to 1 by TCP combined with the large delay-bandwidth product results in the inefficient utilization of link bandwidth. Furthermore, it also results in extreme buffer oscillation in the network, causing the average end-to-end delay incurred by the traffic streams to be higher.
Figure 14 (i, ii). Performance of TCP on Network I and Network II
5.3.4 Exponential Window Increase Strategy on the BoD Network

With a much higher amount of bandwidth available to a particular connection due to the network-wide statistical multiplexing in the BoD network (especially at light load situations), the possible advantage of using a more aggressive window increase algorithm cannot be overlooked. We investigated one such algorithm. Instead of increasing the window linearly with respect to the round-trip delay, an exponential increase similar to Van Jacobson's slow-start phase window increase strategy was used. In this algorithm, window size is doubled every round-trip if there is no congestion. The congestion control algorithm we used in this section is essentially the same as the one described in section 4.3.2 except for one line of code: \( W += k/W \) is replaced by \( W += 1 \) which effectively changes the window increase strategy from linear to exponential.
Figure 15 shows the network performance of the resulting simulations. The exponential window increase strategy is further coupled with two different bandwidth increase strategies. In one case, bandwidth is allocated on a 64 kbps (S\textsubscript{inc} = 64 kbps) discrete rate while in the other case, 128 kbps (S\textsubscript{inc} = 128 kbps). The simulation results show that while exponential window increase strategy leads to better file transfer performance\textsuperscript{18}, it also results in a much larger frame delay and frame loss ratio. In fact, the frame loss ratio for the two cases of exponential window increase strategy is at least two orders of magnitude larger than that of the linear window increase strategy (refer to Table 1) that the latter appears to be zero throughout the load range. It is also observed that the exponential window increase strategy is more effective in increasing the file transfer rate if the bandwidth can be allocated at a higher rate.

\textbf{Average File Transfer Rate}

![Average File Transfer Rate Graph]

\textit{Figure 15(i). Performance of exponential window increase strategy on Network I}

\textsuperscript{18}The throughput performance being shown here will be degraded severely by the large number of retransmissions if a selective repeat strategy is not used. Therefore, the advantage of such a scheme may not be significant at all.
Figure 15 (ii, iii). Performance of exponential window increase strategy on Network I
<table>
<thead>
<tr>
<th></th>
<th>load = 0.2</th>
<th>load = 0.6</th>
<th>load = 0.75</th>
<th>load = 0.85</th>
<th>load = 0.95</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategy II</td>
<td>2.620e-3</td>
<td>5.117e-3</td>
<td>6.011e-3</td>
<td>5.445e-3</td>
<td>5.650e-3</td>
</tr>
<tr>
<td>Strategy III</td>
<td>0.000e+0</td>
<td>3.558e-6</td>
<td>1.483e-5</td>
<td>7.150e-6</td>
<td>1.687e-5</td>
</tr>
</tbody>
</table>

Note: Strategy I - Exponential, 128 kbps  
Strategy II - Exponential, 64 kbps  
Strategy III - Linear, 64 kbps

Table 1. Frame loss ratio corresponding to Figure 15 (iii)

5.3.5 Centralized Control on the BoD Network

In this section, we explore a simple centralized control algorithm for the bandwidth allocation in Network I. As mentioned before, the distributed algorithm is based on the individual link's utilization and does not take into consideration the utilization of the network as a whole. In contrast, the centralized algorithm is such that every link in the network reports its utilization to the NCC periodically, therefore the NCC has the state information of the entire network. The NCC then distributes the bandwidth according to the computational algorithm we are going to describe below. Each link in the network exercises congestion control as in the Non-BoD case. Therefore, the centralized bandwidth allocation algorithm is independent of the congestion control algorithm and they are not integrated as in the previous case.

The centralized algorithm consists of three steps of computation:

- Step 1 - Multiply current utilization of each link with the corresponding link bandwidth; this gives us the actual capacity that is being utilized by the traffic on that link. If this calculation yields an actual capacity less than $S_{\text{init}}$, set actual capacity to $S_{\text{init}}$.

- step 2 - By summing up the actual capacity of all the links, network bandwidth can be distributed according to the following formula:
distributed capacity = (actual capacity / total actual capacity) \times \text{total network bandwidth}^{19}.

• step 3 - the distributed capacity obtained in step 2 may not be an integer. However, bandwidth is distributed in terms of 64 kbps simplex circuits. Therefore, in performing step 2, all numbers obtained are truncated to the lower integer value. This truncation yields some "left-over" capacity, which is distributed in this step to the links with the largest pre-truncated distributed capacity values.

In this algorithm, the entire network bandwidth is always distributed by the NCC regardless of the traffic load. However, it is distributed according to the weighted utilization, rather than the utilization based on a single link as in the distributed algorithm. In the simulations, each link reports its utilization to the NCC every 600 msec, as it has been observed that any period lower than 600 msec leads to fluctuation in the distribution of bandwidth by the NCC.

Figure 16 compares the network performance of the centralized-control BoD network with the performance of the original distributed-control BoD network. Referring to figure 16 (i), we see that the average file transfer rate of the centralized-control network is better than that of the distributed-control network when the traffic load is low. However, when the load is increased to more than 0.7, the reverse scenario takes place. Overall, the two types of BoD network still have considerably better file transfer performance than the Non-BoD network, even though the centralized controlled network seems to scale closer to the Non-BoD network at high load.

Figure 16(ii) compares the average frame delay across the three networks. We observed that the centralized-control BoD network is able to demonstrate a very consistent end-to-end delay independent of the load level - even slightly more

---

19Total network bandwidth is in terms of number of 64 kbps simplex circuits.
consistent than the Non-BoD case. The reason that the end-to-end delay is kept low throughout the load range is obvious - the congestion control algorithm keeps the utilization of each link in the network to 90%, which maintains a small queue length even when the traffic load is high. The fact that it is more consistent than the Non-BoD network may be explained by the adaptivity of the BoD network to dynamically varying load, which does not exist in the Non-BoD network.

The file transfer performance of the centralized-controlled BoD network is quite interesting - one wonders why there is a cross-over point when the input traffic load is increased from 0.6 to 0.75. This phenomena may be related to the fact that the average frame delay is maintained at the same level by the centralized-controlled network while the distributed-controlled network demonstrated increasing delay at high load. It appears to be a case of the classical trade-off between throughput and delay, where in one case one is able to achieve higher throughput with a longer delay and vice-versa. The centralized control problem is a very interesting one; what we have seen is only a very preliminary study of the problem. It would serve as a good topic for future investigation.
Average File Transfer Rate

Average End-to-end Frame Delay

Figure 16. Performance of Network I with centralized and distributed control algorithm
5.4 Parameter Sensitivity Analysis

5.4.1 Utilization Threshold in Bandwidth Control Algorithm

In order to demonstrate the effect of different utilization threshold values for the distributed bandwidth allocation algorithm, the experiment in section 5.2.1 was repeated with the values of utilization threshold set to the following:

utilization threshold:
Uthhigh = 0.7
Uthlow = 0.6

With this set of parameters, each link triggers for more bandwidth when its current utilization hits 70% (instead of 80% as before). Similarly, it releases some bandwidth when its current utilization drops to 60% (instead of 70% as before). Although we could have run these simulations with vastly different parameter values, those values would not have make any logical sense. For example, a Uthhigh of 0.5 will result in each link triggering for more bandwidth when it is only 50% utilized, and may result in inefficient utilization of network bandwidth.

With a small tuning of these two parameters, the results we see in the following figures come as no surprise. The average file transfer rate, average frame delay and overall network throughput for the second set of threshold values show a little inferiority to that of the first set. We believe that the network resource utilization is slightly inefficient by using the second set of parameter values, due to the "pre-mature" request for more bandwidth and the prolonged hogging of underutilized network capacity.
Figure 17 (i, ii). Performance of Network I with different utilization thresholds
Figure 17 (iii). Performance of Network I with different utilization thresholds

5.4.2 Maximum Link Capacity

The maximum link capacity is an optional parameter in the distributed bandwidth allocation algorithm which specifies a ceiling for the link capacity. In the earlier simulations, this parameter is not specified, i.e. each link could potentially acquire as much capacity as it needs (up to a maximum of pool capacity, 4.608 Mbps). By imposing this limit on the link capacity, one would imagine that it would result in a more equal distribution of network resources, as no entity in the network can keep too much bandwidth to itself. However, the simulation results shows that there is not much advantage to be gained by imposing this optional parameter in the distributed bandwidth allocation algorithm.

As figure 18(i) and (ii) shows, the differences among the three curves are insignificant, except for the average file transfer rate when the input traffic load is
0.2. We believe that the results observed can be explained by the following reasoning: when the traffic load is low, a transmission link with file transfer traffic is very likely to be successful in getting additional bandwidth from the NCC, as the probability of "collision" is low. However, due to the average file size in these simulations, it is most likely that before the link capacity reaches the maximum allowed, the file transfer is complete. Since the interarrival time between each file is very large at light load situations, the transmission link will be idle at this time. Consequently it releases the additional bandwidth obtained during the file transfer until its link capacity is once again at its initial value, $S_{\text{init}}$. Therefore, the maximum link capacity imposition is, at best, non-existent in the simulations. We actually did some calculations which proved that with the simulated file size, this is in fact the case for both parameter values in these simulations.

Even in the heavy load conditions, we still do not see any significant differences in the network performance. However, there is a different explanation for these observations. As file transfers arrive more frequently, they are back-logged at the transport layer buffers due to the congestion control. Therefore, the resulting traffic profile may look like a continuous stream of frames, with no discernible "idle" time between each file. The situation in which the transmission link capacity falls back to $S_{\text{init}}$ would not occur in this case. However, as every transmission link in the network will attempt to acquire more bandwidth in the heavy load situations, no one link is likely to obtain more bandwidth than the other. Therefore, the maximum capacity limit parameter once again fails to have any effect on the network performance.
Figure 18. Performance of Network I with various link capacity limits
5.4.3 Transmission Queue Buffer Size

To investigate the effect of buffer size on network performance, Experiments described in section 5.2.1 were repeated with two other buffer values, 96 kbytes (750 frames) and 32 kbytes (250 frames). It is shown that variation in the transmission buffer size does not have any significant effect on the file transfer performance. However, the average frame delay does increase with the buffer size. Figure 19(iii) shows the ratio of frames that are dropped due to buffer overflow at the egress queues (as there is no buffer overflow at the ingress queues for all the simulations we have conducted in this project, except the TCP simulations). There is absolute-zero frame loss across the load range when the buffer size is 96 kbytes. An interesting observation is that the frame loss ratio actually tops at 0.75 load. The traffic load beyond this value is probably so heavy that hardly any inter-nodal link capacity could be large enough to overflow the egress node buffers.

The frame loss ratio for the non-BoD network is not depicted in figure 19(iii) because it is zero for all three buffer sizes throughout the load range. As link capacity is fixed in this network, and the congestion control algorithm always maintains the link utilization at or below 90%, it is hardly surprising that the queue buffer does not overflow. Though the egress queue length is expected to increase momentarily due to the round-trip propagation delay incurred in throttling traffic sources, all the buffer sizes used in these simulations are large enough to accommodate this transient behavior.
Figure 19 (i, ii). Performance of Network I with various queue buffer sizes
5.4.4 Percentage of Traffic Mix (File Transfer and Interactive)

The traffic scenarios in all the previous simulations are wholly dominated by the file transfer traffic type. Even though this is quite a typical application for frame relay networks, it is still imperative for us to investigate the situation in which a variety of traffic types are present in a network. For this purpose, we reran some of the simulations with both file transfer and interactive traffic models. In these simulations, the percentage of interactive traffic in the network is varied and the corresponding network performance is obtained.

The two networks in section 5.2.1 are simulated at a traffic load of 0.5 and 0.85 with percentage of interactive traffic ranging from 0% (all file transfer) to 100% (all interactive). Three general observations can be derived from figure 20 and figure 21 below. First, the network performance is insensitive to the type of traffic going into
the non-BoD network. Second, the BoD network consistently performs better in terms of the average file transfer rate, though its characteristics at the two different load levels are, interestingly enough, somewhat different. Third, the average frame delay for the BoD network converges to that of the non-BoD network as the percentage of interactive traffic is increased. At 100% interactive traffic, the average frame delay for both networks are virtually identical.

In our models, the interactive traffic type is much less bursty than the file transfer traffic type. The interactive traffic is further "smoothed" out at high load when many active connections are multiplexed into the transmission links simultaneously. The Law of Large Numbers tells us that when a number of bursty traffic streams are multiplexed into a single stream, the resulting stream becomes less bursty than the individual streams. The effect of this traffic "smoothing" is that the bandwidth distribution of the BoD network stays fixed in the steady state, and most likely the steady state capacity of each satellite link is the same as the fixed capacity in the Non-BoD network - which would explain the convergence of the end-to-end delay for both Network I and Network II at 100% interactive traffic.
Average File Transfer Rate

Average End-to-end Frame Delay

Figure 20. Performance of Network I and Network II with various traffic mixes at 0.85 load
Figure 21. Performance of Network I and Network II with various traffic mixes at 0.5 load
5.4.5 NCC Signaling Delay

The NCC round-trip signaling delay used for all previous simulations is equal to 600 msec, which is the round-trip propagation delay for satellite links. However, there are some situations where more than one round-trip is required for the signaling mechanism, such as when two satellite nodes desire to execute some form of "hand-shaking" protocols before transmitting data on the newly acquired circuit for reliability. For this reason, the BoD network simulations in section 5.2.1 were repeated with the NCC signaling delay set to a more conservative value of 1200 msec. We see in Figure 22 (i) that the effect of this change is not significant on the average file transfer rate. On the other hand, it results in an increase of the average frame delay.

The increase in the average frame delay due to longer NCC signaling time is a reasonable consequence of the integrated algorithm. When the NCC signaling delay is longer, a larger proportion of frames in the network would be queued at the satellite transmission links. Note that the actual queueing delay of each frame is still under-control (because the transmission queue length is not allowed to grow beyond $Q_{th}$). It is the increase in the proportion of frames that are queued that results in the higher average value.

We also ran the same simulations with the call setup time equal to 100 msec, which corresponds to a situation in which the signaling to the NCC is via terrestrial links. The terrestrial signaling option is not currently available with the BoD system, but is a possible implementation in the future. The average file transfer rate in this case demonstrates an interesting behavior. When the traffic load is at 0.2, its
performance is better than the other two cases. However, as the load is increased, the reverse scenario takes place.

This phenomena can be explained by the following argument. When the network load is high, the amount of network-wide multiplexing performed by the distributed bandwidth allocation algorithm is reduced because most transmission link capacities in the network are heavily utilized. The persistent demand for more bandwidth by each transmission link cannot be met. Consequently, the congestion control algorithm takes over more frequently to maintain the link utilizations at or below 90%. Therefore, we observe a lower file transfer rate for the case of terrestrial NCC connectivity. The more frequent invocation of congestion control also results in a smaller queueing delay. The above explanation is supported by a larger number of BECN messages generated by the network and a smaller transmission queue lengths recorded in the simulations.

![Average File Transfer Rate](image)

**Figure 22 (i). Performance of Network I with different signaling delays**
5.5 Statistical Accuracy

In each of the simulations that were conducted to obtain the results presented in the previous sections, a random seed number was specified to generate a specific traffic scenario. In order to make the comparison of the network performances valid, we have chosen the same random seed number for each of those simulations. However, as different random seed numbers generate different traffic scenarios, the network performances for these traffic scenarios may be different also. Nevertheless, if the performances do not vary greatly, then data from one simulation path list can be safely assumed to be representative of the typical network performance. For this reason, the Non-BoD simulations described in section 5.3.1 were executed with 4 different random seed numbers and the corresponding average file transfer rates are shown in figure 23 below.
The confidence interval for the sample values at each load level was not computed because the sample size is too small and there is no reason to assume that these sample values are normally distributed. However, it can be observed from the graph that the file transfer rates at each load level falls within reasonable range. Therefore, we can conclude that a single random seed number is sufficient to generate the network performance for each simulation.
6. Conclusions

In this thesis, we study various congestion control mechanisms available to a satellite frame relay network. In particular, the suitability of satellite networks, with a dynamic bandwidth switching capability, for providing efficient frame relay services is investigated. In conjunction with this study, two bandwidth allocation algorithms are developed. In one case, a distributed algorithm was designed and integrated with an enhanced congestion control algorithm based on CCITT Recommendation Q.922. In the other case, a centralized bandwidth allocation algorithm, independent of the congestion control algorithm used by the end-users, was designed. The performance of two specific satellite networks with different control algorithms was examined via simulations. The simulation results show that both bandwidth allocation algorithms are able to achieve a much better file transfer rate, while maintaining the network transit delay (i.e. the average frame delay) in close proximity to that demonstrated by the stand-alone congestion control algorithm. Several more specific conclusions can further be drawn from the various simulations performed under different control schemes and parameters:

- The file transfer rate and the network throughput can be improved in a satellite network by using more "aggressive" Q.922 window parameters, without compromising the network transit delay. More specifically, by using a larger additive increase and smaller multiplicative decrease, the large "pipe" size of a satellite link can be better utilized by the end-users.
• The performance of the BoD network, in terms of average file transfer rate and average frame delay, seems to be relatively unaffected by the different types of traffic entering the network. Specifically, when interactive traffic is mixed with the file transfer traffic in various percentages at different load levels, the average file transfer rate is consistently superior while the average frame delay is still maintained at close proximity to that of the Non-BoD network. This result implies that there is a definite advantage provided by the BoD network for all types of users.

• Although an exponential increase window strategy yields better file transfer performance in the BoD network, it also increases the average frame delay and frame loss ratio by several fold. This strategy, though provides better performance to a few file transfer users, is not suitable when the network consists of interactive traffic which demands a short response time. The interactive users will typically suffer a degraded performance due to the prolonged average frame delay caused by large queueing at transmission links and retransmission of dropped frames.

• Evaluation of TCP performance in both networks yields some interesting conclusions. While the BoD network is superior to the Non-BoD network, from the perspective of various network performance measures, it is also observed that an appropriate open-loop explicit congestion notification scheme (for e.g., BECN coupled with end-user congestion control) allows a satellite network to operate more efficiently compared to an implicit/closed-loop congestion
notification scheme (as in the time-out mechanism due to frame loss in TCP).

- The BoD network performance is rather insensitive to some of the network parameters\textsuperscript{20} - for e.g., utilization thresholds and maximum link capacity. However, we should be cautious that the relative insensitivity to these parameters may be due to a specific network scenario. For instance, the average file size used in our simulations is one such scenario that results in the insensitivity to the maximum link capacity parameter (cf. section 5.3.2).

- The file transfer rate in the BoD network does not seem to be affected by the signaling delay to the NCC. However, if a more conservative value for the signaling delay is used, the resulting average frame delay is larger. On the other hand, if signaling to the NCC can be performed via terrestrial links (a possibility in the future), the average frame delay is kept low but the file transfer rate is degraded in heavy load conditions.

- While the advantages of the BoD network over the Non-BoD network from various perspectives are clear, the integrated control algorithm we proposed for the BoD network does have a "side-effect": the possibility of egress node buffer overflow. Although we have seen no such-occurrence in the Non-BoD network, it has been observed to occur to a small degree in the BoD network. However, this

\textsuperscript{20} This is true provided that the parameters are varied within a reasonable and sensible range. For example, setting the maximum link capacity to a very low value will undoubtedly yield vastly different results. However, the choice of this value may not be justifiable in the first place.
phenomena does not affect the other aspects of the network performance.

- A brief investigation into the issue of distributed control versus centralized control in the BoD network yields some interesting conclusions: if the network is designed to operate at or below 70% utilization on average, the centralized-control network excels in all aspects compared to the distributed-control network, assuming equal total capacity for both. In particular, the centralized-control network is able to maintain a very low average frame delay throughout the load range. This implies that if the user traffic is very sensitive to delay, the centralized control network is a better candidate for such traffic. However, it must be pointed out that the centralized bandwidth allocation algorithm we designed always distributes the total network bandwidth to the network transmission links (compared to the distributed algorithm, in which the NCC allocates bandwidth on a per link basis). If the network capacities were to be shared with other systems, the centralized bandwidth allocation algorithm may have to be enhanced to meet such needs.

The centralized-control BoD network seems to demonstrate some desirable behavior. The excess bandwidth at low traffic load conditions is efficiently utilized by the network to "speed" up individual file transfer performances. Furthermore, the average frame delay is kept low even under heavy traffic situations. The algorithm we have developed for the centralized-control network also has the added advantage that it can be implemented independently the end-user congestion control protocol. As such, it is an approach worth pursuing in the future.
Appendix

A.1 Description of the Control Algorithms in Finite State Machine

In order to present our algorithms in a more detailed manner, we include the finite state machine diagrams that implement those algorithms in this appendix. The corresponding pseudo-code further illustrates the specific actions performed by the algorithm at each state transition.

A.1.1 Network Algorithm for Congestion Control (as described in 4.3.2)

![Finite state machine of the network congestion control algorithm.]

In cong_ctrl state:
- at utilization timer interrupt::
  measure link utilization;
  if (link utilization > Uth)
    send BECN;
    go to sleep state;
  else
    stay in cong_ctrl state;

In sleep state:
- at utilization timer interrupt::
  measure link utilization;
  stay in sleep state;
at sleep timer interrupt::
go to cong_ctrl state;

A.1.2 Network Algorithm for Integrated Bandwidth Allocation and Congestion Control (as described in 4.3.3. and 4.3.4)

Explaination:
Event A - utilization timer interrupt
Event B - control message arrival from NCC
avail/not_avail - bandwidth available/not available
normal - within specified range of utilization
over_utilized - over-utilizing the link
under_utilized - under-utilizing the link

Figure 25. Finite state machine of the integrated bandwidth control and congestion control algorithm
In bc_cc state:
   at utilization timer interrupt::
       measure link utilization;
       if (utilization > Uthhigh & cur_svc < max_svc)
           send bandwidth request message to NCC;
           go to wait_acquire state;
       else if (utilization*cur_svc/pre_svc < Uthlow & cur_svc > min_svc)
           send bandwidth release message to NCC;
           cur_svc -= inc_svc; /* service rate is reduced immediately */
           go to wait_release state;
       else if (utilization > Uth)
           send BECN;
           go to sleep state;
       else
           stay in bc_cc state;

In wait_acquire state:
   at utilization timer interrupt::
       measure link utilization;
       if (cur_qsize > Qth)
           send BECN;
           stay in wait_acquire state;
   at NCC message arrival::
       if (connection request granted by NCC)
           cur_svc += inc_svc;
           go to sleep state;
       else /* connection not available */
           /* do congestion control */
           if (utilization > Uth)
               send BECN;
               go to sleep state;
       else
           go to bc_cc state;

In wait_release state:
   at utilization timer interrupt::
       measure link utilization;
       if (cur_qsize > Qth)
           send BECN;
           stay in wait_release state;
   at NCC message arrival::
       /* disconnection request is always granted */
       go to sleep state;

In sleep state:
   at utilization timer interrupt::
       measure link utilization;
       if (cur_qsize > Qth)
A.2 Method of Biasing the File Transfer Traffic to the Specified CIR by the Law of Large Numbers

One important application of frame relay is file transfer, which presents itself to the network as a type of highly bursty traffic. In our model, it is represented as an impulse, modeling the moment when the user actually invokes a file transfer application. Therefore, our file transfer traffic simulator can be represented by the diagram below:

![Diagram of file transfer traffic simulator](image)

* vertical arrow in the graph represents a file transfer invocation with file size $L_i$ at that particular time

Figure 26. File transfer traffic simulator

In order to bias the file transfer traffic towards the user-specified CIR, the long term average traffic rate has to be equal to CIR. To have the CIR built into the file transfer traffic simulator, we make use of the Strong Law of Large Numbers (SLLN):
Theorem: Let $S_n = X_1 + X_2 + \ldots + X_n$ where $X_1, X_2, \ldots$ are Independently and Identically Distributed (IID) random variable (r.v) with finite mean $\mu$. Then with probability equal to 1,

$$\lim_{n \to \infty} \frac{S_n}{n} = \mu$$

Before we go on any further, let us define the following quantities:

- $L_i$: Size of $i$th file to be transmitted
- $X_i$: duration of $i$th file inter-arrival time
- $G_i$: average traffic generation rate of $i$ files

We also assume that $L_i$ and $X_i$ are IID r.v and independent of each other. This assumption will be true when we assign independent probability distributions to generate values for $L_i$ and $X_i$. Then, by simple arithmetic,

- total size of $n$ files $= \sum_{i=1}^{n} L_i$
- total inter-arrival time of $n$ files $= \sum_{i=1}^{n} X_i$
- average traffic rate of $n$ files, $G_n = \frac{\text{total size}}{\text{total time}} = \frac{\sum_{i=1}^{n} L_i}{\sum_{i=1}^{n} X_i}$

$$= \frac{\sum_{i=1}^{n} L_i/n}{\sum_{i=1}^{n} X_i/n} \quad \text{(E3)}$$
By taking the limit of the numerator and the denominator of (E3) as \( n \to \infty \), we obtain the average traffic generation rate, \( G \):

\[
G = \frac{E[L]}{E[X]} \quad \text{by SLLN}
\]

If we assign the average traffic generation rate to the CIR, we get

\[
\frac{E[L]}{E[X]} = \text{CIR} \\
E[X] = \frac{E[L]}{\text{CIR}}
\]

Assuming CIR is given, we can independently select \( E[L] \) or \( E[X] \), but not both. We select \( E[L] \) as the independent parameter because that gives us the freedom to vary the mean file size, which is more important in our simulations. It is also assumed that the file size is uniformly distributed and the file inter-arrival time is exponentially distributed with their respective expected values, \( E[L] \) and \( E[X] \).

A.3 Numerical Verification of Some Simulation Results

In this section, two of our simulation results are verified via simple algebraic calculation and standard queuing theory analysis. The purpose of these computations is to provide a partial check on the correctness of our simulation models. We first compute the average file transfer rate for the BoD network by solving a quadratic equation. Then we compute the queueing delay and the average queue length of a non-BoD network by using the standard M/D/1 queueing model. Both computation results come sufficiently close to the results obtained via simulations.
The first computation is done on a simulation corresponding to a load of 0.2 in the BoD network described in section 5.2.1. This simulation is chosen because at such low load, each file transfer is separated by a large inter-arrival time. Therefore, we can reasonably assume that the probability of file transfer back-logging by the transport layer protocol is very low. This assumption is important to our computation of the average file transfer rate, as it allows us to consequently assume that the network bandwidth is always available to a transmission link (i.e. it does not have to compete with the others during its busy cycle).

We also make use of the following simulation parameters in our computation:

- average file size = 1500 kbytes
- \( W_{\text{min}} \) = 4.8 kbytes (37.5 frames of 128 bytes each)
- \( k \) = 2.4 kbytes (18.75 frames of 128 bytes each)
- satellite round-trip delay = 0.6 sec
- initial link capacity, \( S_{\text{init}} \) = 192 kbps

It is important to note that with the assumption of network bandwidth availability, link capacity increases by 64 kbps every two round trips during the file transfer. The enhanced window congestion control algorithm also ensures that a window size increases by at most \( k \) every round-trip (for a derivation of this claim, refer to [17]). By using the maximum, \( k \), in our calculation, we are deriving the upper bound of the file transfer rate. Our computation is derived from Table 2, which demonstrates the window size and the amount of data sent by one connection in each round-trip:
Table 2. Window size and amount of data sent as a function of round-trip time

If we sum up the third column vertically, the total must equal the size of the file. Therefore, by assuming an average file size of 1500 kbytes, we obtain the following equations:

\[
4.8 + 4.8 + 2.4 + 4.8 + 2(2.4) + \ldots + 4.8 + M(2.4) = \text{avg. file size} \\
4.8(M + 1) + 2.4(1 + 2 + \ldots + M) = 1500
\]

Simplifying, we have a quadratic equation:

\[
1.2M^2 + 6M - 1495.2 = 0
\]

Taking the positive root, we obtain:

\[
M = 32.88 \\
= 33
\]

Therefore,

\[
\text{file transfer delay} = (33 + 1) \times 0.6 \text{ sec} \\
= 20.4 \text{ sec} \\
\text{file transfer rate} = 1500 \times 8 / 20.4 \text{ kbps} \\
= 588.235 \text{ kbps}
\]

This computed value provides the upper bound for its corresponding simulation result, as we assume the ideal conditions above. The simulation result
yields an average file transfer rate of 543 kbps over 500 files, which is reasonably close to the best performance value we obtained above.

Our next computation is performed on the simulation scenario depicted in section 5.2.1 in which we calculate the average network queueing delay and average queue length. We particularly selected the simulation of the Non-BoD network with 100% interactive traffic so that we can use the following properties:

- frame arrival can be modeled by a Poisson process.
- link capacity is fixed at 576 kbps.
- service rate is constant, due to fixed link capacity and frame size.
- the Poisson traffic arrival stream is not affected by its corresponding window size when the load is low, as in this case (load = 0.5).

With the above properties, we can model each queue in the network as a M/D/1 queue. Therefore, the average delay incurred by a frame from the network ingress point to the network egress point is:

\[
\text{average delay} = T_1 + T_2 + T_3 + \text{Propagation Delay}^{21}
\]

where

\[
T_1 = \text{delay at network ingress link}
\]

\[
T_2 = \text{delay at network intermediate link}
\]

\[
T_3 = \text{delay at network egress link}
\]

and

\[
T = \frac{1}{\mu} + \frac{\rho}{2\mu(1-\rho)} \quad \text{M/D/1 formula}
\]

From the simulation parameter values, we obtained the following:

---

21Due to our routing function, traffic arriving to a node consists of about 70% of cross traffic. This randomization process is considered sufficient to allow us to assume that each queue in tandem is independent of each other and the Poisson arrival process in each queue is relatively close to the physical reality.
\[ g_2 = \frac{576}{1.024} \text{ frames/sec} \]
\[ \rho_2 = \frac{\lambda_2}{\mu_2} = 0.5 \]
\[ \lambda_2 = \rho_2 \mu_2 = 281.25 \text{ frames/sec} \]
\[ p_1 = g_3 = \frac{1544}{1.024} \text{ frames/sec} \]
\[ \rho_1 = \frac{\lambda_3}{\mu_3} \]
\[ \lambda_3 = \lambda_2 \]
\[ \rho_1 = \rho_3 = 0.18653 \]

Substituting these values into \( T_1, T_2 \) and \( T_3 \), we get the following result:

average delay = \( 7.3924 \times 10^{-4} + 2.6669 \times 10^{-3} + 7.3924 \times 10^{-4} + 0.3 \text{ sec} \)
\[ = 0.304145 \text{ sec} \]
\[ = 0.3041 \text{ sec} \]

We can also compute the average queue length (including the frame in service) for the intermediate link buffer:

\[ N_2 = \lambda_2 T_2 = (281.25)(2.6669 \times 10^{-3}) = 0.75 \text{ frame} = 1 \text{ frame} \]

Table 3 below shows the results of our calculation and simulation respectively:

<table>
<thead>
<tr>
<th></th>
<th>calculation</th>
<th>simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg. network delay (sec)</td>
<td>0.304145</td>
<td>0.304081</td>
</tr>
<tr>
<td>avg. queue size, ( N_2 ) (frm)</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3. Comparison of calculation and simulation results.
References


