Distributed Bandwidth Regulation Mechanisms for Multiple Traffic Classes in Wide Area Networks

A Thesis

Presented to

The Faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment
of the Requirements for the Degree

Master of Science
in
Computer Science

by

Debapriya Sarkar
August 1995

Approval Sheet

This dissertation is submitted in partial fulfillment of the
requirements for the degree of
Master of Science (Computer Science)

Debapriya Sarkar

This dissertation has been read and approved by the Examining Committee:

Jörg Liebeherr, Advisor

Alfred C. Weaver, Chairman

Alan P. Batson

Accepted for the School of Engineering and Applied Science:

Dean, School of Engineering and Applied Science

August 1995

Abstract

The introduction of multimedia traffic, which is both sensitive to delays, and is bandwidth-intensive, has greatly increased the complexities of bandwidth allocation in computer networks. Existing flow and congestion control techniques, which were designed to provide best-effort service to mainly data traffic, have been shown to be unsuitable in a network where traffic flows with widely different service requirements have to coexist. New traffic control mechanism are urgently needed that can isolate the different classes of network traffic, and can regulate bandwidth according to predetermined policies. In this thesis, we propose a pair of bandwidth regulation schemes that can regulate traffic flows belonging to different classes. The first mechanism, designed for use in a traditional packet-switched network, performs bandwidth regulation at two levels, distributing the total link bandwidth among, first, traffic classes, and second, individual flows within each traffic class. The second mechanism has been designed for use in an ATM network, which supports both connectionless as well as connection-oriented traffic. It performs bandwidth allocation among multiple levels, first allocating bandwidth among connectionless and connection-oriented traffic, then apportioning the bandwidth allocated to connectionless traffic among the connectionless traffic classes, and finally among individual traffic flows. We have developed protocols to implement each of these mechanism in a distributed manner, with minimum overhead. Simulation experiments have been performed to demonstrate the effectiveness of the protocols.

Contents

Abstract **iii**

1 Introduction **1**

1.1 Resource Regulation on Internetworks 6

1.2 Resource Regulation in B–ISDN 8

1.3 Thesis Outline 13

2 Literature Survey **15**

2.1 Bandwidth Regulation Mechanisms for Internetworks 15

2.1.1 Rate-based Scheduling Disciplines 16

2.1.2 Link-Sharing Mechanisms 18

2.1.3 Bottleneck Flow Control Mechanism 19

2.2 Bandwidth Regulation Mechanisms for ATM Networks 20

2.2.1 Burst-level Bandwidth Reservation 21

2.2.2 Window-based Bandwidth Regulation 22

2.2.3 Bandwidth Advertising and Renegotiation 23

2.2.4 Bottleneck Bandwidth Regulation 23

2.3 Available-Bit-Rate Traffic and Bandwidth Regulation Mechanisms 24

2.3.1 Credit-based FCVC Proposal 25

2.3.2 Closed-Loop Rate-based Proposal 26

3 Theoretical Results **28**

3.1 Bandwidth Allocations with Intra-class and Inter-class Fairness 29

3.2 Bandwidth Allocations with Multi-level Regulation 40

3.3 The Single Class Case 44

4 Experimental Results **54**

4.1 A Bandwidth Regulation Protocol for Packet-switched Networks 54

4.1.1 The p–protocol 55

4.1.2 Simulation Experiment 59

4.2 A Multi-level Bandwidth Regulation Protocol for ATM Networks 64

4.2.1 The c–protocol 64

4.2.2 Simulation Experiment 69

5 Conclusions **79**

5.1 Summary of Contributions 80

5.2 Future Work 81

BIBLIOGRAPHY **83**

List of Tables

4.1 Flow Parameters. 60
4.2 Class Guarantees. 72
4.3 Flow Parameters. 72

List of Figures

1.1 Flows and Flow Classes on a Network Link. 7

1.2 Approaches to Supporting Connectionless Traffic on ATM Networks. 9

1.3 Multi-level Bandwidth Regulation. 12

2.1 Throughput Restriction Due to a Bottleneck Link. 19

3.1 Internetwork Model. 29

3.2 Intra-Class Fairness in a Network with Two Links. 33

3.3 ATM Network Supporting Connectionless Traffic. 41

4.1 Simulated Network. 59

4.2 Simulation Results. 63

4.3 Update Intervals. 65

4.4 Simulated Network. 70

4.5 Short-term Regulation. 73

4.6 Medium-term Regulation. 74

4.7 Long-term Regulation. 78

Chapter 1

Introduction

Computer networks were originally implemented to permit the efficient utilization of distributed computing resources and the dissemination of research information. The Internet, for example, was developed in the early seventies to facilitate the collaboration between universities, the military, and defense contractors. The protocols used by the Internet, the Transmission Control Protocol/Internet Protocol (TCP/IP) suite, were developed for the efficient transport of data traffic. The applications that were expected to use the network were text-based, such as remote login, file transfer, and electronic mail. The typical traffic pattern generated by these applications consisted of short periods of high activity (bursts) separated by lengthy inactive periods. They required a reliable end-to-end transport mechanism. These applications, however, did not have very strict requirements on the delay experienced by the traffic.

The choice of a suitable communication paradigm for the Internet was based on the characteristics of the traffic it was expected to support. The Internet designers adopted a *store-and-forward*, packet-switched network paradigm for the Internet. Data transfer took place in *connectionless* mode, in which a source and a destination do not have to engage in an explicit connection establishment procedure before data can be transferred. This was in contrast to the *connection-oriented* mode used in *circuit-switched* networks, (e.g., the telephone network), in which the source and a destination had to establish an end-to-end connection before initiating data transfer. In this thesis, we will refer to an unidirectional end-to-end traffic stream from a source to a destination, traveling over one or more network links, as a *traffic flow* or simply a *flow*. In the connectionless data transfer paradigm, when a user wishes to transmit a message, it simply submits the message to the network, without performing any connection initialization. If the size of the message is greater than the maximum packet size allowed by the network, the message is split into one or more packets. As no connections have been established, the network has to insert the full address of the destination into each packet. The packets are then independently routed through the network to the destination, where they are reassembled if necessary. The message is then delivered to the destination application. Unlike circuit-switched networks, there is no explicit reservation of bandwidth or other network resources for any end-to-end traffic stream between a source and a destination. Instead, the network provides a so-called *best-effort* service, in which network resources are shared among all traffic streams on an on-demand basis. The network only guarantees that all data packets will be correctly delivered to the destination. However, no guarantees are made as to the delay experienced by the packets, nor to the differences in the delay encountered by individual packets (jitter). The lack of timing guarantees was not a hindrance to data applications, because, as noted before, such applications are relatively insensitive to timing constraints.

The paradigm of the Internet was adequate as long as the characteristics of the traffic was comprised, for the most part, of bursty data traffic. However, this situation began to change in the late eighties. The simultaneous development of digital audio and video applications such as digital telephony and compressed video and the development of audio and video hardware for computers led to the development of a new class of network applications, the so-called multimedia applications. Examples of such applications include audio- and video-conferencing tools and the World-Wide Web.

The requirements of multimedia traffic are fundamentally different from that of data traffic. Multimedia applications must provide the appearance of a smooth continuous stream of data. For instance, video on demand applications should provide thirty video frames per second at the destination, otherwise the video will appear jerky. In audio-conferencing, if packets do not arrive at the destination at regular intervals, then there would be unnatural pauses in the conversation. In order to satisfy these timing constraints, multimedia applications impose an upper bound on the transmission delay that each frame can experience.

There must also be an upper bound on the variation of the delay experienced by individual packets. These timing constraints are far more stringent than those on data applications. On the other hand, multimedia applications are more tolerant of errors than data applications. Minor errors within a frame are often undetectable by users. Even the loss of an entire frame may be handled by extrapolation techniques. Therefore, multimedia flows can afford to sacrifice reliability in order to ensure timely deliver of packets. This is in sharp contrast to data applications, which must tolerate delays (e.g., due to retransmission) in order to ensure reliable data transfer.

When this new type of traffic was introduced into existing computer networks, the limitations of the best-effort model became apparent. In order to ensure the service levels demanded by multimedia applications, the network needs to dedicate a certain bandwidth to each multimedia traffic stream. The network also must be prepared to deny access to new traffic streams if doing so would violate existing commitments. However, the existing network paradigm, which provides a best-effort service, does not support either resource reservation or admission control. A multimedia traffic stream on such a network could be easily disrupted by a second multimedia traffic stream or a sudden burst of data traffic. Thus a new mechanism is needed to support multimedia traffic.

A great deal of research has been carried out in order to determine a suitable paradigm for supporting multimedia applications in computer networks. In the rest of this chapter we briefly highlight the main approaches in this discussion. A fuller treatment of the various solutions that have been proposed in the literature is presented in Chapter 2.

1. *Add more bandwidth*: The main cause of the disruptions in multimedia service is the contention for insufficient network bandwidth between multiple flows. If sufficient resources are always available to satisfy the requirements of the network traffic then it would be possible to support multimedia traffic on existing networks. However, this solution does not scale and becomes impractical with the rise in multimedia traffic levels. Furthermore, it has been observed that even with the availability of sufficient resources, the burstiness of data traffic can cause unexpected service deviations for

multimedia traffic.

2. *Allow resource reservation with admission control*: In this approach, each new flow has to specify its requirements to the network prior to initiating transmission. Admission control functions determine whether the network has sufficient resources to support the new flow. The flow will be accepted only if the network can provide the resources needed to support the service requirements of the flow *without* degrading the service commitments made to pre-existing flows. Resource reservation by the network will guarantee the strict requirements of multimedia flows regarding delay, jitter and error rate. However, this approach has a number of problems. The main difficulty is that resource reservation is incompatible with the existing view of the network as a shared resource. Admission control implies that users may be denied access to the network, which is contrary to the shared nature of the network.

3. *Regulate resources without admission control*: This approach attempts to improve the network's ability to cope with the requirements of real-time applications, but maintains the notion of the network as a shared resource. A main advantage of resource regulation schemes over admission control based reservation schemes is that they preserve the existing paradigm of viewing an internetwork as a shared resource. However, due to the absence of admission control, resource regulation schemes have strict limitations. Since the number of flows in the network is not restricted, the service received by individual flows may degrade arbitrarily.

   In general, resource regulation schemes do not dedicate resources to individual flows. Rather, the network enforces policies to distribute available resources to the flows. Resource regulation can be enforced on individual flows or on sets of flows. In this thesis, we define a set of traffic flows that have some common characteristics, e.g., the type of application, the protocol used, the service requirements, or the location of the traffic source, as a *traffic class* or a *flow class* [34]. In flow class regulation schemes, the network controls the bandwidth allocation to flow classes as a whole, as distinct from individual flows [10, 35]. A different policy for resource regulation is to enforce fairness conditions for all flows in the network [20, 43]. Ideally, however, a resource regulation mechanism should simultaneously enforce policies for both flow classes and individual flows.

4. *Develop a new network paradigm*: As we have highlighted before, existing computer networks are not designed either to support the high data transmission rates or the quality of service that new multimedia applications require. The first and third approaches do not address this issue, but concentrate on providing the best possible service over existing networks. Adopting the second approach also has serious disadvantages. A key point that all the three approaches have missed is that data traffic and multimedia traffic have fundamentally different characteristics and must be treated differently by the network. The best solution would be to have the network provide connectionless, best-effort service to data traffic, and provide connection-oriented service with delay guarantees to multimedia traffic.

   The goal of the proposed *Broadband — Integrated Service Digital Network* (B–ISDN) is to develop a high-speed network that would be capable of carrying data, voice, video, and images. It attempts to integrate all telecommunication services, viz. telephone, facsimile, video and data traffic [17]. A B–ISDN network is capable of very high-speed data access rates (155 Mb/s). In addition, the B–ISDN network allows different classes of traffic to be treated separately.

Of the four approaches presented so far, we discard the first approach, that of adding more bandwidth, because it is impractical and highly expensive to increase the bandwidth of a network on a regular basis. Adopting the resource reservation approach implies that admission control mechanisms have to be implemented in the network. This would cause a conflict with existing applications, which view the network as a shared resource. If admission control was enforced on on all traffic flows, existing applications would have to be modified. On the other hand, if existing best-effort applications are allowed free access to the network, then the resource reservation mechanism would be unable to provide guarantees on the quality of service to flows that requested guaranteed service. The resource regulation approach attempts to provide relative service guarantees without enforcing admission control. It avoids the incompatibility problem by maintaining the notion of the network as a shared resource. In §1.1, we discuss the resource regulation approach in more detail. The last approach, the B–ISDN approach, is fundamentally different from the first three approaches. Unlike the previous three approaches, it does not provide all classes of traffic with the same type of service. Rather, it attempts to provide each class of traffic, with the service that best matches the requirements of the traffic class. The details of this approach are discussed in §1.2.

1.1 Resource Regulation on Internetworks

A resource regulation mechanism does not reserve network resources for individual flows, and hence can not provide explicit performance guarantees. Instead, it regulates the use of scarce resources by the flows, so that all flows receive a fair share of the resource. We have chosen to concentrate on regulating bandwidth because it is the scarcest resource in present-day computer networks. Lack of adequate bandwidth is the cause of problems such as delay and jitter variations and packet losses due to buffer overflows. We present a bandwidth regulation mechanism that regulates bandwidth among flows classes and among individual flows belonging to a particular flow class. The objectives of our approach are to implement specific policies to distribute available bandwidth between flow classes (*inter-class regulation*) and between flows from the same class (*intra-class regulation*). The policies for bandwidth regulation that are considered in this study are as follows:

- *Inter-class fairness* — If at any point of time, the flows of a flow class do not fully utilize the current guarantee, the unused bandwidth is divided among all flow classes which can utilize the bandwidth for transmission.

- *Intra-class fairness* — For each flow class, a so-called *share* at a link provides the maximum link bandwidth available to each flow from this class. The maximum end-to-end throughput of a flow is limited by the link with the smallest share on the path

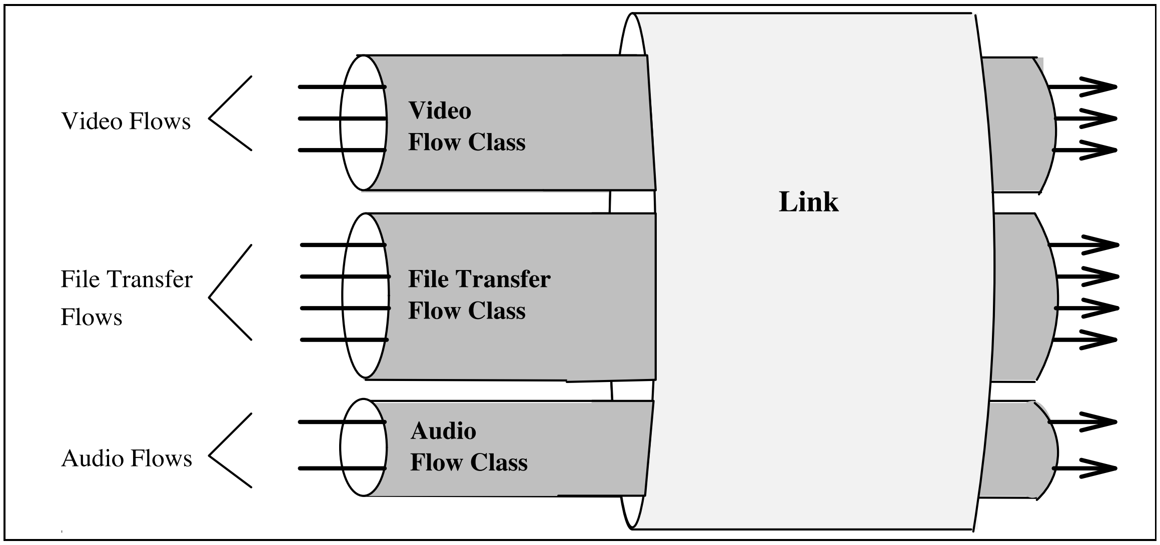


Figure 1.1: Flows and Flow Classes on a Network Link.

of the flow.

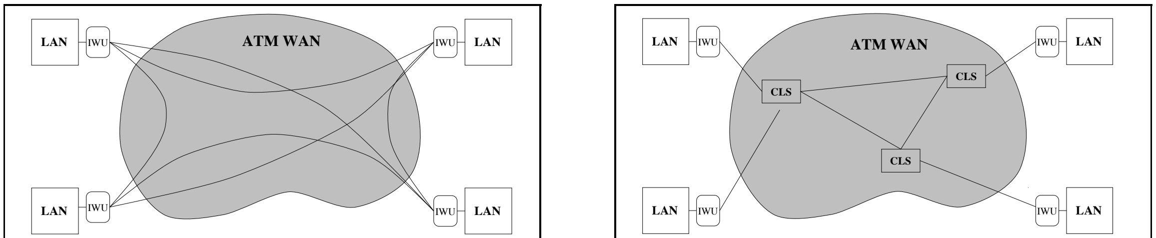
In Figure 1.1 we illustrate the relation between flows, shown as arrows, and flow classes, shown as pipes, for a single link. Inter-class regulation is concerned with allocating link bandwidth to the flow classes, i.e., video, file transfer, and audio flow classes in Figure 1.1. Intra-class regulation is concerned with distributing bandwidth within a single flow class. For example, for the video flow class, intra-class regulation determines the fraction of video-class bandwidth that is made available to a single video flow.

Our solution recognizes the fact that network traffic now consists of components with widely different requirements and attempts to treat these components separately. By providing regulation mechanisms at the both the intra-class and inter-class levels, it prevents flows with large traffic rates from throttling flows with low traffic rates. Consider a situation in which a video flow with a load of 10Mb/s and an audio flow with a load of 64 Kb/s both try to access a link of capacity 10Mb/s. Under the existing FIFO mechanism, the video flow would overwhelm the link and severely restrict the throughput of the audio flow, as both would be contending for the same bandwidth. In our approach, inter-class regulation would prevent the video flow class from obtaining the entire bandwidth of the link. The video class can only obtain the bandwidth not being used by the audio flow class. The video flow will therefore be restricted to the bandwidth not being used by the audio flow. This would allow the much smaller audio flow to obtain its requirement. Even in the situation where the two flows belong to the same traffic class, intra-class fairness will ensure that both flows receive a fair share of the bandwidth.

1.2 Resource Regulation in B–ISDN

The B–ISDN network standards were developed by the ITU-T in response to the emerging needs for high-speed communications and enabling technologies to support new services in an integrated fashion [17]. It is designed to carry data, voice, images, and video at very high speeds ranging from 155Mb/s to the so-called gigabit speeds. In 1988, the ITU-T, as a part of its efforts to standardize B–ISDN, chose Asynchronous Transfer Mode (ATM) as the multiplexing and switching technique for B–ISDN networks. ATM allows transmission of fixed-sized packets, called *cells*, at speeds upto several gigabits per second. In contrast to existing data networks, in which data transfers take place in a *connectionless mode*, ATM only allows *connection-oriented* data transfer. A *virtual connection* (VC) must be set up between a source and a destination before initiating data transfer. ATM also provides the concept of a *virtual path* (VP) to manipulate a set of VCs as a single channel.

ATM networks are expected to be deployed in two phases [23]. In the first phase, ATM is used to form private local area networks (LANs) or as a private backbone to LANs in a customer premise network. In the second phase, ATM LANs are interconnected to form ATM Wide Area Networks (WANs). For a smooth transition to the new technology, it is vital that ATM supports existing data communications applications. In other words, ATM networks must provide a mechanism to support connectionless traffic generated in traditional LANs and MANs. At present, there are several different approaches for providing connectionless service in ATM networks [1, 3, 8, 11, 25, 29]. However, most work in this area conform to either of the two general models outlined by the ITU-T – the *Indirect* Model and the *Direct Model* [16].



(a) Indirect Model (b) Direct Model

Figure 1.2: Approaches to Supporting Connectionless Traffic on ATM Networks.

- *Indirect Model:*

  In the *Indirect Model*, the ATM network provides no specific service to connectionless traffic. The *Inter-Working Units* (IWUs) connecting LANs to the ATM network have to direct the connectionless traffic to the appropriate destination across the ATM network. In order to carry out this function, the source IWU would have to set up connections to the destination IWU, allocate adequate bandwidth for the connection, and also implement flow and congestion control policies. As LAN packets are typically several orders of magnitude larger than an ATM cell, the IWU would also have to segment the packets originating in the LANs into ATM cells. Similarly, the destination IWU will have to reassemble the cells back into a packet before it can be forwarded to the destination system. The connections between the source and destination IWUs may either be set up permanently or on demand. If permanent or semi-permanent connections are established, then the overheads for data transfer will be negligible. But maintenance of the connections would become a problem as the number of IWUs increases. On-demand connection setup can avoid the maintenance problem, but it would lead to significant overhead in data transfer, as a connection would have to be established for each packet that is to be transmitted.

- *Direct Model:*

  In the *Direct Model*, the ATM network supports connectionless service by means of *Connectionless Servers* (CLS), which are entities within the ATM network providing

Connectionless Service Functions (CLSF) [18]. Host systems, which may either be individual computers or local area networks, need to maintain a fixed or semi-permanent connection to any particular CLS, via an IWU. The CLSs themselves are linked together by VPs, forming a virtual overlay network on top of the ATM network. In this scheme, the source IWU will first segment the packet into a sequence of ATM cells. It then transmits the ATM cells to its CLS, irrespective of the actual destination of the packet. The CLS will reassemble each packet, determine its destination and route it to a neighboring CLS. In this manner, the packet is segmented and reassembled and routed by successive CLSs until it reaches its destination IWU. This scheme has the advantage of not requiring a completely connected mesh of VPs to link a given set of IWUs. However, the processing of a packet at an intermediate CLS involves two complex stages. First, the packet has to be reassembled from ATM cells, before the CLS can determine its destination, and second, after its route has been determined, the packet has to be segmented back into ATM cells before it can be transmitted. This extra overhead tends to reduce the efficiency of the network.

Various schemes, all based on the above two models have been proposed in the literature. The CLS-based approach has been proposed as an implementation for the *Switched Multimegabit Data Service* (SMDS) [4, 8]. The indirect model has been found more suitable for ATM LANs [29]. In [25], a virtual overlay network is implemented on top of an ATM network to support connectionless services. The bandwidth allocated to the VPs forming the virtual network are dynamically adjusted based on the traffic demands. A comprehensive discussion of the issues involved in designing an architecture for connectionless data service in a public ATM network may be found in [1].

Any network protocol that implements either of the two models outlines above will have to address the issue of bandwidth allocation to connection-oriented and connectionless traffic. Obviously, a static allocation of bandwidth to VPs carrying connectionless traffic will result in wastage of bandwidth during periods of low intensity of connectionless traffic, and in congestion during periods of high traffic intensity. A better alternative is to vary the bandwidth allocation dynamically, using the actual connectionless traffic load as a heuristic for determining the bandwidth allocation. However, even if sufficient bandwidth is allocated to connectionless traffic, one still has to address the problem of distributing the bandwidth to individual connectionless traffic flows. Treating all connectionless flows equally is not a good solution, because, as noted previously, flows can have widely different service requirements, and consequently, require different bandwidth allocation policies. A better solution would be to allocate bandwidth in two stages. In the first stage, bandwidth would be allocated to different traffic classes. In the second stage, the bandwidth allocated to a traffic class would be divided among individual traffic flows. Thus, a bandwidth regulation mechanism that has to manage different types of connectionless traffic flows should have the capability to adapt to changing traffic loads at multiple levels. We therefore propose a multi-level bandwidth regulation with three levels: *long-term regulation*, *medium-term regulation*, and *short-term regulation*.

- *Long-term regulation (or VP regulation)* is concerned with the allocation of bandwidth to VPs that carry connectionless traffic. At the end of a VP update interval, the bandwidth allocated to a VP with connectionless traffic is increased if the utilization of the VP exceeds a threshold value and there is bandwidth available on the link. The bandwidth allocation is decreased if the utilization falls below the threshold or if there is a high demand from connection-oriented traffic.

- *Medium-term regulation (or class regulation)* distributes the bandwidth allocated to connectionless traffic among flow classes, i.e., a set of traffic flows possessing similar characteristics and service requirements. For instance, all e-mail traffic could comprise a single flow class, and traffic related to distributed processing could constitute another flow class. The division of bandwidth among flow classes consists of providing bandwidth guarantees. The guarantees are not fixed but dynamically adjusted based on the newly introduced concept of *class level fairness*. Classes which cannot utilize their full guarantee have their guarantees reduced. The bandwidth obtained is divided proportionately among classes which can utilize the extra guarantee. It is to be noted that long-term and medium-term regulation have no notion of individual traffic flows and only regulate aggregations of

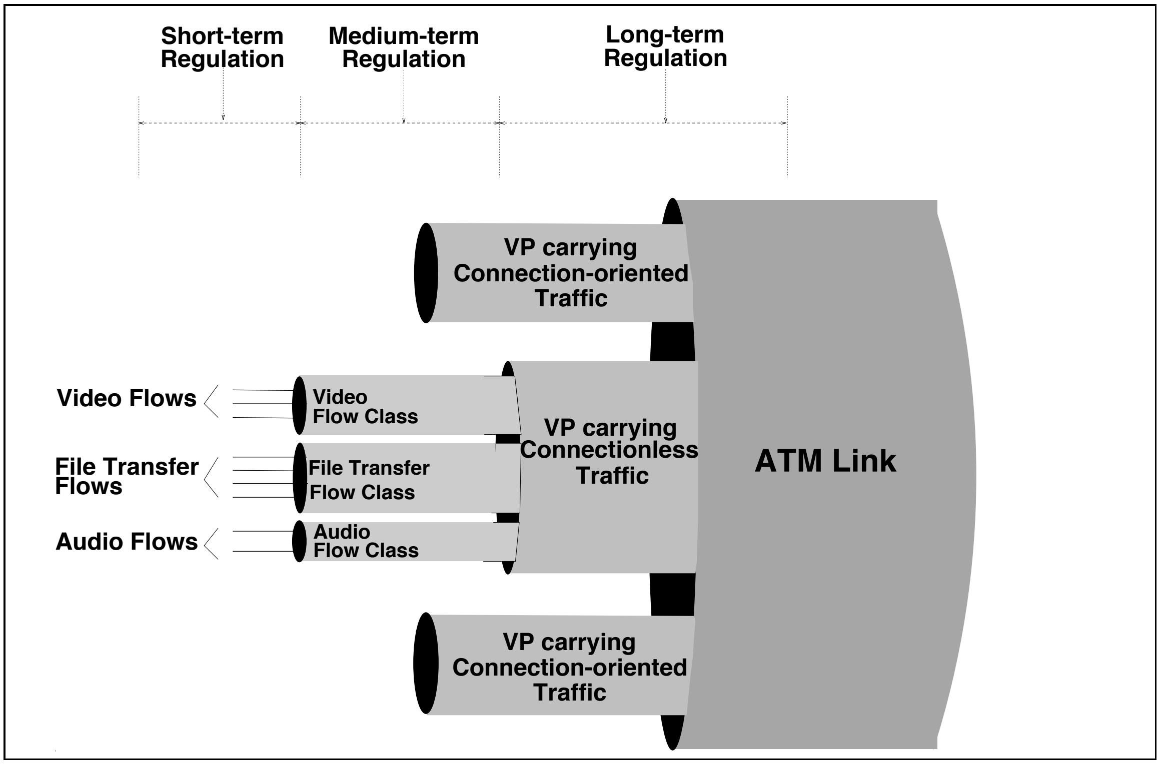


Figure 1.3: Multi-level Bandwidth Regulation.

traffic flows, i.e., entire flow classes.

- For *short-term regulation (or flow regulation)*, which involves the allocation of bandwidth to individual flows, we use the idea of *inter-class fairness* and *intra-class fairness* that were discussed in the context of bandwidth regulation for internetworks.

In Figure 1.3 we illustrate the three levels at which bandwidth regulation is performed. The bandwidth at the ATM link is divided among VPs carrying connectionless and connection-oriented traffic. This stage of the allocation is governed by long-term regulation. The bandwidth allocated to the VP carrying connectionless traffic is then divided among connectionless flow classes, such as video, file transfer, and audio flow classes, by medium-term regulation. Finally, the bandwidth allocated to each flow class is divided among flows belonging to that class. This stage of bandwidth allocation is determined by short-term regulation.

The multi-level regulation addresses two separate but related problems. The first problem, to regulate the amount of bandwidth being allocated to the connectionless traffic flows belonging to the different traffic classes, is similar to the bandwidth regulation problem addressed in §1.1. The solution we have proposed, i.e., the short-term regulation mechanism, is similar to the solution proposed in §1.1. While short-term regulation is primarily concerned with regulating bandwidth allocation to individual flows, we also take advantage of the virtual path mechanism of the ATM network to regulate the bandwidth guarantees made to the flow classes (medium-term regulation). This allows the network to respond to more long-term changes in the behavior of the traffic. This is in contrast to the solution proposed in §1.1, where the class guarantees are fixed. The second problem, which is unique to ATM networks, is the problem of regulating the bandwidth being allocated to connectionless and connection-oriented traffic. This problem is addressed by the long-term regulation mechanism.

1.3 Thesis Outline

The remainder of the thesis is outlined as follows. In Chapter 2, we discuss the various solutions that have been proposed in the literature. Earlier regulation mechanisms focussed on the problem of regulating bandwidth on internetworks. A popular technique is the use of scheduling disciplines at routers to regulate bandwidth utilization. We consider several different types of scheduling disciplines and study their relative advantages and disadvantages. We also consider the use of rate-control mechanisms, especially in the context of hierarchical scheduling architectures.

The emergence of ATM networks as the technology for B–ISDN has led to the development of a new class of bandwidth regulation mechanisms. We survey several different classes of techniques for achieving bandwidth regulation in ATM networks. In particular, we focus on the efforts of the ATM Forum to develop flow and congestion control techniques for the newly-defined Available-Bit-Rate Traffic [12].

In Chapter 3, we formally propose our bandwidth regulation mechanism. We first present the mathematical requirements of inter-class and intra-class fairness in a packet-switched network and provide a maximal bandwidth allocation that simultaneously satisfies both inter-class and intra-class fairness. We then develop a multi-level bandwidth regulation mechanism for regulating connectionless traffic in an ATM network. The first level, *short-term* regulation, is based on the idea of inter-class and intra-class fairness, similar to the mechanism developed for packet-switched networks. The second level, *medium-term* regulation, dynamically adjusts the class guarantees based on the average load of the classes. The third level, *long-term* regulation, controls the allocation of bandwidth to VPs carrying connectionless traffic.

In Chapter 4, we first develop a simple protocol to implement the bandwidth allocation scheme developed for packet-switched internetworks. The protocol is presented as a set of extensions to a connectionless network layer protocol. We describe a simulation experiment that demonstrates the effectiveness of the protocol. We then develop a protocol to implement the multi-level bandwidth allocation mechanism for ATM networks. The protocol is similar to the one developed for packet-switched networks and assumes that the ATM network supports connectionless traffic via a virtual network of CLSs. We also present a series of simulation experiments designed to illustrate the multilevel regulation achieved by the protocol.

Finally in Chapter 5, we discuss the contributions of the thesis, and the directions of future work.

Chapter 2

Literature Survey

The problem of regulating scarce bandwidth among competing flows has been extensively discussed in the literature. Early investigations into this problem focussed on packet-switched networks. The studies proposed various methods, such as the use of appropriate scheduling disciplines coupled with admission control functions [6, 7], or the use of rate-control mechanisms [10, 34, 35]. These are discussed in §2.1.

The development of ATM networks necessitated the development of a new class of bandwidth regulation mechanisms, which take into account the demands of the high switching rate and the small size of cells. In §2.2 we give a brief overview of bandwidth control mechanisms for ATM networks in general. We conclude the chapter by considering the efforts of the ATM Forum to develop flow and congestion control mechanisms for the newly-defined *Available-Bit-Rate* (ABR) traffic in §2.3 [12].

2.1 Bandwidth Regulation Mechanisms for Internetworks

Early research on bandwidth regulation concentrated on providing a minimum guaranteed service to individual flows irrespective of the network load. The preferred mechanism for achieving this goal is by means of appropriate scheduling disciplines at the routers. However, scheduling disciplines need admission control functions in order to maintain the guaranteed service levels. As the Internet does not support admission control, this class of algorithms cannot be applied. Rate control mechanisms have been investigated in an effort to retain the traditional, best-effort paradigm of the Internet [10, 35]. This class of bandwidth regulation mechanisms define a hierarchical set of flow classes and share the link bandwidth among the classes according to fairness criteria. Another scheme [20] allocates a certain maximum rate of transmission to each flow in the network. The network will only accept packets from a flow as long as the flow does not violate the allocated transmission rates. In contrast to admission control mechanisms, no restrictions are placed on the admissibility of a flow. Fair bandwidth regulation can be achieved by using fairness mechanisms.

2.1.1 Rate-based Scheduling Disciplines

One of the earlier approaches to regulating bandwidth is based on suitable scheduling algorithms at the intermediate gateways. A rate-based scheduling discipline guarantees a minimum rate of service to each flow, irrespective of the behavior of other flows. Several variations have been proposed to replace the traditional FIFO scheduling mechanism. In the *Fair Queuing* mechanism [6, 7], the bandwidth available on an output line is equally divided among all flows sharing that line. If a flow uses less than its share, the spare bandwidth is equally divided among the rest. Fair Queuing emulates a bitwise round–robin service. Variations of Fair Queuing can assign different fractions of the bandwidth to different flows by assigning weights to the flows [28]. In [14], the Fair Queuing discipline has been enhanced with a window-based flow control mechanism to avoid having to drop packets in the network. A scheme to emulate the Time Division Multiplexing (TDM) service discipline, the so-called *Virtual Clock* scheme, is proposed in [42]. It allocates a virtual 'transmission time' to each packet, which is the time at which the packet would be transmitted if the scheduling discipline was actually TDM. The *Delay Earliest-Due-Date* scheduling mechanism [9] is an extension of the classic Earliest-Due-Date scheduling [26]. In this mechanism, each flow has to negotiate with the router to obtain a service contract. The router contracts to provide a certain delay bound on each packet if the source submits packets in accordance with pre-specified peak and average sending rates. The router sets the deadline for each packet to the time it would have been sent if it had been transmitted according to the agreed sending rate. The router provides a hard delay bound by reserving bandwidth at the peak rate for each flow.

The *Jitter Earliest-Due-Date* scheme [37] extends the Delay-EDD scheme to enforce bounds on the minimum as well as the maximum delay (delay-jitter bounds). At every router a packet is stamped with the difference between its deadline and the actual finishing time. At the next router, a packet is made eligible for service only after it has waited for a period of time according to the timestamp. The Jitter Earliest-Due-Date mechanism enforces that a packet receives a constant delay at every router, hence jitter guarantees can be provided. The *Stop-and-Go* service discipline [13] divides time into frames. In each frame, only packets arriving at the router in the previous frame are sent. This prevents the traffic from becoming bursty and enforces a delay-jitter bound on the packets. The delay and jitter bounds that can be provided by a Stop-and-Go mechanism are determined by the length of the frame. Hence multiple frame sizes must be used to provide different delay and jitter bounds. The *Hierarchical Round Robin* (HRR) scheme [21] defines several service levels, each of which has a fixed number of slots (for packets). A router cycles through the levels, servicing the slots in each level in a round-robin manner. In this mechanism, the router requires a constant amount of time, called the *frame time*, to service all the slots at a given level. Thus, each level receives a fixed share of the link bandwidth. A flow is assigned to a particular service level, based on its service requirements. Bandwidth allocation is performed by reserving a number of slots for the flow in the selected service level. Each slot reserved by the flow will be serviced exactly once in every frame time. This allows the HRR scheme to provide a maximum delay bound to each packet.

The comparative advantages and drawbacks of the scheduling disciplines discussed above have been studied in [41]. The biggest complaint against the scheduling disciplines is that their only mechanism of congestion control is the dropping of cells at the congested gateway. The cells that are dropped in the network have, however, already consumed network resources, and as such, dropping them implies a wastage of network resources. In current high-speed networks, even a brief period of congestion is likely to cause a large number of cells to be dropped. If there is an end-to-end retransmission mechanism in place, then the congestion will be further worsened by the retransmission of the lost cells. This observation leads us to the conclusion that rate-based scheduling disciplines must be supplemented by some form of end-to-end rate control mechanism, such as [14].

2.1.2 Link-Sharing Mechanisms

A hierarchical scheduling architecture has been proposed in [34], in which the link bandwidth is divided amongst various traffic classes, such as guaranteed traffic, predictive traffic and elastic traffic classes. The *guaranteed service* class consists of those applications which require perfectly reliable delay and jitter bounds (so-called *hard* guarantees). The *predictive service* class comprises applications that can tolerate minor fluctuations in the service provided. Predictive service applications do not receive hard guarantees, but merely a commitment from the network that only a limited percentage of packets will fail to receive the service guaranteed. The *elastic service* class consists of applications that do not require quantitative performance guarantees from the network and receive "as–soon–as–possible" service only. Packets belonging to each class are queued separately at the switches. The hierarchical scheduling mechanism services the queue with guaranteed traffic only when necessary to meet guaranteed delay bounds. However at a given point in time, if there exists a packet in the guaranteed service queue that must be transmitted in order to meet delay requirements, then the scheduling mechanism will give absolute priority to the guaranteed service queue. This allows the scheduling mechanism to enforce hard guarantees on the delay and jitter bounds provided to flows in the guaranteed service class. Otherwise, the scheduler is free to transmit a packet from either the predictive service queue or the elastic service queue, subject to the restriction that choosing the elastic service queue will not violate commitments made to the predictive service class.

A mechanism based on the above architecture, *link-sharing*, has been proposed in [10]. In this mechanism, flows are organized into hierarchical classes, and bandwidth assigned accordingly. Flows may obtain excess, or unutilized bandwidth, in a hierarchical manner. The unutilized bandwidth is first claimed from the flow's immediate siblings in the hierarchy, before bandwidth from more distant flow's are claimed. Another approach to hierarchical link-sharing is presented in [35], in which flows are organized in a hierarchy of classes,

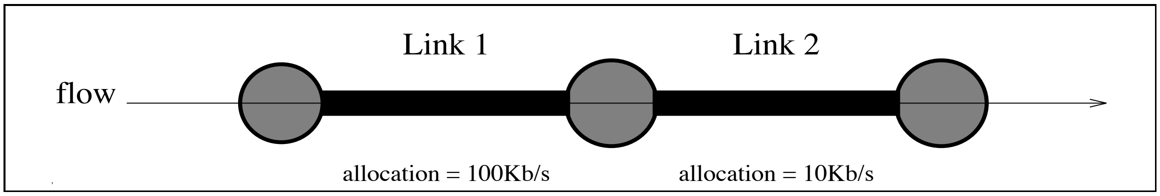


Figure 2.1: Throughput Restriction Due to a Bottleneck Link.

with bandwidth guarantees to classes at each level. It provides for a mixture of static high-priority flow classes and dynamic low priority flow classes. However, both approaches do not investigate the interactions between flows covering multiple links. Neither do they provide a mechanism for implementing intra-class fairness. In particular, they do not regulate the bandwidth allocation to individual flows explicitly. Such regulation can only be achieved by making each flow a separate class at the bottom of the class hierarchy.

2.1.3 Bottleneck Flow Control Mechanism

In contrast to the mechanisms discussed in §2.1.1 and §2.1.2, [20] regulates bandwidth in a complex network by controlling the rate at which traffic can enter the network. The motivation for this work is as follows. A flow is typically routed through more than one switch to the destination. The bandwidth that the flow obtains will be different for each switch. However, the end-to-end throughput of the flow will depend only on the minimum of the bandwidths that it is able to obtain at the switches on its route. The switch at which the flow receives the least allocation is the *bottleneck* switch for the entire route. Since the switch regulates access to a link, we shall use the terms *bottleneck switch* and *bottleneck link* interchangeably. As an example, consider the flow in Figure 2.1, which is routed over two links. It is able to receive an allocation of 100Kb/s at link 1 but only 10Kb/s at link 2. Assume further that the flow transmits at 100Kb/s. The flow will not experience any packet losses at link 1, but will lose 90% of its packets at link 2, which is its bottleneck link. Accordingly, its total throughput will be only 10Kb/s. Note further that out of the 100Kb/s utilized by the flow at link 1, 90Kb/s is actually wasted as those packets are subsequently dropped in the network. Hence, the allocation to this flow at link 1 could be reduced to 10Kb/s without affecting the end-to-end throughput of the flow.

The bottleneck flow control mechanism tries to maximize the overall throughput of the network by limiting the rate of transmission of a flow to the allocation received by the flow at its bottleneck link. This prevents packets from consuming network resources before being dropped. Such an allocation maximizes the overall throughput of the network, without discriminating against any flow. A distributed algorithm to achieve such an optimal bandwidth allocation has been developed in [20]. The allocation has been shown to satisfy fairness criteria while maximizing the network throughput. However, [20] does not consider different classes of flows, and provides no priority mechanisms.

2.2 Bandwidth Regulation Mechanisms for ATM Networks

The deployment of ATM in data networks led to a situation where connectionless traffic had to be supported on a connection-oriented network. Existing flow and congestion control algorithms, which were designed either for connectionless or connection-oriented networks, were inadequate for regulating data traffic in ATM networks. Traffic control mechanisms used in traditional packet switched networks could not cope with the high transmission rates and small cell size in ATM networks. Flow and congestion control techniques used in connection-oriented networks, on the other hand, could not handle the unpredictability and burstiness of data traffic. New traffic control mechanisms had to be developed to regulate data traffic in ATM networks. Any such protocol had to consider three key aspects. First, the small cell size and the high transmission rates of ATM meant that the time available to process each cell at a switch was very limited. This implied that protocols could not require complex processing at the switches. Second, as a higher level packet is carried by a number of cells, the loss of a single cell would lead to the retransmission of the entire packet. Thus, avoiding cell loss became vital. Lastly, the high transmission rate of ATM ensured that congestion will build up very rapidly. Therefore, protocols have to take steps to avoid congestion altogether, or be able to react very rapidly to congestion.

Several different approaches to controlling connectionless traffic in ATM networks have been explored in the literature. Burst-level bandwidth regulation mechanisms statistically multiplexed bursty traffic, allocating bandwidth to a flow only when the flow needed to transmit a burst of cells. Window-based regulation mechanisms adapted traditional window flow control techniques taking into account the high speed and low cell size of ATM networks. Bandwidth advertising schemes improved upon burst-level mechanisms by monitoring overall network utilization.

2.2.1 Burst-level Bandwidth Reservation

The problem of statistically multiplexing high-speed bursty traffic at the burst level has been studied in [2, 15, 36]. In [15], each burst is preceded by a *pilot cell*, which reserves bandwidth for the burst on one of a set of output links. The choice of the particular output link is determined by the availability of bandwidth on the link. If none of the links in the set can accommodate the burst, then the burst is either buffered or diverted to another set of output links. In a similar scheme [2], the *Fast Reservation Protocol* establishes VPs connecting a source switch to a destination switch without allocating any bandwidth. On arrival of a burst of cells, the VP attempts to allocate the required bandwidth on all its links. If it is successful, then the burst is transmitted, otherwise the burst is buffered and the bandwidth allocated to any links of the path is deallocated.

For both [2] and [15], the reservation is made at the peak rate of the source. This does not allow efficient statistical multiplexing of the flows, leading to a high burst blocking probability for high-speed flows. A better mechanism is proposed in [36], which reduces the blocking probability by maintaining multiple paths for a single source-destination pair. A path, in turn, consists of a number of hops, each of which comprises multiple links. When a burst arrives at the source, a request for bandwidth allocation is broadcast on all paths. The burst is transmitted on any one of the paths which succeeds in obtaining the required allocation. The rest of the paths release the bandwidth they had obtained. However, the delay involved in obtaining the bandwidth for each burst is considerable. Also, the bandwidth that is allocated to the alternate paths is wasted, which could result in other flows being blocked from the network.

2.2.2 Window-based Bandwidth Regulation

Window-based flow control schemes have been widely used in traditional packet-switched networks [19, 32]. But in case of ATM networks, window-based mechanisms could not be efficiently applied, because of the small size of the individual cells and the high rate of transmission. In [38], a modified window-based mechanism is used. The key observation underlying this scheme is that connectionless traffic is originally composed of packets which are several orders of magnitude larger than ATM cells. Since packets are segmented into ATM cells at the source gateway and reassembled at the destination gateway, it is possible to acknowledge the entire packet at once instead of acknowledging every cell. It is claimed in [38] that a packet size of 4500 bytes (FDDI packet size) is sufficiently large for an efficient acknowledgment mechanism. The maximum transmission rate of a flow will be controlled by the window size. The window size is dynamically varied using a pair of algorithms. The *bandwidth allocation* algorithm is invoked periodically, at the end of a so-called *slide interval*. It estimates the bandwidth utilized by a virtual path (VP) over the slide interval by counting the number of frames successfully transmitted by the VP over the *measuring interval*, which is some multiple of the slide interval. The algorithm then sets the bandwidth allocation of the VP for the next slide interval equal to the estimated bandwidth utilization. The bandwidth allocation algorithm tends to decrease the bandwidth allocated to a VP. A different algorithm is used to increase bandwidth allocation to VPs. The *bandwidth enlargement* algorithm monitors the input queue at the source of the VP. At the end of a slide interval, if the input queue exceeds a threshold, then the bandwidth enlargement algorithm increments the bandwidth allocation of the VP by a fixed amount. The principle drawback of this scheme is that it does not guarantee congestion avoidance, nor does it specify any fairness mechanism to regulate the bandwidth between multiple flows. In particular, the bandwidth enlargement algorithm increases the window size on demand, without considering the ability of the network to support the increased traffic.

2.2.3 Bandwidth Advertising and Renegotiation

The concept of *bandwidth advertising* is introduced in [5]. In this scheme, the *available bandwidth* on each outgoing link is defined as the difference between the link capacity and the sum of the bandwidth allocated to the existing VPs on that link. This bandwidth represents the spare capacity of the link and may be used by flows in excess of their allocated capacity for a short period of time. Each link periodically calculates its available capacity, and advertises the capacity to the sources of all flows passing through that link. Each flow then calculates the minimum available capacity that it encounters. A bandwidth policing mechanism enforces that a flow does not exceed the total of its own allocated capacity and the minimum available capacity on its route. Also, cells sent by a flow in excess of its allocation, i.e., by utilizing the available bandwidth, are marked by the policing mechanism and may be selectively dropped by the network when congestion occurs. This mechanism allows existing flows to exceed their bandwidth allocation temporarily if the load on the network is light. The *bandwidth renegotiation* algorithm tracks the rate of transmission of a flow, and requests an increase in the bandwidth allocation, if the flow continuously exceeds its allocation during a given interval. Likewise, the algorithm reduces the bandwidth allocation if the flow underutilizes its allocation. Similar mechanisms have also been proposed in [11, 27]. However, none of the mechanism give an algorithm to allocate bandwidth among competing flows, that will maximize the throughput of the network, without discriminating against any flow. Also, congestion is handled by dropping cells, which is a waste of network resources.

2.2.4 Bottleneck Bandwidth Regulation

Several schemes have been proposed to utilize the idea of the bottleneck link [20] to allocate bandwidth in a fair and waste-free manner. The conditions that an optimal bandwidth allocation must satisfy are specified in [43]. A critical value of traffic load is defined on each link, based on the number and relative usage of the flows using that link. This critical value is used to define two distinct class of flows: an *uncontrolled* flow receive an allocation equal to its offered traffic and a *controlled* flow receives an allocation less than its offered load. The primary fairness criterion, as defined in this work, requires that the relative throughput of a *controlled* flow will not be lower than that of any other flow sharing its bottleneck link. The proposed criterion is an essential requirement of a globally fair bandwidth allocation. However, a suitable algorithm to actually enforce this criterion in a practical network has not been proposed.

An algorithm to enforce the fairness criteria defined in [43] has been proposed in [44]. The algorithm implements the fairness discarding algorithm [30] at the switches to enforce compliance to the fair allocations. An adaptive flow control algorithm [31] is used to regulate the offered load of individual flows. The adaptive algorithm continuously increments the offered traffic of a flow by a fixed amount. If congestion is detected, then the offered load is reduced by a factor. The main drawback of this mechanism is the high overhead at the switches. The cell discard algorithm must maintain the actual usage statistics as well as the allocated bandwidth for each flow at every link. Also, the adaptive flow control algorithm will require a considerable time to increase the transmission of a flow.

2.3 Available-Bit-Rate Traffic and Bandwidth Regulation Mechanisms

The *Available-Bit-Rate* (ABR) traffic has been provisionally defined by the ATM Forum to be used for data applications such as LAN interconnection and emulation, interactive data transfer, file transfer, etc. [12]. This class of traffic is characterized by its extreme burstiness, caused by the fact that the size of the typical *Protocol Data Unit* (PDU) is several orders of magnitude larger than that of an ATM cell. As the PDUs are of varying size, the burst size and the short-term average rate of ABR traffic is unpredictable. The ATM Forum has proposed three performance requirements for ABR traffic. Firstly, ABR traffic should have a low cell loss rate (of the order of the transmission error rate). Secondly, during a burst, the minimum committed bandwidth required could be as high as the upper limit defined (e.g. access link rate). Finally, no delay or jitter requirements need to be specified.

Several mechanisms for controlling ABR traffic have been proposed to the ATM Forum. Prominent among these mechanisms are the *credit-based* congestion control approach and the *rate-based* flow control approach. We now compare the two approaches and discuss the final approach adopted by the ATM Forum.

2.3.1 Credit-based FCVC Proposal

The credit-based *Flow Controlled Virtual Circuit* mechanism proposes to achieve congestion control of ABR traffic by using a modified window flow control mechanism on a per-link basis [24]. In this model, a flow-controlled VC will pass through a number of switches. Each switch provides a pair of "send" and "receive buffers" for each VC. Arriving data cells are queued in the send buffers. A *credit balance* is maintained for each VC at the upstream switch, which is decremented every time a cell is transmitted. The initial value of the *credit balance* depends upon the size of the buffers present at the destination switch. A switch is allowed to transmit a cell on a VC if and only if the *credit balance* is positive. The downstream switch periodically sends a *Credit_Update* cell to the upstream switch. A *Credit_Update* cell increases the *credit balance* of the upstream switch. Buffer overflow at the switches is prevented by using credit cells (which transport values of the credit balance and other credit-related management information) to achieve flow control.

The allocation of the buffers may be done statically or dynamically. Static allocation is useful at switches on the boundaries of the network, where the number of simultaneously active VCs is expected to be small. Dynamic allocation policies are more suited for the internal switches, where the number of VCs are expected to be larger. This allows a rate control mechanism to work in conjunction with the FCVC mechanism. The rate control mechanism will regulate the amount of buffers allocated to each flow, and the FCVC mechanism will enforce compliance to the rates. This mechanism results in a fast feedback response to congestion, since congestion will be detected at the link level itself.

2.3.2 Closed-Loop Rate-based Proposal

Another mechanism that was considered by the ATM Forum was the closed-loop rate-based proposal [39]. Unlike the FCVC approach, the rate-based approach controls congestion on an end-to-end basis by utilizing the *explicit forward congestion indication* (EFCI) bit in the header of an ATM cell. The EFCI bit of a data cell may be set by any intermediate switch that experiences congestion. The destination periodically generates a control cell that is sent back to the source, indicating whether or not congestion was experienced. In the event of congestion, the source adapts by reducing its rate of transmission. The source adapts its transmission rate as follows. It will continue to increase its transmission additively as long as it does not experience congestion. When it receives a control cell carrying congestion notification its transmission rate undergoes a multiplicative decrease.

In another version of the proposal [40], a special *Resource Monitor* (RM) cell is generated by each source after a fixed number of data cells has been sent, and transmitted to the destination. The intermediate switches mark the EFCI bit of the data cells as in the previous instance. The destination marks the *Congestion Indicator* (CI) field of the RM cell, if one of the data cells received before the RM cell has its EFCI bit set or the destination itself is experiencing congestion. The destination then returns the RM cell back to the source. Intermediate switches can set the CI field of the returning RM cell. The source adjusts its transmission rate based on two parameters: the *Initial Cell Rate* (ICR) and the *Allowed Cell Rate* (ACR). The source starts transmitting at the ICR, which is established at connection setup. Its ACR is set equal to the ICR. Next, it decreases its ACR until it receives a backward RM cell whose CI is set to 0, i.e., the RM cell did not encounter congestion. On receiving the RM cell, the source increases its transmission rate such that its new rate is greater than the rate it had before it transmitted the last RM cell. Both rate decreases and rate increases are proportional to the current ACR of the source.

An attractive feature of [40] is that the feedback loop need not be end–to–end, but can be terminated by any intermediate switch. Thus, the mechanism can be used to tunnel a flow through an older network that does not support the mechanism.

A proposed extension to [40] adds two extra fields to the RM cell: the *Explicit Rate* (ER) and the *Allowed Cell Rate* (ACR) [33]. The ER field of the forward RM cell is set to the peak cell rate at the source. As the RM cell travels along the path, the ER may be reduced by intermediate networks. When the RM cell loops back to the source, the ACR of the source is set to the smaller of the received ER value and the current ACR value of the source. This mechanism allows the intermediate networks to reduce the connection rate in response to transient conditions. The ACR field can be used to convey the ACR information to intermediate networks. The ACR information may be used by intermediate networks to selectively indicate congestion according to predetermined fairness or priority mechanisms.

Overall, the rate-based approach has several advantages over the credit-based approach. The main advantage is that the rate-based approach does not require a complicated switch architecture, unlike the credit-based approach, where separate buffers have to be managed for every VC, and is therefore compatible with existing switches and systems. The ATM Forum has decided to adopt the rate-based approach in preference to the credit-based approach. However, the exact details of the algorithms to be used has not been finalized. The motion adopted at the September 1994 meeting of the ATM Forum limits the scope of the mechanism to support ABR service to developing a framework based on feedback control of the source rate that supports end-to-end flow control. The participation of the switches is limited to using the EFCI bit to signal congestion in the forward direction, or to inform the source to dynamically change the explicit upper bound on the source rate. Accordingly, the selected mechanism should have control information formats that allow both types of switches to coexist within the same control loop and interoperate with the source. The selected mechanism should also allow intermediate networks the option of segmenting the control loop at any point. Finally, the ATM Forum placed a high priority on simplifying the complexity of the end system.

Chapter 3

Theoretical Results

In this chapter we present our bandwidth regulation mechanisms. First, in §3.1, we devise a bandwidth regulation mechanism for traffic flows in a traditional packet-switched internetwork. The flows may be classified into several so-called *flow classes*, which are sets of flows with similar service requirements. We define two fairness conditions: *inter-class* fairness, for allocating bandwidth among the various flow classes, and *intra-class* fairness for allocating bandwidth among flows belonging to the same flow class. We use these fairness criteria to formally develop a bandwidth allocation mechanism that achieves the twin goals of inter-class and intra-class fairness.

In §3.2, we consider an ATM network that supports connectionless traffic by means of connectionless servers. As before, the connectionless traffic consists of multiple traffic classes. We take advantage of the flexibility of bandwidth allocation offered by ATM to achieve bandwidth regulation at three different levels. *Short-term* bandwidth regulation is similar to the mechanism developed for the packet-switched internetwork. *Medium-term* regulation is concerned with monitoring and allocating so-called *class guarantees* to the flow classes. Finally, *long-term* regulation is concerned with bandwidth allocation to connectionless traffic as a whole.

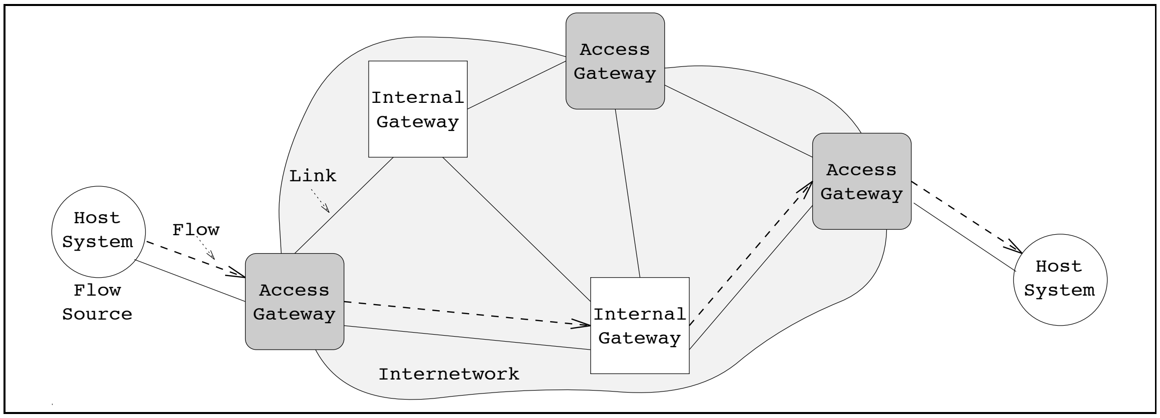


Figure 3.1: Internetwork Model.

3.1 Bandwidth Allocations with Intra-class and Inter-class Fairness

The network model assumed in this section consists of a set of gateways which are connected via point-to-point links as shown in Figure 3.1. Hosts access the network by connecting to a so-called *access gateway*. Each host can transmit to any other host connected to the network. A traffic stream from a source host to a destination host is referred to as a *flow*. We assume that each flow is carried over a fixed route of network gateways. The network distinguishes different types of traffic, the abovementioned *flow classes*, and may have bandwidth guarantees for flow classes on some network links. We assume that all traffic in the network can be accurately described in terms of traffic rates. The traffic rate which describes the bandwidth demand of a flow is referred to as the *offered load*. The rate of actual data transmission is called the *throughput* of the flow.

We describe the network by a tuple $\mathcal{T} = (\mathcal{P} \cup \{0\}, \mathcal{F}, \mathcal{L})$, where $\mathcal{P} \cup \{0\}$ is the set of flow classes that are distinguished in the network. Traffic that does not belong to one of the classes in $\mathcal{P}$ is assigned to the *default class* '0'. $\mathcal{F} = \bigcup_{p\in\mathcal{P}\cup\{0\}} \mathcal{F}_p$ is the set of flows in the network, and $\mathcal{F}_p$ is the set of flows with traffic from flow class p. $\mathcal{L}$ is a set of unidirectional *network links* which connect the gateways, and C_l denotes the capacity of link $l \in \mathcal{L}$ (in bits per second).

For each flow i, the fixed route of the flow is given by a sequence of links $\mathcal{R}_i = (l_{i_1}, l_{i_2}, \ldots, l_{i_K})$ with $l_{i_k} \in \mathcal{L}$ for $1 \leq k \leq K$. We use Δ_{lp} to denote the set of flows from class p which have link l on their route, that is, $\Delta_{lp} = \{i \mid l \in \mathcal{R}_i \ , \ i \in \mathcal{F}_p\}$.

At each link, flow class p may have a *bandwidth guarantee* of $G_{lp} > 0$ with $\sum_{p \in \mathcal{P}} G_{lp} < C_l$. Let $\mathcal{P}_l$ denote the set of classes with a positive guarantee at link l, that is, $\mathcal{P}_l = \{p \in \mathcal{P} \mid G_{lp} > 0\}$. If a class-$p$ flow i has link l on its route, i.e., $i \in \Delta_{lp}$, but link l does not have a bandwidth guarantee for class p, i.e., $p \notin \mathcal{P}_l$, flow i is assigned to default class '0' at this link. The bandwidth guarantee to class 0 at link l is given by $G_{l0} = C_l - \sum_{p \in \mathcal{P}} G_{lp}$.

Let the *surplus* of a flow class, ϕ_{lp} be the maximum bandwidth that a class can utilize at a link in excess of its guarantee, G_{lp}. A class may receive different surpluses at different links, and classes on the same link may receive different surpluses. A class can utilize bandwidth in excess of its guarantee only when there exists some other class which does not utilize its full guarantee. It does so by 'borrowing' bandwidth from the class which is unable to utilize its full guarantee.

Let the *ceiling* of a class-p flow i, $\alpha_{ip}(l)$ be the maximum bandwidth that flow i can receive at each link on its route. The ceiling $\alpha_{ip}(l)$ may be different at different links along the route of a flow and may be different for flows sharing the same link. The *bottleneck* link for a flow i, l_i^* , is the link at which its ceiling is the minimum, that is $\alpha_{ip}(l_i^*) = \min\limits_{l \in \mathcal{R}_i} (\alpha_{ip}(l))$.

Let $\lambda_i \geq 0$ and $\gamma_i \geq 0$, respectively, denote the offered load and the throughput of flow i. The offered load of all flows is given by the load set $\mathbf{\Lambda}$ which contains the λ_i as elements. The throughput of all flows is given by the throughput set $\mathbf{\Gamma}$ which contains the γ_i as elements. Finally, we define the surplus set, $\mathbf{\Phi}$, which contains the sets $\{(\alpha_{ip}(l), \ \phi_{lp}) \mid i \in \mathcal{F}_p \text{ and } l \in \mathcal{R}_i\}$ for each class-p flow, i.

With the above notation at hand we can introduce the notion of a *bandwidth allocation* which maps the offered load of each flow into its throughput.

Definition 3.1 *Given a network topology $\mathcal{T}$ with offered load set $\mathbf{\Lambda}$ and surplus set $\mathbf{\Phi}$. A* <u>bandwidth allocation</u> *is a relation*

$\Omega = \{(\lambda_i, \{(\alpha_{ip}(l), \ \phi_{lp}) \mid l \in \mathcal{R}_i\}, \ \gamma_i) \in \mathbf{\Lambda} \times \mathbf{\Phi} \times \mathbf{\Gamma} \mid i \in \mathcal{F}_p \ \}$ *such that*

1. $\gamma_i \leq \min(\lambda_i, \alpha_{ip}(l_i^*))$.

2. $\displaystyle\sum_{p \in \mathcal{P} \cup \{0\}} \sum_{i \in \Delta_{lp}} \gamma_i \leq C_l \quad \text{for all} l \in \mathcal{L}$.

3. $\displaystyle\sum_{j \in p} \gamma_j \leq G_{lp} + \phi_{lp}$ *for all* $p \in \mathcal{P}$.

The first condition enforces that no flow can have a higher throughput than given by its load. The second condition enforces that the throughput at a link cannot exceed its capacity. The third condition enforces that a class can exceed its guarantee by no more than its surplus.

Next we introduce bandwidth allocations which provide *inter-class* fairness. Recall that the capacity C_l of a link l is divided into bandwidth guarantees G_{lp} for each class $p \in \mathcal{P}_l$ with $\sum_{p \in \mathcal{P}_l} G_{lp} = C_l$. If a flow class p does not utilize its bandwidth guarantee at a link, the unused bandwidth, i.e., $G_{lp} - \sum_{i \in \Delta_{lp}} \gamma_i$, can be made available to other flow classes. A flow class may not utilize its guarantee at a link for three reasons. The total load of the class can be less than its guarantee, or the ceilings of the flows may be less than the guarantee, or the throughput of class-p flows is limited due to restrictions at other links. An inter-class fair bandwidth allocation will assign the unused bandwidth equally, i.e., fairly, among flow classes which can take advantage of the additional capacity. The maximum bandwidth at link l that a flow class p can 'borrow' from the guarantees of other classes is the *surplus*, ϕ_{lp}. Then inter-class fairness will enforce that $\phi_l \equiv \phi_{lp}$ for all classes $p \in \mathcal{P}_l$.

In the following, we will use C_{lp} to denote the *available bandwidth* of flow class p at link l with $C_{lp} = \sum_{j \in \Delta_{lp}} \gamma_j$.

Definition 3.2 *A bandwidth allocation is said to be* <u>inter-class fair</u> *if for each link* $l \in \mathcal{L}$ *there exists a surplus value* ϕ_l *such that for all* $p \in \mathcal{P}_l$

$$C_{lp} = \min\left(\sum_{i \in \Delta_{lp}} \min(\lambda_i, \alpha_{ip}(l_i^*)), G_{lp} + \phi_l\right)$$

In particular, a bandwidth allocation which does not permit flow classes to borrow unused bandwidth from other flow classes, i.e., $\phi_l \equiv 0$, satisfies inter-class fairness. However, such an allocation results in a waste of link bandwidth. In Lemma 3.1 we state that by selecting ϕ_l as large as possible, one can make the entire link bandwidth available for transmission.

Lemma 3.1 *Given a bandwidth allocation which satisfies inter-class fairness. The surplus ϕ_l at link l is maximal, if and only if*

$$\sum_{p \in \mathcal{P}_l} \sum_{i \in \Delta_{lp}} \gamma_i = C_l$$

whenever $\sum_{i \in \Delta_{lq}} \gamma_i = G_{lq} + \phi_l$ for at least one flow class $q \in \mathcal{P}_l$.

Proof:

Obviously, if the entire capacity of link l is utilized the surplus cannot be increased. On the other hand, if $\sum_{p \in \mathcal{P}_l} \sum_{i \in \Delta_{lp}} \gamma_i < C_l$ we can increase the surplus ϕ_l by dividing all unused bandwidth, that is, $C_l - \sum_{p \in \mathcal{P}_l} \sum_{i \in \Delta_{lp}} \gamma_i$ to all flow classes with $\sum_{i \in \Delta_{lq}} \gamma_i = G_{lq} + \phi_l$. □

Next we discuss *intra-class fair* bandwidth allocations. For the special case of only one flow class our fairness criteria is similar to the fairness definitions in [20]. *Intra-class fairness* is concerned with allocating bandwidth to flows from the same flow class. It is based on the notion of a so-called *class-p share*, denoted by $\alpha_p(l)$, which defines a throughput bound of a class-p flow on a link $l \in \mathcal{L}$. Recall that each flow i, has a *ceiling* $\alpha_{ip}(l)$ at each link. Intra-class fairness enforces that for all flows $i \in \Delta_{lp}$,

$$\alpha_{ip}(l) = \alpha_p(l)$$

We set $\alpha_p(l) := \alpha_0(l)$ if $p \notin \mathcal{P}_l$. Since a flow is carried over possibly many links, the maximum throughput of a class-p flow is set to the smallest class-p share of all links on the route of a flow. For a given class-p flow i, we refer to the link with the smallest class-p share as the *bottleneck* link of flow i, denoted by l_i^*. Then the maximum throughput of a flow is given by:

$$\alpha_p(l_i^*) = \min_{l \in \mathcal{R}_i} (\alpha_p(l))$$

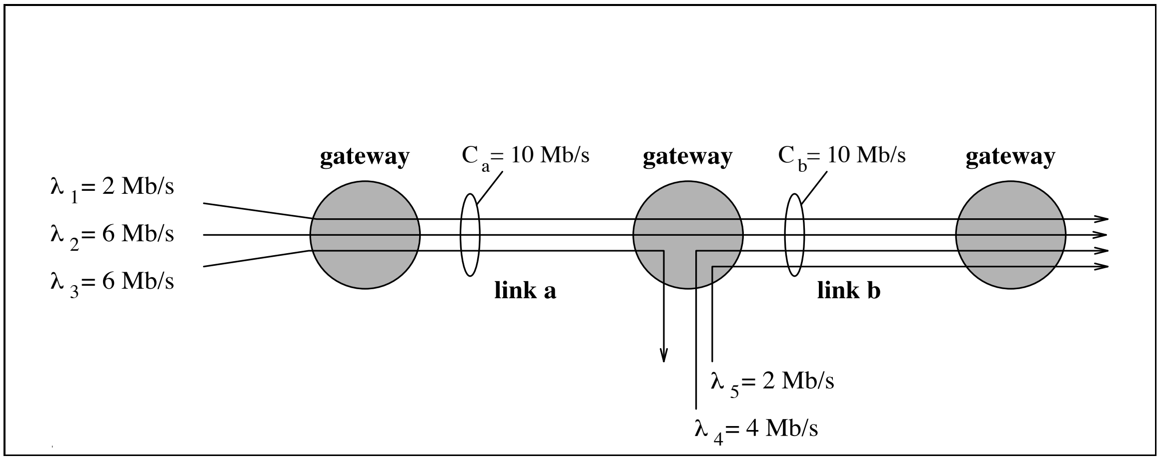


Figure 3.2: Intra-Class Fairness in a Network with Two Links.

Definition 3.3 *A bandwidth allocation is said to be* intra-class fair *if for each link $l \in \mathcal{L}$ there exist values $\alpha_p(l) > 0$ for all $p \in \mathcal{P}_l$ such that for all flows $i \in \mathcal{F}_p$*

$$\gamma_i = \min\left(\lambda_i, \alpha_p(l_i^*)\right)$$

As an example of intra-class fairness, consider the network in Figure 3.2 with two links, denoted by 'a' and 'b', and one flow class. Each link has a capacity of 10 Mb/s. Flows from the set $\mathcal{F} = \{1, 2, 3, 4, 5\}$ have routes in this network as shown in the Figure, and the offered loads are given by $\lambda_1 = 2$ Mb/s, $\lambda_2 = 6$ Mb/s, $\lambda_3 = 6$ Mb/s, $\lambda_4 = 4$ Mb/s, and $\lambda_5 = 2$ Mb/s. Setting the share values to $\alpha_a = 5$ Mb/s and $\alpha_b = 3$ Mb/s, respectively, for link a and link b, we obtain the following throughput values from Definition 3.3: $\gamma_1 = 2$ Mb/s, $\gamma_2 = 3$ Mb/s, $\gamma_3 = 5$ Mb/s, $\gamma_4 = 3$ Mb/s, and and $\gamma_5 = 2$ Mb/s. Flows *1* and *5* satisfy $\lambda_1 \leq \alpha_b \leq \alpha_a$ and $\lambda_5 \leq \alpha_b$, respectively, and obtain a throughput equal to their offered load. Both flows *2* and *4* have their bottleneck at link *'b'*, and satisfy $\lambda_2 \geq \alpha_b$ and $\lambda_4 \geq \alpha_b$, respectively. Hence, both flows obtain the same throughput $\gamma_2 = \gamma_4 = \alpha_b$. Flow *3* has its bottleneck at link *'a'* and $\gamma_3 = \min\left(\lambda_3, \alpha_a\right) = \alpha_a$.

In the above example, a different selection for the values of the link shares α_a and α_b either leaves a portion of the link bandwidth unused, e.g., if $\alpha_b < 3$ Mb/s, or will violate the given fairness rules, e.g., if $\alpha_b > 3$ Mb/s. We refer to the maximum values for shares, that do not leave capacity available to a flow class unused if the total offered load exceeds the capacity as the *maximal share*. In Lemma 3.2 we give the condition that must hold if the shares in a network with multiple flow classes are maximal.

Lemma 3.2 *The values of the class-p shares in an intra-class fair bandwidth allocation are maximal, if and only if for all flows $i \in \mathcal{F}_p$ with $\gamma_i < \lambda_i$*

$$\sum_{j \in \Delta_{l_i^* p}} \gamma_j = G_{l_i^* p} + \phi_{l_i^*}$$

In other words, the shares are maximized if and only if the available bandwidth at the bottleneck of all those flows which cannot transmit their entire load is fully utilized.

Proof:

Consider the bottleneck link l_i^* of flow i. Clearly, the class-p shares at this link cannot be increased if the available bandwidth is fully utilized. On the other hand, if $\sum_{j \in \Delta_{l_i^* p}} \gamma_j < C_{l_i^* p}$, the class-$p$ share of the link can be increased by dividing the unused available bandwidth over all flows $i \in \Delta_{l_i^* p}$ with $\lambda_i > \gamma_i$. $\square$

Our two fairness definitions are concerned with allocating bandwidth to flows of the same flow class (*intra-class fairness*), and to entire flow classes (*inter-class fairness*). Indeed, inter-class and intra-class fairness are two independent concepts. One can easily imagine bandwidth allocations that are inter-class but not intra-class fair, and vice versa. In particular, all proposals for hierarchical link sharing [10, 34, 35] enforce certain fairness criteria for flow classes (different from our inter-class fairness), but do not solve the fairness problem for flows from the same class.

We can follow from Lemma 3.1 that an intra-class fair bandwidth allocation without maximal shares can result in a waste of available bandwidth. Likewise, Lemma 3.2 implies that a bandwidth allocation with inter-class fairness but without the maximal surplus values may leave bandwidth unused. Therefore, one is interested in finding bandwidth allocations which are inter-class fair with maximal surplus values, and intra-class fair with maximal shares. In Theorem 3.1, our main result of this study, we state that such a bandwidth allocation is uniquely determined for general networks, and can be effectively constructed.

Theorem 3.1 *Given a network with topology $\mathcal{T}$ with offered load set $\mathbf{\Lambda}$. Then the following bandwidth allocation is both intra-class fair with maximal shares $\alpha_p^*(l)$ and inter-class fair with maximal surplus values ϕ_l^** [1].

$$\alpha_p^*(l) = \begin{cases} \dfrac{G_{lp} + \phi_l^* - \Theta_{lp}}{|\mathbf{O}_{lp}|} & \textit{if } \mathbf{O}_{lp} \neq \emptyset \\ \infty & \textit{otherwise} \end{cases} \tag{3.1}$$

and

$$\phi_l^* = \begin{cases} \dfrac{C_l - \sum_{\mathbf{O}_{lq} \neq \emptyset} G_{lq} - \sum_{\mathbf{O}_{lq} = \emptyset} \Theta_{lq}}{|\{q \in \mathcal{P}_l \mid \mathbf{O}_{lq} \neq \emptyset\}|} & \textit{if } \bigcup_{q \in \mathcal{P}_l} \mathbf{O}_{lq} \neq \emptyset \\ \infty & \textit{otherwise} \end{cases} \tag{3.2}$$

subject to the side conditions:

$$G_{lp} + \phi_l^* - \Theta_{lp} \geq 0 \tag{3.3}$$

$$C_l - \sum_{\mathbf{O}_{lq} \neq \emptyset} G_{lq} - \sum_{\mathbf{O}_{lq} = \emptyset} \Theta_{lq} \geq 0 \tag{3.4}$$

where $\Theta_{lp} = \sum_{i \in \mathbf{U}_{lp}} \lambda_i + \sum_{k \in \mathcal{L}} |\mathbf{R}_{lp}(k)| \cdot \alpha_p^*(k)$ *and the sets* $\mathbf{U}_{lp}$, $\mathbf{R}_{lp}$, *and* $\mathbf{O}_{lp}$ *are defined for all* $p \in \mathcal{P}_l$ *as*

$$\mathbf{U}_{lp} = \{i \in \Delta_{lp} \mid \alpha_p^*(l) \geq \lambda_i \,,\ i \notin \bigcup_{k \in \mathcal{L}} \mathbf{R}_{lp}(k)\} \tag{3.5}$$

$$\mathbf{O}_{lp} = \{i \in \Delta_{lp} \mid l = l_i^* \,,\ \alpha_p^*(l) < \lambda_i\} \tag{3.6}$$

$$\mathbf{R}_{lp}(k) = \{i \in \Delta_{lp} \mid k = l_i^* \,,\ \alpha_p^*(k) < \lambda_i\} \qquad \textit{for } k \neq l \tag{3.7}$$

Note that each class-p flow i with link l on its route belongs to one of the sets $\mathbf{U}_{lp}$, $\mathbf{O}_{lp}$, or $\mathbf{R}_{lp}(k)$ ($k \in \mathcal{R}_i$). Recall that for $p \notin \mathcal{P}_l$, a class-p flow i with link l on its route is assigned to default class '0' for link l. $\mathbf{U}_{lp}$ is interpreted as the set of *underloaded* class-p flows on link l. It contains flows from class p which can satisfy their end-to-end bandwidth demand at link

[1] In equations (3.1) and (3.2), $|X|$ denotes the cardinality of a set X.

l. Thus, if a flow is underloaded on some link, it is underloaded on all links on its route. $\mathbf{O}_{lp}$ and $\mathbf{R}_{lp}(k)$ contain flows i with $\gamma_i < \lambda_i$, that is, the bandwidth demand of the flow is greater than its throughput. $\mathbf{O}_{lp}$, the set of *overloaded* class-p flows on link l, contains flows which have link l as the bottleneck. $\mathbf{R}_{lp}(k)$, the set of *restricted* class-p flows, contains flows whose throughput is restricted and have their bottleneck at link k ($k \neq l$). Since for both overloaded and restricted class-p flows, the throughput is limited to the maximal class-p share at the bottleneck, each restricted flow at link l is overloaded at some other link on its route.

We now proceed to demonstrate that a solution to equations (3.1)–(3.7) satisfies both intra-class and inter-class fairness criteria, and is the maximal solution. In Lemma 3.3, we show that the shares $\alpha_p^*(l)$ obtained from equations (3.1)–(3.7) are both maximal and intra-class fair.

Lemma 3.3 *Let* $\{\alpha_p^*(l) \mid l \in \mathcal{L}\}$ *be obtained from solving the equation system in (3.1) – (3.7). Then the* $\alpha_p^*(l)$ *are maximal shares of an intra-class fair bandwidth allocation.*

Proof:

To show that the shares are maximal, we take an arbitrary flow i, and investigate its bottleneck link, say link l', with

$$\alpha_p^*(l') = \min_{l \in \mathcal{R}_i} (\alpha_p(l)) \tag{3.8}$$

Since link l' is the bottleneck for i we have for all $l \in \mathcal{L}$ that $i \notin \bigcup_{k \in \mathcal{L}} \mathbf{R}_{l'p}(l_k)$. Thus, either $i \in \mathbf{U}_{l'p}$ or $i \in \mathbf{O}_{l'p}$. We show that for either case, the bandwidth allocation to flow i satisfies intra-class fairness, and is maximal.

Case 1: $i \in \mathbf{U}_{l'p}$.

If $i \in \mathbf{U}_{l'p}$, we have per definition of $\mathbf{U}_{l'p}$ (in (3.5)), that $\lambda_i \leq \alpha_p(l')$. Therefore, $\gamma_i = \lambda_i$, which is in accordance with Definition 3.3. Hence, the bandwidth allocation to i satisfies intra-class fairness. The allocation is obviously maximal.

Case 2: $i \in \mathbf{O}_{l'p}$.

If $i \in \mathbf{O}_{l'p}$, we have per definition of $\mathbf{O}_{l'p}$ (in (3.6)) that $\lambda_i > \alpha_p(l')$. Therefore,

$$\gamma_i = \min\left(\lambda_i, \min_{l \in \mathcal{R}_i} \alpha_p(l)\right) \tag{3.9}$$

Hence, from Definition 3.3, we conclude that the bandwidth allocation to i is intra-class fair. Also, we obtain

$$\begin{aligned}
\sum_{j \in \Delta_{l'p}} \gamma_j &= \sum_{i \in \mathbf{U}_{l'p}} \gamma_i + \sum_{k \in \mathcal{L}} \sum_{i \in \mathbf{R}_{l'p}(k)} \gamma_i + \sum_{i \in \mathbf{O}_{l'p}} \gamma_i && (3.10)\\
&= \sum_{i \in \mathbf{U}_{l'p}} \lambda_i + \sum_{k \in \mathcal{L}} \left|\mathbf{R}_{l'p}(l_k)\right| \cdot \alpha_p^*(k) + \\
&\quad |\mathbf{O}_{l'p}| \cdot \frac{G_{l'p} + \phi_{l'}^* - \sum_{i \in \mathbf{U}_{l'p}} \lambda_i - \sum_{k \in \mathcal{L}} |\mathbf{R}_{l'p}(l_k)| \cdot \alpha_p^*(k)}{|\mathbf{O}_{l'p}|} \\
&= G_{l'p} + \phi_{l'}^* && (3.11)
\end{aligned}$$

Hence, it follows from Lemma 3.2 that the bandwidth allocation to flow i is maximal if $i \in \mathbf{O}_{l'p}$.

From Case 1 and Case 2, we have proved that the $\alpha_p^*(l)$ are the maximal shares of an intra-class fair bandwidth allocation. □

In Lemma 3.4, we show that the shares $\phi^*(l)$ obtained from equations (3.1)–(3.7) are both maximal and inter-class fair.

Lemma 3.4 *Let $\{\phi_l^* \mid l \in \mathcal{L}\}$ be obtained from solving the equation system in (3.1) – (3.7). Then the ϕ_l^* are maximal surpluses of an inter-class fair bandwidth allocation.*

Proof:

Let us consider a link l. We consider two cases: one, when the link has no overloaded flows, and two, when the link contains at least one overloaded flow. We show that in both cases the value of ϕ_l^* is maximal and satisfies inter-class fairness conditions.

Case 1: The link has no overloaded flows, i.e., $\bigcup_{q \in \mathcal{P}_l} \mathbf{O}_{lq} = \emptyset$.

The available bandwidth of a class p, C_{lp}, is equal to the sum of the throughputs of all the flows in that class. Formally:

$$C_{lp} = \sum_{i \in \Delta_{lp}} \gamma_i \tag{3.12}$$

As $\bigcup_{q \in \mathcal{P}_l} \mathbf{O}_{lq} = \emptyset$, obviously $\mathbf{O}_{lp} = \emptyset$. Therefore, it follows from equations (3.5)–(3.7) that

$$\begin{aligned} C_{lp} &= \sum_{i \in \Delta_{lp}} \gamma_i \\ &= \Theta_{lp} \\ &= \sum_{i \in \mathbf{U}_{lp}} \lambda_i + \sum_{k \in \mathcal{L}} |\mathbf{R}_{lp}(k)| \cdot \alpha_p^*(k) \end{aligned} \tag{3.13}$$

From the definition of $\mathbf{U}_{lp}$ in (3.5), we may write

$$\lambda_i = \min\left(\lambda_i, \alpha_p^*(l^*)\right) \qquad \text{for all } i \in \mathbf{U}_{lp} \tag{3.14}$$

Similarly, from the definition of $\mathbf{R}_{lp}(k)$ in (3.7), we may write

$$\alpha_p^*(k) = \min\left(\lambda_i, \alpha_p^*(k)\right) \qquad \text{for all } i \in \mathbf{R}_{lp}(k) \tag{3.15}$$

Combining equations (3.13)–(3.15), we have

$$C_{lp} = \sum_{i \in \Delta_{lp}} \min\left(\lambda_i, \alpha_p^*(l*)\right) \tag{3.16}$$

Further, from (3.3) we have

$$\Theta_{lp} \leq G_{lp} + \phi_l^* \tag{3.17}$$

Therefore, equation (3.16) may be rewritten as

$$C_{lp} = \min\left(\sum_{i \in \Delta_{lp}} \min\left(\lambda_i, \alpha_p^*(l^*)\right), G_{lp} + \phi_l^*\right) \tag{3.18}$$

By Definition 3.2 and equation (3.18), the allocation of ϕ_l^* satisfies inter-class fairness criteria. Further, as $\phi_l^* = \infty$, the allocation of ϕ_l^* is obviously maximal.

Case 2: The link has at least one overloaded flow, i.e., $\bigcup_{q \in \mathcal{P}_l} \mathbf{O}_{lq} \neq \emptyset$.

Consider a flow class $p \in \mathcal{P}_l$. If $\mathbf{O}_{lp} = \emptyset$, then we may show, by exactly the same sequence of steps, that

$$C_{lp} = \min\left(\sum_{i \in \Delta_{lp}} \min\left(\lambda_i, \alpha_p^*(l*)\right), G_{lp} + \phi_l^*\right) \tag{3.19}$$

Let p be an overloaded class, i.e., $\mathbf{O}_{lp} \neq \emptyset$. From equations (3.9)–(3.11), we have the result

$$G_{lp} + \phi_l^* = \sum_{i \in \mathbf{U}_{l'p}} \gamma_i + \sum_{k \in \mathcal{L}} \sum_{i \in \mathbf{R}_{l'p}(k)} \gamma_i + \sum_{i \in \mathbf{O}_{l'p}} \gamma_i \tag{3.20}$$

From the definition of $\mathbf{O}_{lp}$ in (3.6), we may write,

$$\alpha_p^*(l) = \min\left(\lambda_i, \alpha_p^*(l)\right) \qquad \text{for all } i \in \mathbf{O}_{lp} \tag{3.21}$$

Combining equations (3.15), (3.16), and (3.21) with equation (3.20), we obtain

$$\begin{aligned} C_{lp} &= \sum_{i \in \Delta_{lp}} \gamma_i \\ &= \min\left(\sum_{i \in \Delta_{lp}} \gamma_i, G_{lp} + \phi_l^*\right) \\ &= \min\left(\sum_{i \in \Delta_{lp}} \min\left(\lambda_i, \alpha_p^*(l^*)\right), G_{lp} + \phi_l^*\right) \end{aligned} \tag{3.22}$$

From (3.19) and (3.22) and Definition 3.2 it follows that the allocation of ϕ_l^* satisfies inter-class fairness.

We now prove that the allocation is maximal.

For all classes $p \in \mathcal{P}_l$, such that $\mathbf{O}_{lp} \neq \emptyset$,

$$\begin{aligned} \sum_{i \in \Delta_{lq}} \gamma_i &= \alpha_p^*(l) \cdot |\mathbf{O}_{lp}| + \sum_{i \in \mathbf{U}_{lp}} \lambda_i + \sum_{k \in \mathcal{L}} |R_{lp}(k)| \cdot \alpha_p^*(k) \\ &= G_{lp} + \phi_l^* \end{aligned}$$

Next, we show that the sum of the throughputs of all flows passing through link l is equal to the capacity of the link, C_l.

$$\sum_{p \in \mathcal{P}_l} \sum_{i \in \Delta_{lp}} \gamma_i = \sum_{\mathbf{O}_{lq} \neq \emptyset} (G_{lq} + \phi_l^*) + \sum_{\mathbf{O}_{lq} = \emptyset} \left(\sum_{i \in \mathbf{U}_{lq}} \lambda_i + \sum_{k \in \mathcal{L}} |R_{lq}(k)| \cdot \alpha_q^*(k)\right)$$

$$\begin{aligned} &= \sum_{\mathbf{O}_{lq}\neq\emptyset} G_{lq} + \sum_{\mathbf{O}_{lq}=\emptyset}\left(\sum_{i\in\mathbf{U}_{lq}}\lambda_i + \sum_{k\in\mathcal{L}}|R_{lq}(k)|\cdot\alpha_q^*(k)\right) \\ &\quad + \phi_l^*\cdot|\{q\in\mathcal{P}_l \mid \mathbf{O}_{lq}\neq\emptyset\}| \end{aligned} \tag{3.23}$$

By substituting the definition of ϕ_l^* from equation (3.2) we have

$$\begin{aligned} \sum_{p\in\mathcal{P}_l}\sum_{i\in\Delta_{lp}}\gamma_i &= \sum_{\mathbf{O}_{lq}\neq\emptyset} G_{lq} + \sum_{\mathbf{O}_{lq}=\emptyset}\left(\sum_{i\in\mathbf{U}_{lq}}\lambda_i + \sum_{k\in\mathcal{L}}|R_{lq}(k)|\cdot\alpha_q^*(k)\right) \\ &\quad + C_l - \sum_{\mathbf{O}_{lq}\neq\emptyset} G_{lq} - \sum_{\mathbf{O}_{lq}=\emptyset}\left(\sum_{i\in\mathbf{U}_{lq}}\lambda_i + \sum_{k\in\mathcal{L}}|R_{lq}(k)|\cdot\alpha_q^*(k)\right) \\ &= C_l \end{aligned} \tag{3.24}$$

Hence, as the bandwidth allocation satisfies inter-class fairness, by Lemma 3.1 ϕ_l^* is maximal.

Therefore, we have shown that in both Case 1 and Case 2, we obtain a maximal bandwidth allocation that satisfies the condition for inter-class fairness, thus proving the lemma. □

3.2 Bandwidth Allocations with Multi-level Regulation

In this section, we consider multi-level bandwidth regulation on an ATM network. We shall concern ourselves only with the virtual overlay network of *Connectionless Servers* (CLSs) which are used to support connectionless traffic. The model of the network assumed in this discussion is illustrated in Figure 3.3. In §3.1, the term "network" denoted a traditional packet-switched network comprising a set of gateways connected by physical links. In this section, we use the term "network" for the virtual network of VPs carrying connectionless traffic. When we need to refer to the actual physical ATM network, we shall state so explicitly.

Theorem 3.1 establishes a formal definition of short-term bandwidth regulation that achieves inter-class and intra-class fairness. We now proceed to develop a similar definition of medium-term bandwidth regulation that achieves so-called *class-level fairness*.

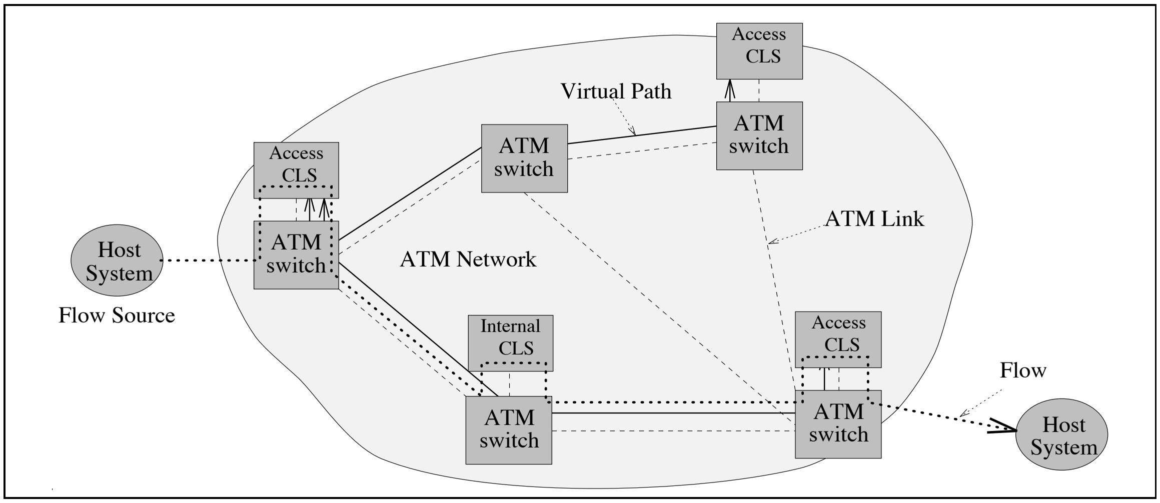


Figure 3.3: ATM Network Supporting Connectionless Traffic.

In this section, we will continue to employ the notation used in §3.1, with some modifications. l now denotes a VP connecting two CLSs, or a CLS and an end-system. Also, the class guarantee of a flow class p at a link l, G_{lp}, is no longer expressed in absolute terms, i.e., in terms of bytes per second, but as a fraction of the total bandwidth allocated to the VP l. The new notations used in this section are defined below.

Let $\Gamma_{lp} = \sum_{i \in \Delta_{lp}} \gamma_i$ be the total throughput of a flow class p at a VP l. Let M_{lp} be the minimum bandwidth guarantee that any flow class p receives at a VP l with $0 \leq M_{lp} \leq 1$ and $G_{lp} < M_{lp}$ if and only if $\Gamma_{lp} < M_{lp} C_l$. We define the *class surplus* of a VP l, φ_l, to be a number such that the maximum guarantee a flow class can receive in excess of its minimum guarantee, M_{lp} is given by $\varphi_l M_{lp}$. We now proceed to define *class level fairness*. As noted before, each flow class is allocated a guarantee. If a flow class is unable to utilize its guarantee fully, the guarantee is reduced and the unutilized bandwidth is divided proportionately among all flow classes which can utilize its bandwidth. It is important to note the distinction between *class level fairness* and *inter-class fairness*. In the former, the guarantee G_{lp} is modified, while in the latter, it is the available bandwidth C_{lp} that is changed. Formally, we may define class level fairness as follows:

Definition 3.4 *A bandwidth allocation is said to provide* <u>class level fairness</u> *if for each VP*

$l \in \mathcal{L}$ there exists a class surplus value φ_l such that for all $p \in \mathcal{P}_l$

$$G_{lp} = \min\left(\frac{\Gamma_{lp}}{C_l}, M_{lp}(1+\varphi_l)\right)$$

The *maximal class surplus* at a link is the maximum value of φ_l which does not leave any bandwidth unallocated to the class guarantees as long as there exists a flow class whose total offered load exceeds its guarantee. In Theorem 3.2 we state that an allocation of class guarantees can be made with class level regulation and maximal class surplus.

Theorem 3.2 *Given a network with topology $\mathcal{T}$ and load set Δ, there exists a unique allocation of class guarantees that provides class level fairness with maximal class surplus values φ_l^*. The class guarantees and the maximal class surplus values are determined by a solution to the following equation system.*

$$G_{lp} = \begin{cases} \frac{\Gamma_{lp}}{C_l} & \textit{if } p \in \mathbf{\Psi}_l \\ M_{lp}(1+\varphi_l^*) & \textit{otherwise} \end{cases} \tag{3.25}$$

and

$$\varphi_l^* = \begin{cases} \infty & \textit{if } \mathbf{\Xi}_l = \emptyset \\ \dfrac{C_l - \sum_{q \in \mathbf{\Xi}_l} M_{lq}C_l - \sum_{q \in \mathbf{\Psi}_l} \Gamma_{lq}}{\sum_{q \in \mathbf{\Xi}_l} M_{lq}C_l} & \textit{otherwise} \end{cases} \tag{3.26}$$

subject to the side condition:

$$C_l - \sum_{q \in \mathbf{\Xi}_l} M_{lq}C_l - \sum_{q \in \mathbf{\Psi}_l} \Gamma_{lq} \geq 0 \tag{3.27}$$

where the sets $\mathbf{\Psi}_l$ and $\mathbf{\Xi}_l$ are defined as follows:

$$\mathbf{\Psi}_l = \{p \mid \Gamma_{lp} < G_{lp}C_l\}$$

$$\mathbf{\Xi}_l = \{p \mid \Gamma_{lp} \geq G_{lp}C_l\}$$

The sets $\boldsymbol{\Psi}_l$ and $\boldsymbol{\Xi}_l$ correspond to the sets $\mathbf{U}_{lp}$ and $\mathbf{O}_{lp}$ in Theorem 3.1. $\boldsymbol{\Psi}_l$ is the set of flow classes which are unable to utilize their guarantees, and $\boldsymbol{\Xi}_{lp}$ is the set of flow classes whose throughputs exceeds or at least equals their guarantees.

Short-term and medium-term bandwidth regulations are concerned only with the virtual network of CLSs, as described in §1.2, and do not require knowledge about the connection-oriented traffic types in the underlying ATM network. However, long-term regulation, which is concerned with the allocation of bandwidth to the virtual network of VPs that interconnect the CLSs, requires knowledge about the connection-oriented traffic in the ATM network. We assume that there is a CLS attached to every ATM switch and that every ATM link has exactly one VP dedicated to connectionless traffic. The capacity of the ATM link l is denoted by $C_{l_{ATM}}$. The VP carrying connectionless traffic is given a minimum bandwidth guarantee of $M_{l_{CL}}$. The actual bandwidth allocated to the VP, which may be different from $M_{l_{CL}}$, is denoted by $C_{l_{CL}}$. Lastly, $\Lambda_{l_{CL}}$ and $\Lambda_{l_{CO}}$ denote the total offered loads of connectionless flows and connection-oriented flows, respectively, on the ATM link l.

We now present a formal definition of *long-term regulation*. The objective of long-term regulation is to provide priority to connection-oriented traffic over connectionless traffic at an ATM link, while guaranteeing a minimum bandwidth allocation to connectionless traffic.

Definition 3.5 *A bandwidth allocation to connectionless traffic is said to provide* <u>long-term regulation</u> *if for each ATM link* l*, the bandwidth allocated to the VP carrying connectionless traffic is given by*

$$C_{l_{CL}} = \max(\ \min(\ \Lambda_{l_{CL}},\ M_{l_{CL}}),\ \min(\ \Lambda_{l_{CL}},\ C_{l_{ATM}} - \Lambda_{l_{CO}}))$$

Note here that the variable $C_{l_{CL}}$ in Definition 3.5 is equivalent to the variable C_l used in Theorems 3.1 and 3.2.

Theorem 3.1 gives a mathematical model for a short-term bandwidth regulation mechanism that achieves the goals of inter-class and intra-class fairness. Theorem 3.2 develops a similar mathematical model for a medium-term bandwidth regulation which has as its goal class level fairness. Finally, Definition 3.5 presents a formal definition of long-term regulation that gives priority to connection-oriented traffic. In the next section, we use Theorems 3.1 and 3.2, and Definition 3.5 to derive a protocol that implements long-term, medium-term, and short-term regulation. We will show that the multi-level bandwidth regulation can be implemented by a relatively simple protocol.

3.3 The Single Class Case

In this section we present a formal proof of a special case of Theorem 3.1, with only one traffic class. Theorem 3.3, which is a restatement of Theorem 3.1 for the special case of a network with only one defined class, (i.e., $|\mathcal{P}| = 0$), provides a mechanism for a bandwidth allocation that provides only intra-class fairness.

Theorem 3.3 *Given a network with topology $\mathcal{T}$ with offered load set $\mathbf{\Lambda}$. Then there exists a unique bandwidth allocation which is intra-class fair with maximal shares $\alpha_p^*(l)$. The maximal shares are determined by a solution of the following equation system* [2]*.*

$$\alpha^*(l) = \begin{cases} \dfrac{C_l - \Theta_l}{|\mathbf{O}_l|} & \textit{if } \mathbf{O}_l \neq \emptyset \\ \infty & \textit{otherwise} \end{cases} \tag{3.28}$$

subject to the side condition

$$C_l - \Theta_l \geq 0 \tag{3.29}$$

where $\Theta_l = \sum_{i \in \mathbf{U}_l} \lambda_i + \sum_{k \in \mathcal{L}} |\mathbf{R}_l(k)| \cdot \alpha^*(k)$ *and the sets* $\mathbf{U}_l$, $\mathbf{R}_l$, *and* $\mathbf{O}_l$ *are defined for all* $l \in \mathcal{L}_l$ *as*

$$\mathbf{U}_l = \{i \in \Delta_l \mid \alpha^*(l) \geq \lambda_i \ , \ i \notin \bigcup_{k \in \mathcal{L}} \mathbf{R}_l(k)\} \tag{3.30}$$

$$\mathbf{O}_l = \{i \in \Delta_l \mid l = l_i^* \ , \ \alpha^*(l) < \lambda_i\} \tag{3.31}$$

$$\mathbf{R}_l(k) = \{i \in \Delta_l \mid k = l_i^* \ , \ \alpha^*(k) < \lambda_i\} \qquad \textit{for } k \neq l \tag{3.32}$$

[2] In equation (3.28) $|X|$ denotes the cardinality of a set X.

We will prove the theorem by presenting an iterative algorithm to determine a solution of the equation system. We show that the algorithm terminates after a fixed number of iterations. We then show that the solution obtained by the algorithm is a solution to the equation system. Finally, we show that the solution to the equation system is

1. maximally fair and waste-free.

2. unique.

Algorithm 1 repeatedly improves the α-values for all links still in consideration during each iteration. In the i^{th} iteration, the link with the smallest α-value, l_i is eliminated from consideration. At this point, we set $\alpha^*(l_i) = \alpha^{(i)}(l_i)$, where $\alpha^{(i)}(l_i)$ is the α-value of link l_i in the i^{th} iteration.

We shall prove the theorem after proving the following lemmas.

Lemma 3.5 $\alpha^{(1)}(1) \leq \alpha^{(2)}(1) \leq \cdots \leq \alpha^{(m)}(1)$ *where* $\alpha^{(i)}(l)$, $1 \leq i \leq n$ *is obtained in step 12 of Algorithm 1, for any link* l *such that* $\alpha^{(i)}(l)$ *is not chosen as the minimum* α*-value in any of the* m *iterations.*

Proof:

We select a link l such that for all $i \leq m$, $\alpha^{(i)}(l) > \min\limits_{l' \in \mathcal{L}^{(i)}} \left(\alpha^{(i)}(l')\right)$. In the n^{th} iteration of the algorithm, the value of $\alpha^{(n)}(l)$ as obtained in step 12 depends on the value of $\mathbf{U}_l^{(n)}$ obtained in step 11. There are three different cases depending on the values of $\mathbf{U}_l^{(n)}$ and $\mathbf{U}_l^{(n+1)}$. We show that in every case the lemma holds.

Case 1: $\mathbf{U}_l^{(n)} = \Delta_l^{(n)}$.

From step 11 of the algorithm, it follows that the following condition must hold:

$$\max_{i \in \Delta_l^{(n)}} \lambda_i \leq C_l - \sum_{j \in \Delta_l^{(n)}} \lambda_j - \sum_{k=1}^{n-1} \left|\mathbf{R}_l^{(k)}(l_k)\right| \cdot \alpha_{l_k}^{(k)} \tag{3.33}$$

In step 12, the algorithm sets $\alpha^{(n)}(l) = \infty$.

Now, from steps 17 and 18 we have,

$$C_l - \sum_{j \in \Delta_l^{(n)}} \lambda_j - \sum_{k=1}^{n-1} \left|\mathbf{R}_l^{(k)}(l_k)\right| \cdot \alpha_{l_k}^{(k)} = C_l - \sum_{j \in \Delta_l^{(n+1)}} \lambda_j - \sum_{k=1}^{n-1} \left|\mathbf{R}_l^{(k)}(l_k)\right| - \sum_{i \in \mathbf{R}_l^{(n)}(l_n)} \lambda_i \tag{3.34}$$

It follows from the selection of $\mathbf{R}_l^{(n)}(l_n)$ that

$$\sum_{i \in \mathbf{R}_l^{(n)}(l_n)} \lambda_i > \left|\mathbf{R}_l^{(n)}(l_n)\right| \cdot \alpha^{(n)}(l_n) \tag{3.35}$$

By combining (3.34) and (3.35) we get

$$\begin{aligned} C_l - \sum_{j \in \Delta_l^{(n)}} \lambda_j - \sum_{k=1}^{n-1} \left|\mathbf{R}_l^{(k)}(l_k)\right| \cdot \alpha_{l_k}^{(k)} &< C_l - \sum_{j \in \Delta_l^{(n+1)}} \lambda_j - \sum_{k=1}^{n-1} \left|\mathbf{R}_l^{(k)}(l_k)\right| - \left|\mathbf{R}_l^{(n)}(l_n)\right| \cdot \alpha^{(n)}(l_n) \\ &< C_l - \sum_{j \in \Delta_l^{(n+1)}} \lambda_j - \sum_{k=1}^{n} \left|\mathbf{R}_l^{(k)}(l_k)\right| \cdot \alpha_{l_k}^{(k)} \end{aligned} \tag{3.36}$$

Since $\Delta_l^{(n+1)} \subseteq \Delta_l^{(n)}$, it follows with (3.36) that

$$\max_{i \in \Delta_l^{(n+1)}} \lambda_i \leq C_l - \sum_{j \in \Delta_l^{(n+1)}} \lambda_j - \sum_{k=1}^{n} \left|\mathbf{R}_l^{(k)}(l_k)\right| \cdot \alpha_{l_k}^{(k)} \tag{3.37}$$

In the $(n+1)^{th}$ iteration, in step 11, because of (3.37), $\mathbf{U}_l^{(n+1)} = \Delta_l^{(n+1)}$. Therefore, in step 12, $\alpha^{(n+1)}(l) = \infty$.

Case 2: $\mathbf{U}_l^{(n)} \neq \Delta_l^{(n)}$ and $\mathbf{U}_l^{(n+1)} = \Delta_l^{(n+1)}$.

It follows from step 12 that $\alpha^{(n)}(l) \leq \infty$ and $\alpha^{(n+1)}(l) = \infty$. Therefore $\alpha^{(n)}(l) \leq \alpha^{(n+1)}(l)$.

Case 3: $\mathbf{U}_l^{(n)} \neq \Delta_l^{(n)}$ and $\mathbf{U}_l^{(n+1)} \neq \Delta_l^{(n+1)}$.

In the n^{th} iteration of the algorithm, in step 12, we get

$$\alpha^{(n)}(l) = \frac{C_l - \sum_{i \in \mathbf{U}_l^{(n)}} \lambda_i - \sum_{k=1}^{n-1} \left|\mathbf{R}_l^{(k)}(l_k)\right| \cdot \alpha^{(k)}(l_k)}{\left|\Delta_l^{(n)}\right| - \left|\mathbf{U}_l^{(n)}\right|} \tag{3.38}$$

In the next iteration, in step 12

$$\begin{aligned} \alpha^{(n+1)}(l) &= \frac{C_l - \sum_{i \in \mathbf{U}_l^{(n+1)}} \lambda_i - \sum_{k=1}^{n} \left|\mathbf{R}_l^{(k)}(l_k)\right| \cdot \alpha^{(k)}(l_k)}{\left|\Delta_l^{(n+1)}\right| - \left|\mathbf{U}_l^{(n+1)}\right|} \\ &:= \frac{\mathbf{N}}{\mathbf{D}} \end{aligned} \tag{3.39}$$

With (3.40), equation (3.38) is rewritten as

$$\begin{aligned}
\alpha^{(n)}(l) &= \frac{C_l - \sum_{i\in\mathbf{U}_l^{(n+1)}} \lambda_i - \sum_{k=1}^{n-1} \left|\mathbf{R}_l^{(k)}(l_k)\right| \cdot \alpha^{(k)}(l_k) - \sum_{i\in\mathbf{U}_l^{(n)}\cap\mathbf{R}_l^{(n)}(l_n)} \lambda_i}{\left|\Delta_l^{(n+1)}\right| - \left|\mathbf{U}_l^{(n+1)}\right| + \left|\mathbf{R}_l^{(n)}(l_n)\right| + \left|\left\{i \mid i \in \mathbf{U}_l^{(n)} \cap \mathbf{R}_l^{(n)}(l_n)\right\}\right|} \\
&= \frac{C_l - \sum_{i\in\mathbf{U}_l^{(n+1)}} \lambda_i - \sum_{k=1}^{n} \left|\mathbf{R}_l^{(k)}(l_k)\right| \cdot \alpha^{(k)}(l_k) + \left|\mathbf{R}_l^{(n)}(l_n)\right| \cdot \alpha^{(n)}(l_n) - \sum_{i\in\mathbf{U}_l^{(n)}\cap\mathbf{R}_l^{(n)}(l_n)} \lambda_i}{\left|\Delta_l^{(n+1)}\right| - \left|\mathbf{U}_l^{(n+1)}\right| + \left|\mathbf{R}_l^{(n)}(l_n)\right| + \left|\left\{i \mid i \in \mathbf{U}_l^{(n)} \cap \mathbf{R}_l^{(n)}(l_n)\right\}\right|} \\
&= \frac{\mathbf{N} + \left|\mathbf{R}_l^{(n)}(l_n)\right| \cdot \alpha^{(n)}(l_n)}{\mathbf{D} + \left|\mathbf{R}_l^{(n)}(l_n)\right| + \left|\{i \mid i \in \mathbf{U}_l^{(n)} \cap \mathbf{R}_l^{(n)}(l_n)\}\right|} \\
&= \frac{\mathbf{N} + \left|\mathbf{R}_l^{(n)}(l_n)\right| \cdot \left(\alpha^{(n)}(l_n) - \alpha^{(n)}(l)\right) - \left|\{i \mid i \in \mathbf{U}_l^{(n)} \cap \mathbf{R}_l^{(n)}(l_n)\}\right| \cdot \alpha^{(n)}(l)}{\mathbf{D}} \qquad (3.40)
\end{aligned}$$

As $\alpha^{(n)}(l_n) \leq \alpha^{(n)}(l)$ it follows from equation (3.40) that

$$\begin{aligned}
\alpha^{(n)}(l) &< \frac{\mathbf{N}}{\mathbf{D}} \\
&= \alpha^{(n+1)}(l) \qquad (3.41)
\end{aligned}$$

Therefore, the lemma is proved.

Lemma 3.6 *Algorithm 1 gives a solution to the equation system in (3.28) – (3.32).*

Proof:

We will prove the lemma by induction over the number of iterations in Algorithm 1. In the following, we assume without loss of generality that link l_k is selected in the k^{th} iteration of Step 12 of the algorithm.

- First Iteration :

Since in the first iteration, $\mathbf{R}^{(0)}(l_k) = \emptyset$, we obtain in Steps 11 and 12 of Algorithm 1:

$$\mathbf{U}_l^{(1)} = \begin{cases} \Delta_l & \text{if } \max_{i \in \Delta_l} \lambda_i \leq C_l - \sum_{j \in \Delta_l} \lambda_j \\ \{i \in \Delta_l \mid \lambda_i \leq \dfrac{C_l - \sum_{j \in \Delta_l,\, \lambda_j < \lambda_i} \lambda_j}{|\Delta_l| - |\{j \in \Delta_l \mid \lambda_j \leq \lambda_i\}|}\} & \text{otherwise} \end{cases} \tag{3.42}$$

$$\alpha^{(1)}(l) = \begin{cases} \infty & \text{if } \mathbf{U}_l^{(1)} = \Delta_l \\ \dfrac{C_l - \sum_{i \in \mathbf{U}_l^{(1)}} \lambda_i}{|\Delta_l| - |\mathbf{U}_l^{(1)}|} & \text{otherwise} \end{cases} \tag{3.43}$$

In (3.42) and (3.43), we observe that for $\lambda_{\hat{i}} = \max_{i \in \mathbf{U}_l^{(1)}} \lambda_i$ it holds that

$$\sum_{j \in \Delta_l,\, \lambda_j < \lambda_{\hat{i}}} \lambda_j = \sum_{j \in \mathbf{U}_l^{(1)}} \lambda_j \tag{3.44}$$

$$\{j \in \Delta_l \mid \lambda_j \leq \lambda_{\hat{i}}\} = \mathbf{U}_l^{(1)} \tag{3.45}$$

Thus, for all sets $\mathbf{U}_l^{(1)}$ we get:

$$\mathbf{U}_l^{(1)} = \{i \in \Delta_l \mid \alpha^{(1)}(l) \geq \lambda_i\} \tag{3.46}$$

Next we consider link l_1 as selected in Step 12 and set

$$\alpha^*(l_1) := \alpha^{(1)}(l_1) \tag{3.47}$$

From the previous lemma, it follows that $\alpha^*(l_1) = \min_{l \in \mathcal{L}}(\alpha^*(l))$, that is, link l_1 is the link with the smallest share. We obtain in (3.32):

$$\mathbf{R}_l(l_1) = \emptyset \qquad \text{for all } l \neq l_1 \tag{3.48}$$

$$\mathbf{O}_{l_1} = \Delta_{l_1} \setminus \mathbf{U}_{l_1}^{(1)} \tag{3.49}$$

All flows $i \in \Delta_{l_1}$ which satisfy $\lambda_i > \alpha^*(l_1)$ have their bottleneck at link l_1. Thus, we obtain for all $l \neq l_1$ that

$$\mathbf{R}_l(l_1) = \mathbf{R}^{(1)}(l) \tag{3.50}$$

Finally, (3.46) and (3.47) yield:

$$\mathbf{U}_{l_1} = \mathbf{U}_{l_1}^{(1)} \tag{3.51}$$

We have found a solution for $\alpha^*(l_1)$, $\mathbf{U}_{l_1}$, and $\mathbf{O}_{l_1}$, and $\mathbf{R}_l(l_1)$ in (3.28) – (3.32).

- <u>nth Iteration :</u>

Assume that Algorithm 1 has executed $(n-1)$ iterations, and found sets

$$\mathbf{R}_{l_k}^{i}(l_i) \qquad \text{for } 1 \leq i < n \tag{3.52}$$

and values

$$\alpha^{(1)}(l_1) \leq \alpha^{(2)}(l_2) \leq \ldots \alpha^{(n)}(l_{n-1}) \tag{3.53}$$

Algorithm 1 has found the following solutions for all $1 \leq i < n$:

$$\alpha^*(l_i) = \alpha^{(i)}(l_i) \tag{3.54}$$

$$\mathbf{U}_{l_i} = \mathbf{U}_{l_i}^{(i)} \tag{3.55}$$

$$\mathbf{R}_{l_k}(l_i) = \mathbf{R}_{l_k}^{(i)}(l_i) \qquad k > i \tag{3.56}$$

Next we show that Algorithm 1 correctly calculates $\alpha^*(l_n)$, $\mathbf{U}_{l_n}$, and $\mathbf{R}_{l_k}(l_n)$ for all $k > n$. Also, we will show that $\alpha^*(l_n)$ satisfies the condition from the previous iterations, that is $\alpha^*(l_n) \geq \alpha^*(l_{n-1})$.

Note that from Step 18 in the previous $(n-1)$ iterations, we obtain for all $k \geq n$:

$$\Delta^{(k)}(l_k) = \Delta(l_k) \setminus \bigcup_{j<n} \mathbf{R}_{l_k}^{(j)}(l_j) \tag{3.57}$$

$$= \Delta(l_k) \setminus \bigcup_{j<n} \mathbf{R}_{l_k}(l_j) \tag{3.58}$$

From Step 11 and Step 12 in Algorithm 1, we observe that $\lambda_{\hat{i}} = \max_{i \in \mathbf{U}_l^{(n)}} \lambda_i$ satisfies:

$$\sum_{j \in \Delta_l^{(n)},\, \lambda_j < \lambda_{\hat{i}}} \lambda_j = \sum_{j \in \mathbf{U}_l^{(n)}} \lambda_j \tag{3.59}$$

$$\{j \in \Delta_l^{(n)} \mid \lambda_j \leq \lambda_{\hat{i}}\} = \mathbf{U}_l^{(n)} \tag{3.60}$$

Therefore, we can write for each $l \in \mathcal{L}^{(n)}$:

$$\mathbf{U}_l^{(n)} = \{i \in \Delta_l^{(n)} \mid \alpha^{(n)}(l) > \lambda_i\} \tag{3.61}$$

With (3.58), the last equation can be rewritten as:

$$\mathbf{U}_{l_k}^{(n)} = \{i \in \Delta_{l_k} \mid \alpha^{(n)}(l_k) > \lambda_i, k \geq n\} \tag{3.62}$$

Now consider l_n, selected in Step 12 of Algorithm 2, and set

$$\alpha^*(l_n) = \alpha^{(n)}(l_n) \tag{3.63}$$

Since, by Lemma 3.5, $\alpha^{(n-1)}(l_n) \leq \alpha^{(n)}(l_n)$, we obtain our first claim, that is,

$$\alpha^*(l_n) \geq \alpha^*(l_{n-1}) \tag{3.64}$$

As $\alpha^*(l_n) = \min_{l \in \mathcal{L}^{(n)}} \left(\alpha^{(n)}(l)\right)$, i.e., link l_n has the smallest share among the remaining links, we obtain

$$\mathbf{R}_{l_n}^{(n)}(l) = \emptyset \qquad \text{for all } l \in \mathcal{L}^{(n)} \tag{3.65}$$

Also, it follows with (3.58) that

$$\begin{aligned} \mathbf{O}_{l_n} &= \Delta_{l_n}^{(n)} \setminus \mathbf{U}_{l_n}^{(n)} && (3.66) \\ &= (\Delta_{l_n} \setminus \bigcup_{j<n} \mathbf{R}_{l_n}(l_j)) \setminus \mathbf{U}_{l_n}^{(n)} && (3.67) \end{aligned}$$

From (3.61) it follows with (3.63) that

$$\mathbf{U}_{l_n} = \mathbf{U}_{l_n}^{(n)} \tag{3.68}$$

Thus we have found a solution for $\alpha^*(l_n)$, $\mathbf{U}_{l_n}$, and $\mathbf{O}_{l_n}$ in (3.28) – (3.32). With (3.64) we have verified that l_n satisfies the condition on $\alpha^*(l_n)$ from the previous iterations. If $n \neq |\mathcal{L}|$, all flows $i \in \Delta_{l_n}^{(n)}$ with $\lambda_i > \alpha^*(l_n)$ have their bottleneck at link l_n. Thus, we can verify that the selected $\alpha^*(l_n)$ is such that for all $l \in \mathcal{L} \setminus \{l_n\}$ we have

$$\mathbf{R}_l(l_n) = \mathbf{R}_l^{(n)}(l_n) \tag{3.69}$$

The induction shows that Algorithm 1 constructs a solution of the equation system in (3.28) – (3.32). □

$\mathcal{L}^{(0)} := \mathcal{L}$;
for each $l \in \mathcal{L}$ **do**
  $\Delta^{(1)}(l) := \Delta(l)$;
  for each $k \in \mathcal{L}$, $k \neq l$ **do**
    $\mathbf{R}^{(0)}(l) := \emptyset$;
  endfor
endfor
for $n := 1$ **until** $n = |\mathcal{L}|$ **do**
  $\mathcal{L}^{(n)} := \mathcal{L}^{(n-1)}$;
  for each $l \in \mathcal{L}^{(n)}$ **do**

$$\mathbf{U}_l^{(n)} = \begin{cases} \Delta_l^{(n)} & \textbf{if } \max_{i \in \Delta_l^{(n)}} \lambda_i \leq C_l - \sum_{j \in \Delta_l^{(n)}} \lambda_j - \sum_{k=1}^{n-1} |\mathbf{R}_l^{(k)}(l_k)| \cdot \alpha_{l_k}^{(k)} \\ \left\{ i \in \Delta_l^{(n)} \;\middle|\; \lambda_i \leq \dfrac{C_l - \sum_{j \in \Delta_l^{(n)}, \lambda_j \leq \lambda_i} \lambda_j - \sum_{k=1}^{n-1} |\mathbf{R}_l^{(k)}(l_k)| \cdot \alpha_{l_k}^{(k)}}{|\Delta_l^{(n)}| - |\{j \in \Delta_l^{(n)} \mid \lambda_j < \lambda_i\}|} \right\} & \textbf{otherwise} \end{cases}$$

$$\alpha^{(n)}(l) = \begin{cases} \infty & \textbf{if } \mathbf{U}_l^{(n)} = \Delta_l^{(n)} \\ \dfrac{C_l - \sum_{i \in \mathbf{U}_l^{(n)}} \lambda_i - \sum_{k=1}^{n-1} |\mathbf{R}_l^{(k)}(l_k)| \cdot \alpha_{l_k}^{(k)}}{|\Delta_l^{(n)}| - |\mathbf{U}_l^{(n)}|} & \textbf{otherwise} \end{cases}$$

  endfor

  Select index l_n with $\alpha^{(n)}(l_n) = \min_{l \in \mathcal{L}^{(n)}} \left(\alpha^{(n)}(l)\right)$;
  $\mathcal{L}^{(n)} := \mathcal{L}^{(n)} \setminus \{l_n\}$;
  for each $l \in \mathcal{L}^{(n)}$ **do**
    $\mathbf{R}_l^{(n)}(l_n) = \{i \in \Delta_l^{(n)} \mid \alpha^{(n)}(l_n) < \lambda_i, i \in \Delta_{l_n}^{(n)}\}$;
    $\Delta^{(n+1)}(l) := \Delta^{(n)}(l) \setminus \mathbf{R}_l^{(n)}(l_n)$;
  endfor
endfor

Algorithm 1.

Lemma 3.7 *The equation system in (3.28) – (3.32) has an unique solution.*

Proof:

Let there exist two maximally fair solutions to the equation system (3.28) – (3.32), which are denoted by $\{\alpha_1^*(l) \mid l \in \mathcal{L}\}$ and $\{\alpha_2^*(l) \mid l \in \mathcal{L}\}$, respectively. The corresponding throughputs obtained by a flow i is denoted by γ_i^1 and γ_i^2 respectively. As they are different solutions, there must exist a flow i such that $\gamma_i^1 \neq \gamma_i^2$. Without loss of generality, we assume that the links $l_j \in \mathcal{L}$ are ordered such that $\alpha_1^*(l_j) \leq \alpha_1^*(l_{j+1})$, $1 \leq j < n$, where $n = |\mathcal{L}|$. We show by induction on the links l_j that all flows receive identical allocations in both solutions.

Induction Basis: $i \notin \Delta_{l_1}$

We assume that there is a flow $i \in \Delta_{l_1}$, such that $\gamma_i^1 \neq \gamma_i^2$. As $\alpha_1^*(l_j) \leq \alpha_1^*(l_{j+1})$, $1 \leq j < n$, i must either be an underloaded flow, or overloaded at link l_1, according to the first solution.

Case 1: i is an underloaded flow in the first solution, i.e., $\gamma_i^1 = \lambda_i$.

Obviously, $\gamma_i^2 < \gamma_i^1 < \lambda_i$ as a flow cannot exceed its offered load.

Case 2: i is an overloaded flow in the first solution, i.e., $\gamma_i^1 = \alpha_1^*$.

In this case there must exist a flow i such that $\gamma_i^2 < \gamma_i^1$, because if for all $i \in \Delta_{l_1}$, $\gamma_i^2 \geq \gamma_i^1$, then $\sum_{i\in\Delta_{l_1}} \gamma_i^2 \geq \sum_{i\in\Delta_{l_1}} \gamma_i^1$. But as $\sum_{i\in\Delta_{l_1}} \gamma_i^1 = C_{l_1}$, this implies that $\sum_{i\in\Delta_{l_1}} \gamma_i^2 \geq C_{l_1}$, which is contrary to equation 3.3. Obviously, $\gamma_i^2 < \gamma_i^1 < \lambda_i$.

Therefore in both cases, there exists a flow i such that $\gamma_i^2 < \gamma_i^1$. Let $\gamma_i^2 = \alpha_2^*(l_k)$, $1 \leq k \leq n$. Now, note that since $\gamma_i^1 \leq \alpha_1^*(l_1)$, $\alpha_2^*(l_k) \leq \alpha_1^*(l_1)$. By the assumption on the ordering of the links, $\alpha_1^*(l_m) \geq \alpha_1^*(l_1) > \alpha_2^*(l_k)$ for all $1 < m \leq n$. Now, consider a flow $h \in \Delta_{l_k}$. If $h \in \mathbf{U}_{l_k}^2$, then $\gamma_h^1 = \gamma_h^2 = \lambda_h$. Otherwise, $\gamma_h^1 > \gamma_h^2$. Therefore, $\sum_{h\in\Delta_{l_k}} \gamma_h^2 < \sum_{h\in\Delta_{l_k}} \gamma_h^1 = C_{l_k} < C_{l_k}$. Hence, the solution $\alpha_2^*(l)$, $l \in \mathcal{L}$ is not maximal. Therefore $\alpha_1^*(l_1) = \alpha_2^*(l_1)$, and for all flows $i \in \Delta_{l_1}$, $\gamma_i^2 = \gamma_i^1$, i.e., all flows passing through link l_1 obtain equal throughput in both solutions.

Induction Hypothesis: $i \notin \Delta_{l_j}$, $j < J \leq n$

We assume that all flows passing through link l_j, $j < J$, obtain equal throughput in both solutions, i.e., $\alpha_1^*(l_j) = \alpha_2^*(l_j)$, and for all flows $i \in \Delta_{l_j}$, $\gamma_i^2 = \gamma_i^1$.

Induction Step: $i \notin \Delta_{l_j}$, $j = J$

We assume that there exists a flow $i \in \Delta_{l_J}$ such that $\gamma_i^2 \neq \gamma_i^1$.

Note that by assumption on the ordering of the links, and the definition of $\mathbf{R}^1_{l_J}(l_k)$, $k < J$. Hence by the induction hypothesis, for all flows $i \in \Delta_{l_k}$, $\gamma_i^2 = \gamma_i^1$. Therefore, the flow i has to be a member of exactly one of the sets $\mathbf{O}^1_{l_J}$ or $\mathbf{U}^1_{l_J}$.

Case 1: $i \in \mathbf{U}^1_{l_J}$, i.e., $\gamma_1^i = \lambda_i^1$.

Obviously, $\gamma_i^2 < \gamma_i^1 = \lambda_i$, as a flow cannot exceed its offered load.

Case 2: $i \in \mathbf{O}^1_{l_J}$.

We claim that there must exist a flow i such that $\gamma_i^2 < \gamma_i^1$, because, if for all $k \in \Delta_{l_J}$, $\gamma_k^2 \geq \gamma_k^1$ and as by assumption $\gamma_i^2 \neq \gamma_i^1$, then $\sum_{i\in\Delta_{l_J}} \gamma_i^2 > \sum_{i\in\Delta_{l_J}} \gamma_i^1$. But, as $\sum_{i\in\Delta_{l_J}} \gamma_i^1 = C_{l_J}$, this implies that $\sum_{i\in\Delta_{l_J}} \gamma_i^2 > C_{l_J}$, which is contrary to equation (3.29). Therefore, there always exist a flow i such that $\gamma_i^2 < \gamma_i^1$.

Let link l_m be the bottleneck of flow i in the second solution. Therefore, we have $\gamma_i^2 = \alpha_2^*(l_m)$. Note that by the induction hypothesis, if $i \in \Delta_{l_k}$, $k < J$, then $\gamma_i^2 = \gamma_i^1$. Therefore, $m > J$. Since $\gamma_i^2 < \gamma_i^1$, therefore $\alpha_2^*(l_m) < \alpha_1^*(l_j)$. By the assumption on the ordering of the links, $\alpha_2^*(l_m) < \alpha_1^*(l_k)$ for $J \leq k \leq n$.

Now consider a flow h passing through link l_m, i.e., $h \in \Delta_{l_m}$. If the flow h also passes through a link l_k, $k < J$, then by the induction hypothesis, $\gamma_h^2 = \gamma_h^1$. Otherwise, $\gamma_h^2 < \gamma_h^1$. Therefore, for all flows $h \in \Delta_{lm}$, $\gamma_h^2 \leq \gamma_h^1$. Also, we have already obtained that $\gamma_i^2 < \gamma_i^1$. Hence, it follows that $\sum_{h\in\Delta_{l_m}} \gamma_h^2 < \sum_{h\in\Delta_{l_m}} \gamma_h^1 = C_{l_m}$. But, then the second bandwidth allocation is not maximal. Hence, for all $i \in \Delta_{l_j}$, $\gamma_i^2 = \gamma_i^1$.

Therefore, there can be only one maximal solution to the equation system. □

Proof of Theorem 3.3:

In Lemma 3.6, we have proved that the equation system in (3.28)–(3.32) has a solution. As equations (3.28)–(3.32) are a special instance of the equation system (3.1)–(3.6), by Lemma 3.3, a solution of the equation system (3.28)–(3.32) will be maximally fair. Finally, in Lemma 3.7, we show that the solution of the equations (3.28)–(3.32) is unique. Thus, we have proved Theorem 3.3. □

Chapter 4

Experimental Results

In Chapter 3, we developed a bandwidth allocation scheme that described a fair and waste-free allocation for a complex internetwork. We then extended the scheme to describe multi-level bandwidth allocation in an ATM network. In §4.1, we present a simple protocol, called the p–protocol, to implement the bandwidth allocation algorithm in the context of a packet-switched internetwork. The protocol achieves both inter-class regulation among different traffic classes as well as intra-class regulation among flows belonging to each traffic class. We demonstrate the effectiveness of the protocol with the aid of a simulation experiment. In §4.2, we extend the protocol to perform multi-level bandwidth allocation in an ATM network. We present a set of simulation experiments to show that the modified protocol, referred to as the c–protocol, effectively achieves the desired multi-level regulation.

4.1 A Bandwidth Regulation Protocol for Packet-switched Networks

The p–protocol is designed to be used in a packet-switched internetwork as illustrated in Figure 3.1. The internetwork consists of a set of gateways connected by network links. The gateways may be either *internal* gateways or *access* gateways. Internal gateways are connected only to other gateways, while access gateways are linked to other gateways as well as to host systems, either directly, or through a local area network. The p–protocol is implemented exclusively at the gateways, except for a rate controller at the sources. The protocol is described in detail in §4.1.1 and the simulation results are detailed in §4.1.2.

4.1.1 The p–protocol

The p–protocol implements intra-class and inter-class bandwidth allocations developed in Chapter 3 by providing a distributed mechanism for calculating the maximal shares and surplus values. The mechanisms for implementing the protocol are described as a set of extensions to a connectionless network layer protocol, such as IP or CLNP.

The protocol recognizes three sets of entities, the flow sources, the access gateways, and the internal gateways. A flow source, usually a host computer, is the origin of the flow and is assumed to have knowledge about the offered load of the flow. It also maintains a rate controller, which enforces compliance to the bandwidth allocation. All gateways, both access and internal, maintain a set of counters to keep track of the bandwidth being utilized on each outgoing link during an interval (called the update interval). At the end of an update interval, a gateway calculates share values for each outgoing link, and transmits them to access gateways. In addition to the above, an access gateway also calculates the throughput limits for all flows accessing the network through it. The throughput limit is calculated from the share values received from other gateways, and is made available to the respective flow source.

(a) Extensions to Packet Header

For the p–protocol we require four additional fields in the packet header, referred to as *class field*, *link-id field*, *plus flag*, and *minus flag*. The *class field* contains information on the flow class of a packet. The *link-id field* must be large enough to accommodate a unique identification of a network link. In the following we assume that a link identification consists of a pair '`gw:li`' where '`gw`' is the network address of a gateway, and '`li`' identifies an outgoing link of the gateway. The *plus flag* and the *minus flag* have a length of one bit. The content of the header fields is described by:

class field	*link-id field*	*plus flag*	*minus flag*

In the following, we will use '+' to indicate a set *plus flag* in a packet header, '–' to indicate a set *minus flag*, and '$\cdot$' to indicate that a flag is not set.

(b) Update Intervals and Rate Control at Sources

The p–protocol has a system parameter, the so-called *update interval.* We assume the size of the update interval to be of the same order as update periods in routing protocols. At the end of an update interval, each gateway `gw` sends for each outgoing link `gw:li` a control packet with content $\boxed{p \mid \texttt{gw:li} \mid share_p}$ to all access gateways. The control packet indicates the maximum number of bytes that any class-p flow can transmit on link `gw:li` during an update interval. Below, in (e), we will discuss how a gateway calculates the values for $\texttt{Share}_p(\texttt{gw:li})$. After receiving the control packets, the access gateway which is closest to the source of a class-p flow, say flow i, calculates

$$\texttt{Quota[i]} = \min\left(\texttt{Share}_p(\texttt{gw:li}) \mid \texttt{gw:li} \text{ is on the route of class-}p\text{ flow } i\right) \tag{4.1}$$

and communicates the value of `Quota[i]` to the source of flow i, typically a host system. The source of flow i maintains a rate control mechanism which limits the transmission to `Quota[i]`, the maximum amount of data that flow i can transmit during an update interval. We ignore the details of the rate controller and assume only that it does not allow excessive traffic bursts.

(c) States of Flows

Sources of flows have information on the bandwidth demands of their flows, denoted by `Load[i]` for flow i. Also, the sources keep state information on their flows. A flow is either *underloaded*, or *overloaded* at some gateway on its route.

- If $\texttt{Load[i]} \leq \texttt{Quota[i]}$, then flow i is *underloaded.*

  For *underloaded* flows, the source sets the header of each packet to $\boxed{p \mid \textbf{NIL} \mid \cdot \mid \cdot}$.

- If $\texttt{Load[i]} > \texttt{Quota[i]}$, and `Quota[i]`, as calculated in equation (4.1), is such that $\texttt{Quota[i]} = \texttt{Share}_p(\texttt{gw:li})$, then flow i is '*overloaded at* `gw:li`'.

  In this case the source of the flow sets all packet headers to $\boxed{p \mid \texttt{gw:li} \mid \cdot \mid \cdot}$.

Flows can change their state due to changes of their bandwidth demand or changes of `Quota[i]`. The following state transitions can occur:

- *underloaded* $\Longrightarrow$ *overloaded at* `gw:li`.

  In this case, the source sets the header of the flow's next packet to $\begin{array}{|c|c|c|c|}\hline p & \texttt{gw:li} & + & \cdot \\ \hline\end{array}$.

- *overloaded at* `gw:li` $\Longrightarrow$ *underloaded.*

  Then, the source sets the header of the next packet to $\begin{array}{|c|c|c|c|}\hline p & \texttt{gw:li} & \cdot & - \\ \hline\end{array}$.

- *overloaded at* `gw1:li1` $\Longrightarrow$ *overloaded at* `gw2:li2`.

  This state transition is only feasible if both links `gw1:li1` and `gw2:li2` are on the flow's route. The header of the first packet after the state transition is set to $\begin{array}{|c|c|c|c|}\hline p & \texttt{gw:li2} & + & \cdot \\ \hline\end{array}$, and in the immediately following packet, the header is set to $\begin{array}{|c|c|c|c|}\hline p & \texttt{gw:li1} & \cdot & - \\ \hline\end{array}$.

(d) Operations at the Gateways

Next we discuss the functions performed by a gateway, say gateway `gw`. Each outgoing link of the gateway, say, `gw:li`, is assigned a capacity `Cap(gw:li)` which expresses the number of bytes that the link can transmit in an update interval. For a flow class p, the bandwidth guarantee at link `gw:li`, denoted by $\texttt{Guar}_p(\texttt{gw:li})$, gives the transmission guarantee of flow class p during an update interval. The gateway maintains two counters, $\texttt{Rate}_p(\texttt{gw:li})$ and $\texttt{OL}_p(\texttt{gw:li})$, and two variables, $\texttt{Share}_p(\texttt{gw:li})$ and $\texttt{Surplus}_p(\texttt{gw:li})$, for each flow class with $\texttt{Guar}_p(\texttt{gw:li}) > 0$. The counters and variables are mandatory for default class '0'.

The counters at gateway `gw` are updated upon receiving a packet that will be routed on outgoing link `gw:li`. If the fields of the packet header are

$$\begin{array}{|c|c|c|c|}\hline p & \textbf{NIL} & \cdot & \cdot \\ \hline\end{array} \quad \text{or} \quad \begin{array}{c}\begin{array}{|c|c|c|c|}\hline p & \texttt{gw:li1} & \cdot & \cdot \\ \hline\end{array} \\ \text{with} \\ \texttt{gw1:li1} \neq \texttt{gw:li}\end{array}$$

with $\texttt{Guar}_p(\texttt{gw:li}) > 0$, then $\texttt{Rate}_p(\texttt{gw:li})$ is incremented by the packet size. With the same packet header, but $\texttt{Guar}_p(\texttt{gw:li}) = 0$, $\texttt{Rate}_0(\texttt{gw:li})$ is incremented.

If the packet header contains $\boxed{p \mid \texttt{gw:li} \mid + \mid \cdot}$, and $\texttt{Guar}_p(\texttt{gw:li}) > 0$, then $\texttt{OL}_p(\texttt{gw:li})$ is incremented by one. For $\texttt{Guar}_p(\texttt{gw:li}) = 0$, $\texttt{OL}_0(\texttt{gw:li})$ is incremented by one. Likewise, if the packet header reads $\boxed{p \mid \texttt{gw:li} \mid \cdot \mid -}$, then $\texttt{OL}_p(\texttt{gw:li})$ is decremented by one. If $\texttt{Guar}_p(\texttt{gw:li}) = 0$, then $\texttt{OL}_0(\texttt{gw:li})$ will be decremented by one.

(e) Calculation of Share and Surplus Values

After the end of an update interval, each gateway updates its variables $\texttt{Share}_p(\texttt{gw:li})$ and $\texttt{Surplus}_p(\texttt{gw:li})$ by performing the following computations.

$$\texttt{Share}_p(\texttt{gw:li}) = \begin{cases} \texttt{infinity} & \text{if } \texttt{OL}_p(\texttt{gw:li}) = 0 \\ \\ \dfrac{\texttt{Guar}_p(\texttt{gw:li}) + \texttt{Surplus}_p(\texttt{gw:li}) - \texttt{Rate}_p(\texttt{gw:li})}{\texttt{OL}_p(\texttt{gw:li})} & \text{otherwise} \end{cases} \tag{4.2}$$

and

$$\texttt{Surplus}_p(\texttt{gw:li}) = \begin{cases} \texttt{infinity} & \text{if } \texttt{OL}_p(\texttt{gw:li}) = 0 \text{ for all } p \\ \\ \dfrac{\texttt{Cap}(\texttt{gw:li}) - \sum\limits_{\texttt{OL}_q(\texttt{gw:li})>0} \texttt{Guar}_q(\texttt{gw:li}) - \sum\limits_{\texttt{OL}_q(\texttt{gw:li})=0} \texttt{Rate}_q(\texttt{gw:li})}{|\{q \mid \texttt{OL}_p(\texttt{gw:li}) > 0\}|} & \text{otherwise} \end{cases} \tag{4.3}$$

The results for the new values for $\texttt{Share}_p(\texttt{gw:li})$ and $\texttt{Surplus}_p(\texttt{gw:li})$ are sent to all access gateways in the abovementioned control packets with content

$$\boxed{p \mid \texttt{gw:li} \mid \texttt{Share}_p\ (\texttt{gw:li}) \mid \cdot \mid -} \text{ if } \texttt{OL}_p(\texttt{gw:li}) = 0,$$

and

$$\boxed{p \mid \texttt{gw:li} \mid \texttt{Share}_p\ (\texttt{gw:li}) \mid + \mid \cdot} \text{ otherwise.}$$

Finally, the gateway resets its counters $\texttt{Rate}_p(\texttt{gw:li})$ to zero.

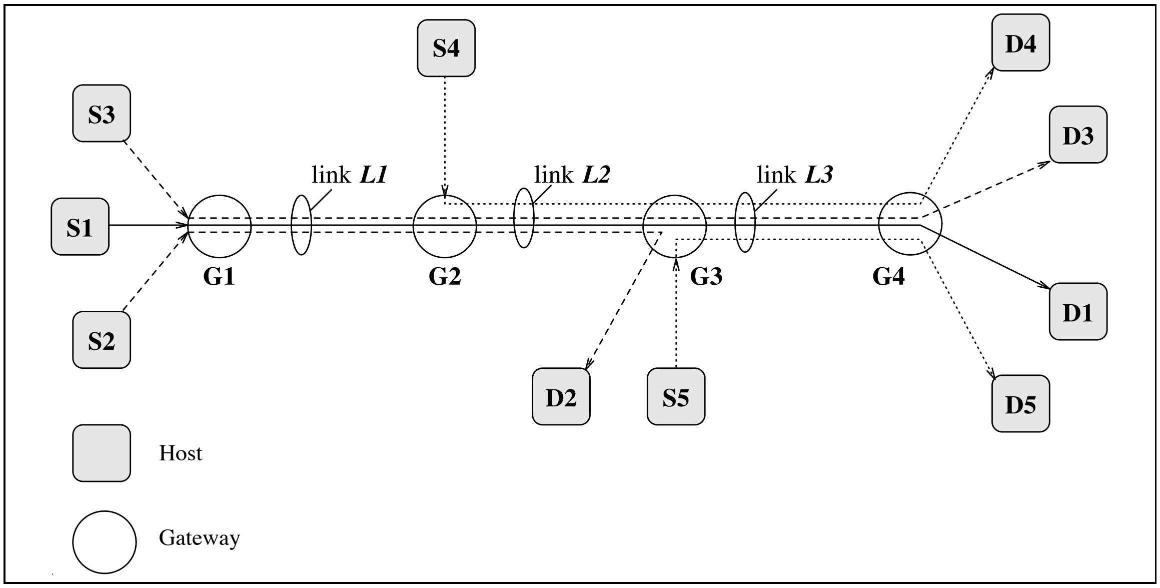


Figure 4.1: Simulated Network.

Note that equations (4.2) and (4.3) are based on our Theorem 3.1. In equations (4.2) and (4.3), `infinity` is chosen such that `infinity` $\gg$ `Cap` (`gw:li`). Both equations can be computed for all flow classes without information on the share or surplus values at other gateways. By setting `gw:li` $\equiv l$, and by neglecting that Theorem 3.1 is expressed in terms of data rates, we obtain the following relation between equations (4.2) – (4.3) and Theorem 3.1:

$\texttt{Share}_p(\texttt{gw:li})$	$\alpha^*_p(l)$	$\texttt{Surplus}_p(\texttt{gw:li})$	ϕ^*_l
$\texttt{Cap}(\texttt{gw:li})$	C_l	$\texttt{Guar}_p(\texttt{gw:li})$	G_{lp}
$\texttt{OL}_p(\texttt{gw:li})$	$\lvert\mathbf{O}_{lp}\rvert$	$\texttt{Rate}_p(\texttt{gw:li})$	Θ_{lp}

4.1.2 Simulation Experiment

To provide insight into the dynamics of our bandwidth regulation p–protocol we present a simulation experiment that shows the transient behavior during changes of the network load. The simulation was implemented using the REAL (version 4.0) network simulator [22]. We modified the source code of REAL to include our protocol.

For the simulations, we make the following assumptions. Packet sizes are constant for all flows and set to 125 Bytes. Propagation delays are small and set to $10\mu s$. Each source

Flow (Source Host)	Destination Host	Route	Class	Offered Load	Start Time (*in s*)
S1	*D1*	(*L1*, *L2*, *L3*)	*0*	10 Mb/s	$t = 0$
S2	*D2*	(*L1*, *L2*)	*II*	40 Mb/s	$t = 20$
S3	*D3*	(*L1*, *L2*, *L3*)	*II*	70 Mb/s	$t = 40$
S4	*D4*	(*L2*, *L3*)	*0*	70 Mb/s	$t = 90$
S5	*D5*	(*L3*)	*I*	60 Mb/s	$t = 140$

Table 4.1: Flow Parameters.

of a flow, i.e., a host, has knowledge of the offered load, and generates packets after fixed time intervals. Packet losses due to transmission errors or buffer overflows at gateways do not occur. The latter is achieved by selecting the buffer sizes at gateways sufficiently large. Also, end-to-end window flow control mechanisms are not used in the simulation. Finally, the scheduling discipline at all gateways is assumed to be FIFO.

As shown in Figure 4.1, the topology of the simulated network consists of ten hosts, *S1* – *S5* and *D1* – *D5*, and four gateways, *G1* – *G4*. The network links, denoted by *L1*, *L2* and *L3*, each have a capacity of 10 Mb/s. We simulate the behavior of five flows from three different flow classes: *0*, *I*, and *II*. The bandwidth guarantees of the flow classes are identical at all links, and denoted by G_0, G_I, and G_{II}. The guarantees are set to:

$$\begin{aligned}
G_0 &= 15 \quad \text{Mb/s for class } \mathit{0}, \\
G_I &= 30 \quad \text{Mb/s for class } \mathit{I}, \\
G_{II} &= 55 \quad \text{Mb/s for class } \mathit{II}.
\end{aligned}$$

The parameters of the five flows in Figure 4.1, that is, source host, destination host, route, flow class membership, offered load, and time of first packet transmission, are summarized in Table 4.1. Since each host is the source or destination of at most one flow, we will use the source host to identify a flow. The length of the update interval between calculations of share and quota values is set to 2 seconds.

In the simulations, we measure the data that each flow can transmit on a link during an update interval. The simulation results are summarized in Figure 4.2. The figure depicts three graphs which show, separate for each link, the bandwidth (in Mb/s) utilized by each flow. From top to bottom, the graphs show the transmissions by gateway *G1* on link *L1*, by gateway *G2* on link *L2*, and by gateway *G3* on link *L3*. Each data point in the graph corresponds to the amount of data that is transmitted during an update interval of 2 seconds.

Next we discuss the outcome of the simulation.

- At $t = 0$, flow *S1* from class *0* starts transmission on all three links. Since no other flow is transmitting, flow *S1* is underloaded and can send its entire load of 10 Mb/s.

- At $t = 20$, class-*II* flow *S2* with a load of 40 Mb/s becomes active on links *L1* and *L2*. Since both flows *S1* and *S2* are underloaded with respect to their class guarantees, they are allowed to transmit at their offered loads.

- At $t = 40$, another class-*II* flow, *S3*, starts to transmit over links *L1*, *L2*, and *L3*, with an offered load of 70 Mb/s. With *S3*, class *II* requires more bandwidth on link *L1* than it is guaranteed. As it is the only such class, inter-class regulation permits class *II* to borrow from the bandwidth guarantees made to other classes. Thus, class *II* obtains 90 Mb/s bandwidth for transmissions on link *L1*. Within class *II*, there is one underloaded flow (*S2*) and one overloaded flow (*S3*). Intra-class regulation now controls the bandwidth allocation to these flows. The theoretical share and surplus values for link *L1*, as well as the flow throughputs after $t = 40$ are calculated as follows[1]:

	$\alpha_0(L1)$	$\alpha_I(L1)$	$\alpha_{II}(L1)$	ϕ_{L1}
Link *L1*	–	–	50	35

γ_{S1}	γ_{S2}	γ_{S3}
10	40	50

In Figure 4.2 it can be seen that the protocol quickly settles at the predicted values.

- At $t = 90$, flow *S4* from class *0* starts transmission on links *L2* and *L3* with an offered load of 70 Mb/s. Then, both classes *0* and *II* require their respective bandwidth guarantees

[1]The data in the tables is given in Mb/s. For clarity, we substituted the symbol '∞' by '–'.

on link *L2*. Since there is no class-I traffic on link *L2*, inter-class regulation permits the bandwidth guarantee to class *I* to be split between classes *0* and *II*. After $t = 90$, the expected share and surplus values for link *L2*, and the throughputs of flows with traffic on link *L2* are as follows:

	$\alpha_0(L2)$	$\alpha_I(L2)$	$\alpha_{II}(L2)$	ϕ_{L2}
Link *L2*	20	–	35	15

γ_{S1}	γ_{S2}	γ_{S3}	γ_{S4}
1	35	35	20

Within class *0*, flow *S1* is underloaded and *S4* is overloaded at link *L2*. Note in Figure 4.2 that the throughputs of *S2* and *S3* drop to 35 Mb/s.

- At $t = 140$, flow *S5* from class *I* becomes active on link *L3* with a load of 60 Mb/s. Since flow *S5* requires its entire bandwidth guarantee of 30 Mb/s at link *L3*, inter-class regulation forces all other classes to reduce transmissions to their respective guarantees. This results in an interesting shift of bottleneck links. The reduced bandwidth at link *L3* decreases the throughput available to *S4* (from class *0*), and causes a shift of flow *S4*'s bottleneck from link *L2* to *L3*. This in turn, makes bandwidth available for class-*II* flows on link *L2*, yielding a throughput increase for flows *S2* and *S3*. However, since flow *S2* is still restricted at its bottleneck link *L2*, it cannot fully utilize its bandwidth guarantee at link *L3*. Hence, flow *S4* from class *0* and flow *S5* from class *I* can borrow the unused class-*II* guarantee on link *L3*. Note from Figure 4.2 that the protocol requires a few iterations before settling at the correct bandwidth allocation. Eventually, the following theoretically expected values are obtained :

	$\alpha_0(L2)$	$\alpha_I(L2)$	$\alpha_{II}(L2)$	ϕ_{L2}
Link *L2*	–	–	38.3	21.7
	$\alpha_0(L3)$	$\alpha_I(L3)$	$\alpha_{II}(L3)$	ϕ_{L3}
Link *L3*	13.3	38.3	–	8.3

γ_{S1}	γ_{S2}	γ_{S3}	γ_{S4}	γ_{S5}
10	38.3	38.3	13.3	38.3

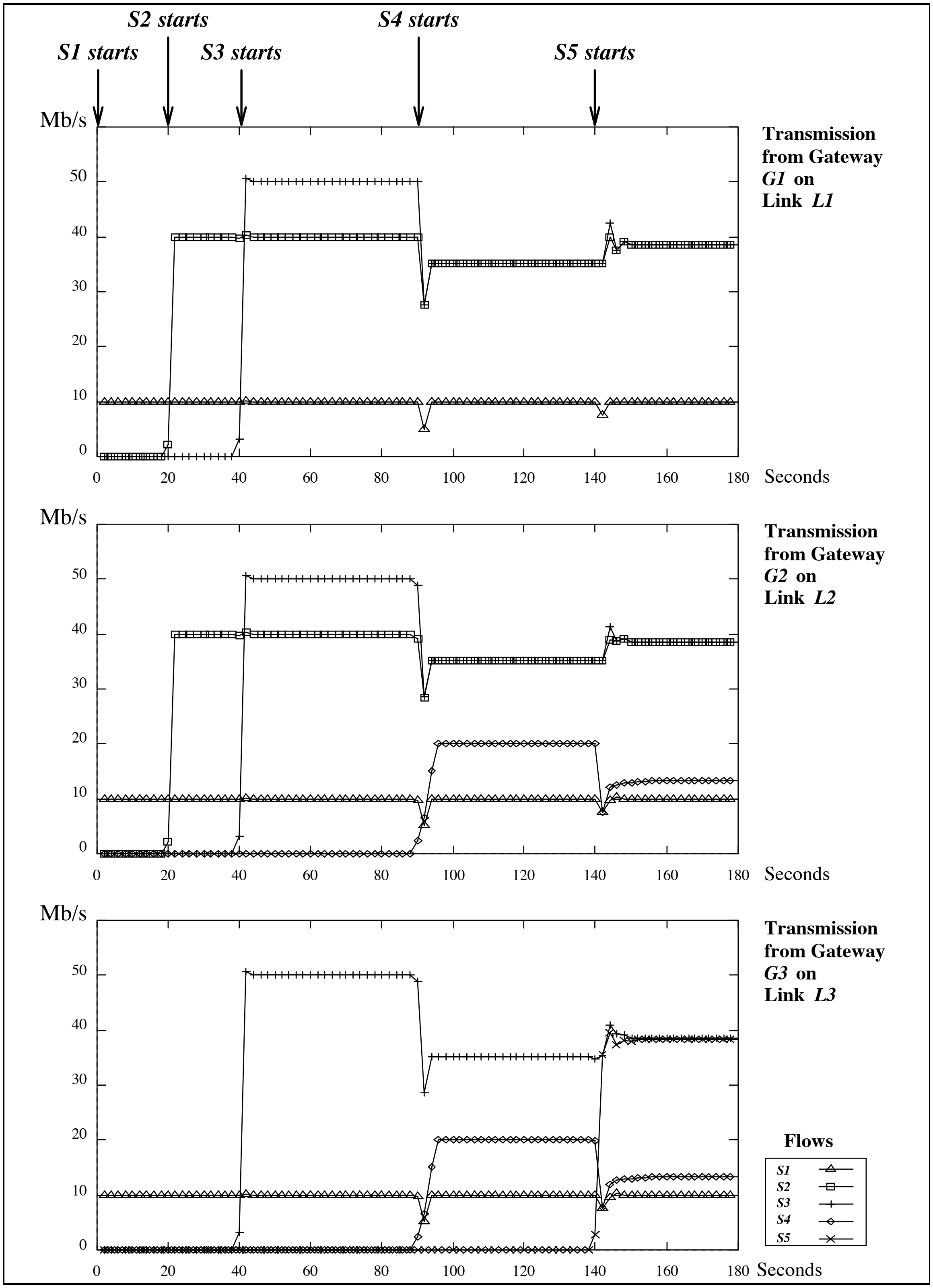


Figure 4.2: Simulation Results.

4.2 A Multi-level Bandwidth Regulation Protocol for ATM Networks

The c–protocol is designed to be implemented in an ATM network which supports both connection-oriented and connectionless traffic. The protocol assumes that the *direct* model discussed in Chapter 1, is used by the ATM network to support connectionless traffic. As shown in Figure 3.3, a number of CLSs are linked together by virtual paths to form a virtual overlay network. Host systems access the network via *access* CLSs, which are connected to both host systems and other CLSs. The *internal* CLSs are connected only to other CLSs. The c–protocol is limited to regulating the traffic carried on the virtual network of CLSs. The c–protocol is discussed in greater detail in §4.2.1, and the results of our simulation experiments are presented in §4.2.2.

4.2.1 The c–protocol

The c–protocol is able to implement the three-level bandwidth regulation discussed in Chapter 3. It relies upon the mathematical relations developed in Theorem 3.1 to achieve short-term bandwidth regulation which is both inter-class and intra-class fair. Medium-term regulation is achieved using Theorem 3.2 to satisfy class level fairness conditions. Finally, Definition 3.5 is used to achieve long-term regulation.

Unlike the p–protocol, the c–protocol does not regulate all traffic flows in the network. It only regulates the connectionless traffic that is carried on the virtual network of CLSs. Connection-oriented traffic carried by the ATM network separately is not controlled by the c–protocol. The short-term regulation achieved by the c–protocol is similar to the bandwidth regulation implemented by the p–protocol, and also achieves inter-class and intra-class fairness. Medium-term regulation is achieved by dynamically varying the guarantees to the flow classes, which were assumed to be constant in the p–protocol. The c–protocol achieves long-term regulation by adjusting the bandwidth allocated to the VPs forming the virtual network. Long-term regulation is beyond the scope of the p–protocol, as it deals with actual network links with fixed capacity.

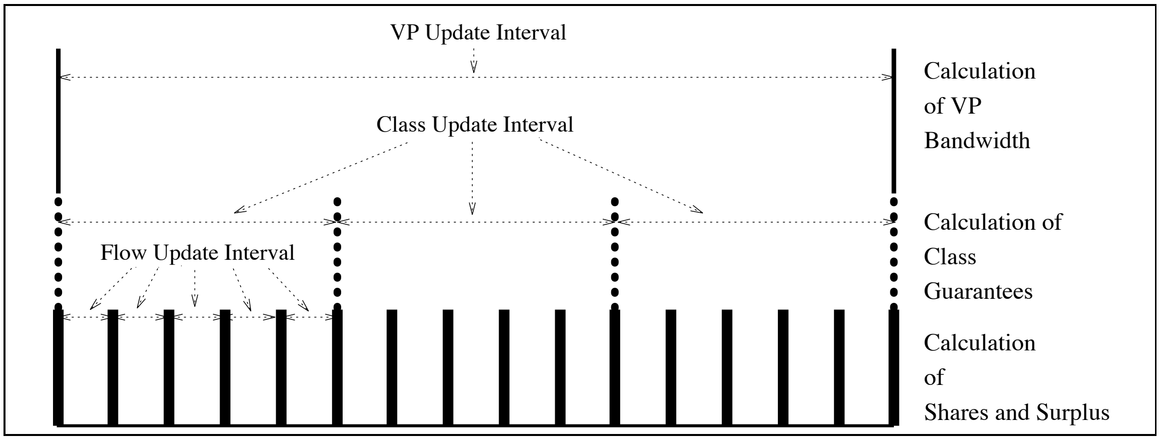


Figure 4.3: Update Intervals.

(a) Extensions to CL-PDU Header

The p–protocol requires four additional fields in the CL-PDU header. The contents of the header fields are described by :

class field	*VP-id field*	*plus flag*	*minus flag*

The header fields are identical to that of the c–protocol except for the *VP-id field*, which denotes a unique identification of a network VP. Similar to §4.1.1, we assume that a VP identification consists of a pair '`cls:li`' where '`cls`' is the network address of a CLS, and '`li`' identifies an outgoing VP of the CLS. The CL-PDU size refers to the total number of bytes needed to transmit the CL-PDU using 53 byte ATM cells.

(b) Update Intervals

The c–protocol has three system parameters, the *flow update interval* T_f, the *class update interval* T_c, and the *VP update interval* T_v. We assume the size of T_v to be of the same order as update periods in routing protocols. T_v is an integral multiple of T_c which is in turn an integral multiple of T_f. The relations between the three update intervals and the calculations performed at the end of each are illustrated graphically in Figure 4.3.

At the end of every *flow update interval*, the bandwidth allocated to each flow is re-calculated by performing the calculations detailed in (f). At the end of every *class update* *interval*, the bandwidth guarantee of each flow class is recalculated by performing the calculations described in (g) in addition to the calculations in (f). Finally, at the end of every T_v the bandwidth allocation to the VPs carrying connectionless traffic is recalculated using the calculations in (h) after performing the calculations in (f) and (g).

(c) Rate Control at Sources

At the end of a *flow update interval*, the CLSs perform the same operations undertaken by the gateways in §4.1.1. The CLS broadcasts a control CL-PDU with contents $\boxed{p \mid \texttt{cls:li} \mid share_p}$. The control CL-PDU indicates the maximum number of bytes that any class-p flow can transmit on VP `cls:li` during an update interval. The exact procedure for calculating the values of $\texttt{Share}_p(\texttt{cls:li})$ is detailed in (f). The value of `Quota[i]` for a flow i is calculated according to the following equation:

$$\texttt{Quota[i]} = \min\left(\texttt{Share}_p(\texttt{cls:li}) \mid \texttt{cls:li} \text{ is on the route of class-}p \text{ flow } i\right) \tag{4.4}$$

The value of `Quota[i]` is then communicated to the source of flow i, for use by the rate controller.

(d) States of Flows

Like the p–protocol, the c–protocol also defines two states of flows. A flow is either *underloaded*, or *overloaded* at some CLS on its route. The definitions of the two states, as well as the permitted state transitions are identical to those described in §4.1.1.

(e) Operations at the Connectionless Servers

The Connectionless Servers in the c–protocol perform the same functions as the gateways of the p–protocol. In addition to the counters and variables maintained at the gateways, the CLSs maintain a separate counter $\texttt{Total}_p(\texttt{cls:li})$, for each flow class with $\texttt{Guar}_p(\texttt{cls:li}) > 0$. The counters and variables are mandatory for default class '0'. In addition, each CLS maintains a counter `Util(cls:li)` for each outgoing VP (`cls:li`).

The counters $\texttt{Rate}_p(\texttt{cls:li})$ and $\texttt{OL}_p(\texttt{cls:li})$ are updated upon receiving a CL-PDU which will be routed on outgoing VP `cls:li`. The rules for updating the counters are the same as in the p–protocol (§4.1, part **(d)**). In addition, for all CL-PDUs received, $\texttt{Total}_p(\texttt{cls:li})$ is incremented by the CL-PDU size.

(f) Short-term Regulation

The c–protocol achieves short-term regulation by recalculating share and surplus values after every *flow update interval.* The calculations performed are similar to the calculations performed by the p–protocol at the end of each update interval (§4.1.1, part (e)). After the end of every T_f time units, each CLS updates its variables $\texttt{Share}_p(\texttt{cls:li})$ and $\texttt{Surplus}_p(\texttt{cls:li})$ by performing the following computations:

$$\texttt{Surplus}_p(\texttt{cls:li}) =$$
$$\begin{cases} \texttt{infinity} & \text{if } \texttt{OL}_p(\texttt{cls:li}) = 0 \text{ for all } p \\ \dfrac{\texttt{Cap}(\texttt{cls:li}) - \sum\limits_{\texttt{OL}_q(\texttt{cls:li})>0} \texttt{Guar}_q(\texttt{cls:li}) - \sum\limits_{\texttt{OL}_q(\texttt{cls:li})=0} \texttt{Rate}_q(\texttt{cls:li})}{|\{q \mid \texttt{OL}_p(\texttt{gw:li}) > 0\}|} & \text{otherwise} \end{cases} \tag{4.5}$$

and

$$\texttt{Share}_p(\texttt{cls:li}) =$$
$$\begin{cases} \texttt{infinity} & \text{if } \texttt{OL}_p(\texttt{cls:li}) = 0 \\ \dfrac{\texttt{Guar}_p(\texttt{cls:li}) - \texttt{Surplus}_p(\texttt{cls:li}) - \texttt{Rate}_p(\texttt{cls:li})}{\texttt{OL}_p(\texttt{cls:li})} & \text{otherwise} \end{cases} \tag{4.6}$$

The above equations are based on our Theorem 3.1. Note that equations (4.5) and (4.6) can be computed for all flow classes without information on the share or surplus values at other CLSs. The results for the new values for $\texttt{Share}_p(\texttt{cls:li})$ are encapsulated in the following control CL-PDU: $\boxed{p \mid \texttt{cls:li} \mid \texttt{Share}_p}$. The control CL-PDUs are sent to all access CLSs as described in (c). Finally, the CLS resets its counters $\texttt{Rate}_p(\texttt{cls:li})$ to zero.

(g) Medium-term Regulation

Medium-term regulation involves periodic recalculation of guarantees provided to each flow class. At the end of every T_c time units, the following calculations are performed at each CLS after performing the calculations detailed in (f). In the following equations we shall use the notation:

$$\mathtt{T}_p(\mathtt{cls{:}li}) \equiv \mathtt{Total}_p(\mathtt{cls{:}li}) \cdot \frac{\text{length of flow update interval}}{\text{length of class update interval}} \tag{4.7}$$

We use a variable $\mathtt{Min\text{-}Guar}_p(\mathtt{cls{:}li})$ as a measure of the minimum guarantee that class p is entitled to receive at VP "$\mathtt{cls{:}li}$". The guarantee $\mathtt{Guar}_p(\mathtt{cls{:}li})$ will be less than $\mathtt{Min\text{-}Guar}_p(\mathtt{cls{:}li})$ if and only if $\mathtt{T}_p(\mathtt{cls{:}li}) < \mathtt{Min\text{-}Guar}_p(\mathtt{cls{:}li})$. First, the fraction of bandwidth that is available for redistribution is calculated:

$$\mathtt{Spare}(\mathtt{cls{:}li}) = \frac{\mathtt{Cap}(\mathtt{cls{:}li})(1 - \sum\limits_{\mathtt{T}_p(\mathtt{cls:li}) \geq \mathtt{Guar}_p(\mathtt{cls:li})} \mathtt{Min\text{-}Guar}_p(\mathtt{cls{:}li})) - \sum\limits_{\mathtt{T}_p(\mathtt{cls:li}) < \mathtt{Guar}_p(\mathtt{cls:li})} \mathtt{T}_p(\mathtt{cls{:}li})}{\mathtt{Cap}(\mathtt{cls{:}li})} \tag{4.8}$$

We then divide the spare bandwidth among all classes that have utilized their full guarantees. $\mathtt{Guar\text{-}share}(\mathtt{cls{:}li})$ is the variable denoting the fraction of the surplus bandwidth that each such class is entitled to receive.

$$\mathtt{Guar\text{-}share}(\mathtt{cls{:}li}) = \begin{cases} 0 & \text{if } \mathtt{T}_p(\mathtt{cls{:}li}) < \mathtt{Guar}_p(\mathtt{cls{:}li}) \text{ for all } p \\ \dfrac{\mathtt{Spare}(\mathtt{cls{:}li})}{\sum\limits_{\mathtt{T}_p(\mathtt{cls:li}) \geq \mathtt{Guar}_p(\mathtt{cls:li})} \mathtt{Min\text{-}Guar}_p(\mathtt{cls{:}li})} & \text{otherwise} \end{cases} \tag{4.9}$$

Then the class guarantee $\mathtt{Guar}_p(\mathtt{cls{:}li})$ is calculated as follows:

$$\mathtt{Guar}_p(\mathtt{cls{:}li}) = \begin{cases} \mathtt{T}_p(\mathtt{cls{:}li}) & \text{if } \mathtt{T}_p(\mathtt{cls{:}li}) < \mathtt{Guar}_p(\mathtt{cls{:}li}) \\ \mathtt{Min\text{-}Guar}_p(\mathtt{cls{:}li}) + \mathtt{Guar\text{-}share} & \text{otherwise} \end{cases} \tag{4.10}$$

Finally, the total utilization of the VP in the last class update interval is calculated.

$$\mathtt{Utilized} = \sum_p \mathtt{T}_p(\mathtt{cls{:}li})$$

The counters $\texttt{Total}_p(\texttt{cls:li})$ are reset to 0 and the counter $\texttt{Util}(\texttt{cls:li})$ is incremented by the value of `Utilized`.

(h) Long-term Regulation

The c–protocol achieves long-term regulation by recalculating the bandwidth allocated to the VP carrying connectionless traffic at the end of every T_v time units. In the following, $\texttt{Min-Cap}(\texttt{cls:li})$ denotes the minimum bandwidth guarantee provided by an ATM link to a VP `cls:li` carrying connectionless traffic. First, the average utilization of each virtual path over a VP update interval is calculated as follows:

$$\texttt{LU}(\texttt{cls:li}) = \texttt{Util}(\texttt{cls:li}) \cdot \frac{\text{length of VP update interval}}{\text{length of class update interval}}$$

Then, the bandwidth at the ATM link is allocated among connection-oriented and connectionless traffic according to the following algorithm:

1. (a) If $\texttt{LU}(\texttt{cls:li})$ is less than $\texttt{Cap}(\texttt{cls:li})$ then the capacity allocated to `cls:li`, $\texttt{Cap}(\texttt{cls:li})$ is reduced to $\texttt{LU}(\texttt{cls:li})$.

   or

   (b) Otherwise $\texttt{Cap}(\texttt{cls:li})$ is set equal to $\texttt{Min-Cap}(\texttt{cls:li})$.

2. Connection-oriented traffic can allocate bandwidth from the capacity available at the ATM link after performing the allocations in (1).

3. If the ATM link has any bandwidth remaining after the allocations in step (3), the bandwidth is allocated to the VP carrying connectionless traffic.

4.2.2 Simulation Experiment

We now present a set of simulation experiments to demonstrate the capabilities of the c–protocol. The experiments have been designed to highlight the effectiveness of the protocol in achieving a fair and maximal bandwidth allocation in a changing network environment. The simulation protocol has been implemented by modifying the source code of the REAL (version 4.0) network simulator [22].

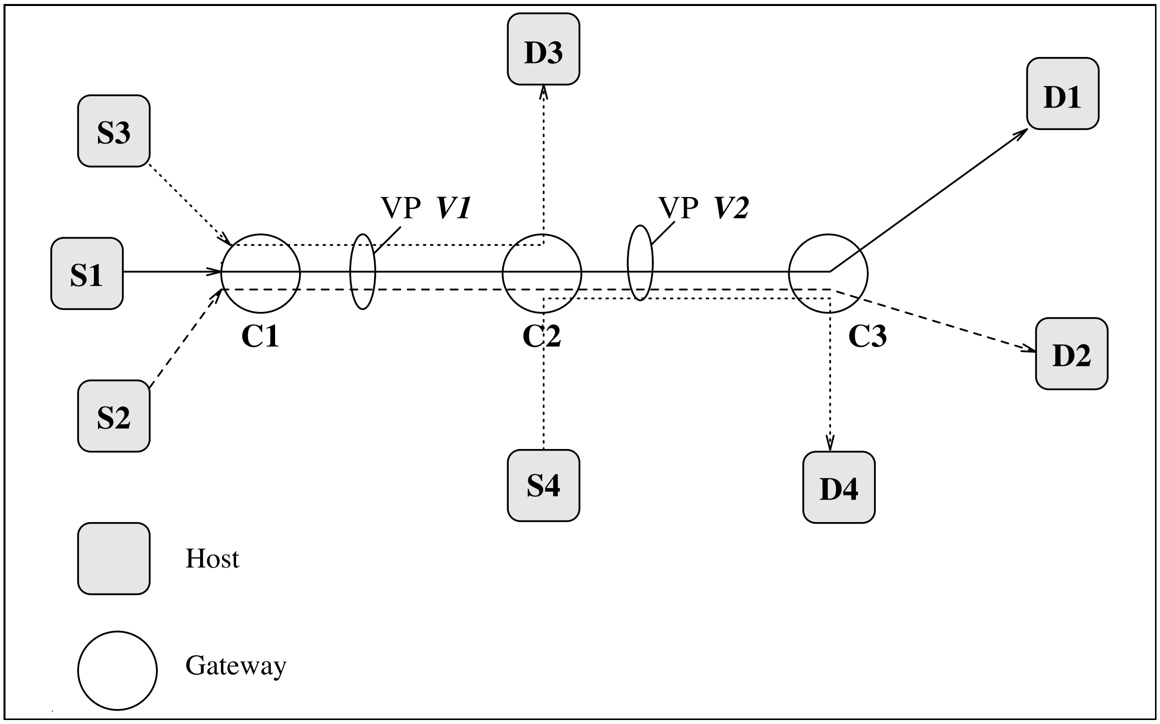


Figure 4.4: Simulated Network.

We make the following assumptions in the simulations. The CL-PDU sizes are constant for all flows and set to 1056 bytes. This allows each CL-PDU to be split into twenty-four 44-byte chunks. These are carried across the network by 24 ATM cells using the AAL 3/4 adaptation layer. We chose AAL 3/4 over AAL 5 inspite of the latter's higher payload, because AAL 3/4 allows multiplexing of flows over the same VP. Propagation delays are small and set to 1 ms. Each source of a flow, i.e., a host, is assumed to have knowledge of the offered load, and generates CL-PDUs after fixed time intervals. Cell losses due to transmission errors or buffer overflows at CLSs do not occur. The latter is achieved by selecting the buffer sizes at CLSs sufficiently large. Also, end-to-end window flow control mechanisms are not used in the simulation. Finally, the scheduling discipline at all CLSs is assumed to be FIFO.

As shown in Figure 4.4, the topology of the simulated network consists of eight hosts, *S1 – S4* and *D1 – D4*, and three CLSs, *C1 – C3*. The network VPs, denoted by *V1* and *V2*, each have a capacity of 155 Mb/s. The VPs are part of ATM lines of capacity 622 Mb/s. The minimum bandwidth guaranteed to connectionless flows at each ATM link, $M_{l_{CL}}$, is 93

Mb/s. We assume that the total connection-oriented traffic load, $\Lambda_{l_{CO}}$, is 467 Mb/s initially. In accordance with long-term regulation policies, the VPs carrying connectionless traffic are allocated the bandwidth unutilized by connection-oriented traffic. Therefore, *V1* and *V2* have an initial capacity $C_{l_{CL}}$ of 155 Mb/s. We simulate the behavior of four flows from three different flow classes: *0*, *I*, and *II*. The bandwidth guarantees of the flow classes are initially identical at all VPs, and denoted by G_0, G_I, and G_{II}. The minimum guarantees of the flow classes are equal at all VPs and denoted by M_0, M_I, and M_{II}. The values of the class guarantees, the actual bandwidth guarantees initially, and the minimum guarantees are given in Table 4.2. The actual bandwidth guarantees have been calculated using the initial value of $C_{l_{CL}}$, 155 Mb/s.

All flows start out with a low offered load of 10 Mb/s initially and increase their loads at different time instants. The parameters of the four flows in Figure 4.4, that is, source host, destination host, route, flow class membership, final offered load, and time of first CL-PDU transmission at the increased rate, are summarized in Table 4.3. Since each host is the source or destination of at most one flow, we will use the source host to identify a flow. The length of the *flow update interval* between calculations of share and quota values is $T_f = 1$ second. The lengths of the *class* and *VP update intervals* are $T_c = 10$ seconds and $T_v = 50$ seconds, respectively.

In order to demonstrate the effect of the three-level regulation mechanism clearly, we have performed three simulation experiments. The first experiment shows the effect of short-term regulation alone. In this experiment the class guarantees as well as the VP capacity is fixed. The second experiment shows the effects of short-term and medium-term regulation combined. We allow the class guarantees to be variable, that is, they are regulated by our protocol. Only the capacity of the VPs remain fixed. Finally, the third experiment demonstrates long-term bandwidth regulation with both medium-term and short-term regulation. The VP capacity as well as the class guarantees are now regulated by our protocol. In the simulation experiments, we have made a few modifications to the protocol. In the medium-term regulation of class guarantees, we have tried to avoid sharp increases or decreases in the guarantees to the flow classes. In the simulation, the guarantee

	class-*0*	class-*I*	class-*II*
Initial Guarantees	$G_0 = 0.25$	$G_I = 0.30$	$G_{II} = 0.45$
Bandwidth Guaranteed Initially	38.75 Mb/s	46.5 Mb/s	69.75 Mb/s
Minimum Guarantees	$M_0 = 0.13$	$M_I = 0.15$	$M_{II} = 0.23$

Table 4.2: Class Guarantees.

Flow (Source Host)	Destination Host	Route	class	Offered Load	Start Time (*in s*)
S1	*D1*	(*V1*, *V2*)	*0*	70 Mb/s	$t = 10$
S2	*D2*	(*V1*, *V2*)	*II*	65 Mb/s	$t = 20$
S3	*D3*	(*V1*)	*II*	70 Mb/s	$t = 30$
S4	*D4*	(*V2*)	*I*	80 Mb/s	$t = 40$

Table 4.3: Flow Parameters.

of any flow class cannot drop below its minimum guarantee. Also, the end of any class update interval, the guarantee cannot be increased by more than 20% of its previous value.

Short-term Regulation

In this experiment, both the medium-term and long-term regulations have been turned off in order to demonstrate the effects of short-term regulation. The simulation results are summarized in Figure 4.5 which depicts two graphs which show the bandwidth (in Mb/s) utilized by each flow at the two VPs, *V1* and *V2*. Each data point in the graph corresponds to the amount of data that is transmitted during a flow update interval of $T_f = 1$ second. The experimental results have been verified with theoretically expected values from §4.2.1 and found to be accurate. Next we discuss the outcome of the simulation.

- At $t = 0$, all flows *S1*–*S4* start transmission with an initial offered load of 10 Mb/s each. Each flow is underloaded and can send its entire load of 10 Mb/s.

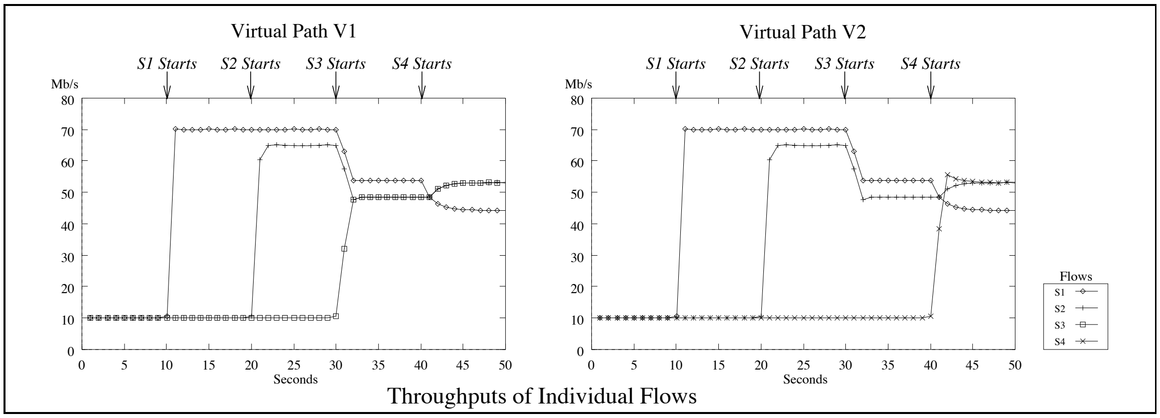


Figure 4.5: Short-term Regulation.

- At $t = 10$, class-*0* flow *S1* increases its load to 70 Mb/s on VPs *V1* and *V2*, which exceeds the guarantee of class *0*. As none of the other classes have utilized their full guarantees, inter-class fairness allows class *0* to borrow extra bandwidth from the other classes. This allows *S1* to transmit at its offered load.

- At $t = 20$, a class-*II* flow, *S2*, transmitting over VPs *V1* and *V2* increases its offered load to 65 Mb/s. This causes class *II* to exceed its guarantee. However, inter-class fairness permits class *II* to borrow sufficient bandwidth from class *I* to allow *S2* to transmit at its offered rate.

- At $t = 30$, flow *S3* from class *II* increases its transmission on VP *V1* with an offered load of 70 Mb/s. Then, classes *0* and *II* require their respective bandwidth guarantees on VP *V1*. Since there is no class-*I* traffic on VP *V1*, inter-class regulation permits the bandwidth guarantee to class *I* to be split between classes *0* and *II* in the ratio of their respective guarantees. As *S1* is the only flow in class *0*, it gets the entire bandwidth available to its class. However, intra-class fairness causes the bandwidth available to class *II* at VP *V1* to be split evenly between *S2* and *S3*. *V1* thus becomes the bottleneck for flows *S1*–*S3*.

- At $t = 40$, flow *S4* from class *I* becomes active on VP *V2* with an increased load of 80 Mb/s. Since flow *S4* requires its entire bandwidth guarantee of 46.5 Mb/s at VP *V2*, inter-class regulation forces all other classes to reduce transmissions to their respective

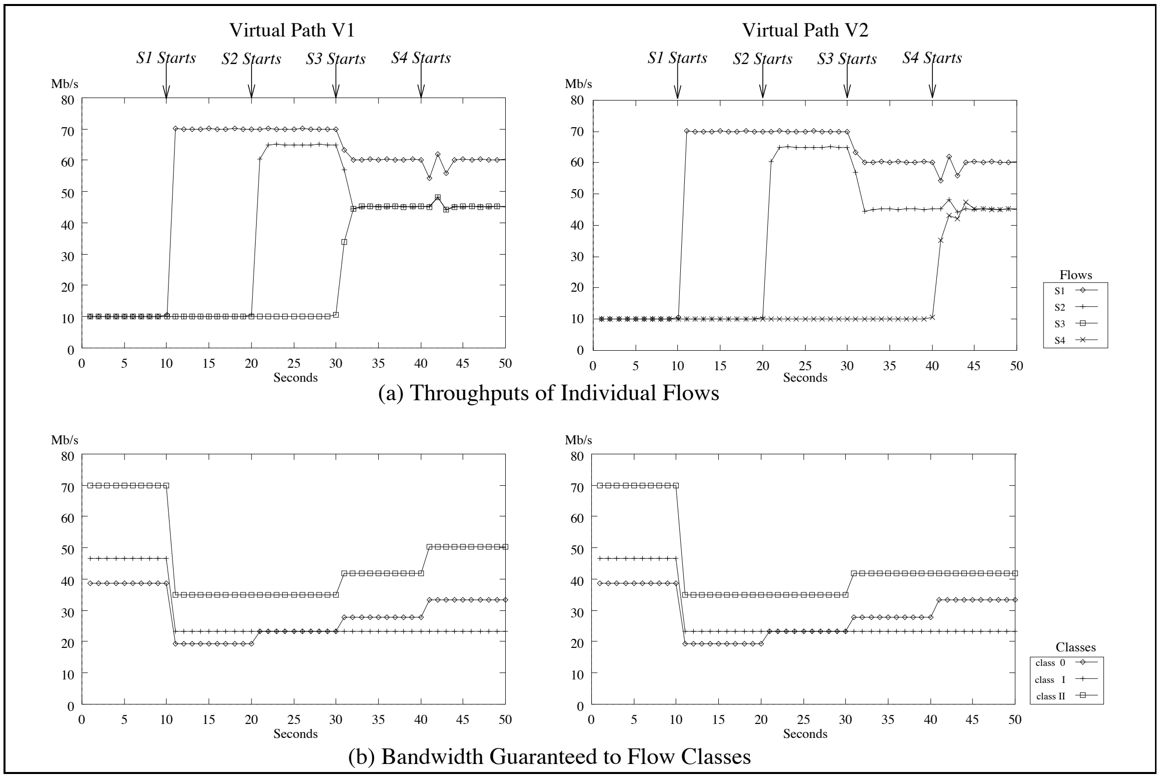


Figure 4.6: Medium-term Regulation.

guarantees. The reduced bandwidth at VP *V2* decreases the throughput available to *S1* (from class *0*), and causes a shift of flow *S1*'s bottleneck from *V1* to *V2*. This in turn, makes bandwidth available for class-*II* flows on VP *V1*, yielding a throughput increase for flows *S2* and *S3*. This reduces the bandwidth available to flows *S1* and *S4*, which have their bottlenecks on *V2*, causing their throughputs to drop. The drop in throughput of *S1* causes another, smaller increase in the throughputs of *S2* and *S3*. Note from Figure 4.5 that the protocol requires a few iterations before settling at the correct bandwidth allocation.

Medium-term Regulation

In this experiment, we have the same set of flows as in the flow level experiment, but now we let the class guarantees be regulated by the medium-term regulation protocol. The bandwidth allocated to the VPs *V1* and *V2* remains fixed at 155 Mb/s each. The initial bandwidth guarantees to each class as well as the minimum guarantees are as given in

Table 4.2. The length of the class level update interval is $T_c = 10$ seconds. The simulation results are summarized in Figure 4.6. The top pair of graphs shows, similar to Figure 4.5, the bandwidth utilized by each flow. The bottom pair of graphs shows the guarantees obtained by the flow classes *0–II* at the two VPs. Both sets of experimental results have been verified to satisfy the conditions for intra-class, inter-class, and class level fairness from Subsections 4.1.1 and 4.2.1.

- At t = 0, flows *S1–S4* start transmission with offered load of 10Mb/s each. All flows are underloaded and well within the class guarantee of each class.

- At t = 10, class guarantees are updated. As in the preceding interval no class utilized even their minimum guarantees, medium-term regulation reduces their guarantees to their respective minimum guarantees. Therefore, the new guarantees for class *0*, class *I* and class *II* are set to 12.5% (19.37Mb/s), 15% (23.25Mb/s), and 22.5% (34.87Mb/s), respectively on both *V1* and *V2*. The class-*0* flow *S1* increases its offered load to 70 Mb/s on VPs *V1* and *V2*, which exceeds the guarantee of class *0*. Inter-class fairness permits class *0* to borrow bandwidth from the other two classes. This allows *S1* to transmit at its offered load.

- At t = 20, flow *S2* of class *II* increases its offered load to 65 Mb/s on VPs *V1* and *V2*. Inter-class fairness allows class *II* to borrow unutilized bandwidth at both VPs. This allows *S2* to transmit at its offered load. As the utilization of class *0* exceeds its guarantee at both VPs, class level fairness allows the class guarantee of class *0* to be increased to 15% (23.25Mb/s) at both VPs *V1* and *V2*.

- At t = 30, flow *S3* of class *II* increases its offered load to 70 Mb/s on *V1*. This causes flows *S1–S3* to become overloaded at *V1*. As there is no class-*I* flow in *V1*, inter-class fairness causes the available bandwidth to be divided between class *0* and class *II*. Both classes *0* and *II* have utilizations exceeding their guarantees. Therefore, class level fairness allows their guarantees to be increased to 18% (27.9 Mb/s) and 27% (41.85 Mb/s), respectively, on both VPs *V1* and *V2*. Note that $S1$ has received a higher throughput compared to the short-term regulation experiment. This is caused by the fact that medium-term regulation has increased the bandwidth guarantee of class *0* relative to that of class *II*. This gives class *0*, and therefore flow $S1$, a bigger portion of the surplus bandwidth.

- At t = 40, flow *S4* of class *I* increases its offered load to 80 Mb/s on VP *S2*. This causes *S4* to become overloaded and the bottleneck of *S1* to shift from *V1* to *V2*. As in the previous experiment, the bandwidth allocation to flows *S1*–*S4* is determined by inter-class and intra-class fairness requirements. Unlike in the previous experiment, class *0* has a higher bandwidth guarantee at $V2$ than class *I*. This prevents the throughput of *S1* from decreasing. The guarantees of class *0* and class *II* are again increased to 21.6% (33.48Mb/s) and 32.4% (50.22 Mb/s), respectively, on both *V1* and *V2*, in accordance with class level fairness.

Long-term Regulation

In this experiment, we consider that the connectionless traffic has to compete for bandwidth with the connection-oriented traffic in the ATM network. The starting conditions for the set of flows is the same as that obtained at $t = 40$ seconds in the experiment in . The simulation results are summarized in Figure 4.7, which comprises three pairs of graphs. The top pair shows the bandwidth allocated to the individual flows at the two links. The middle pair of graphs display the bandwidth guaranteed to the flow classes at *V1* and *V2*. Finally, the bottom pair of graphs show the bandwidth allocated to the VPs *V1* and *V2*. We show how the protocol copes with a sudden increase in connection-oriented traffic. We assume that the ATM network gives higher priority to connection-oriented traffic and therefore preempts bandwidth from the VPs dedicated to the network of CLS.

- At t = 50, the total offered load of connection-oriented traffic, $\Lambda_{l_{CO}}$, increases to 600 Mb/s on both ATM links. As the connectionless traffic is currently using bandwidth in excess of the minimum guarantee, long-term regulation causes bandwidth to be preempted from the VPs *V1* and *V2* and allocated to the VPs carrying connection-oriented traffic. The capacities of both *V1* and *V2*, $C_{l_{CL}}$, are reduced from 155 Mb/s to $M_{l_{CL}}$, i.e., 93 Mb/s. The total bandwidth allocated to connection-oriented traffic is now 529 Mb/s. Throughputs of all flows are reduced proportionately, while still satisfying inter-class and intra-class fairness requirements. The class guarantees to all flow classes are also reduced proportionately while maintaining class level fairness.

- At t = 100, $\Lambda_{l_{CO}}$ is reduced to 512 Mb/s, making 17 Mb/s available to connectionless traffic. As the VPs *V1* and *V2* were fully utilized in the previous VP update interval, long-term regulation allows their capacity, to increase by 17 Mb/s each. The new capacity for both VPs is $C_{l_{CL}}$ = 110 Mb/s. We observe that all flows increase their throughputs immediately while maintaining both inter-class and intra-class fairness. The class guarantees are also increased proportionately in accordance with class level fairness criteria.

- At t = 150, $\Lambda_{l_{CO}}$ is further reduced to 489 Mb/s, making 23 Mb/s available to connectionless traffic. As both VPs have fully utilized their allotted capacities, the unutilized bandwidth of 23 Mb/s is made available to them. The new VP capacity is $C_{l_{CL}}$ = 133Mb/s for both *V1* and *V2*. The through puts of individual flows and the class guarantees of flow classes are increased accordingly. It is to be noted here that the protocol ensures that inter-class, intra-class and class level fairness criteria are still satisfied.

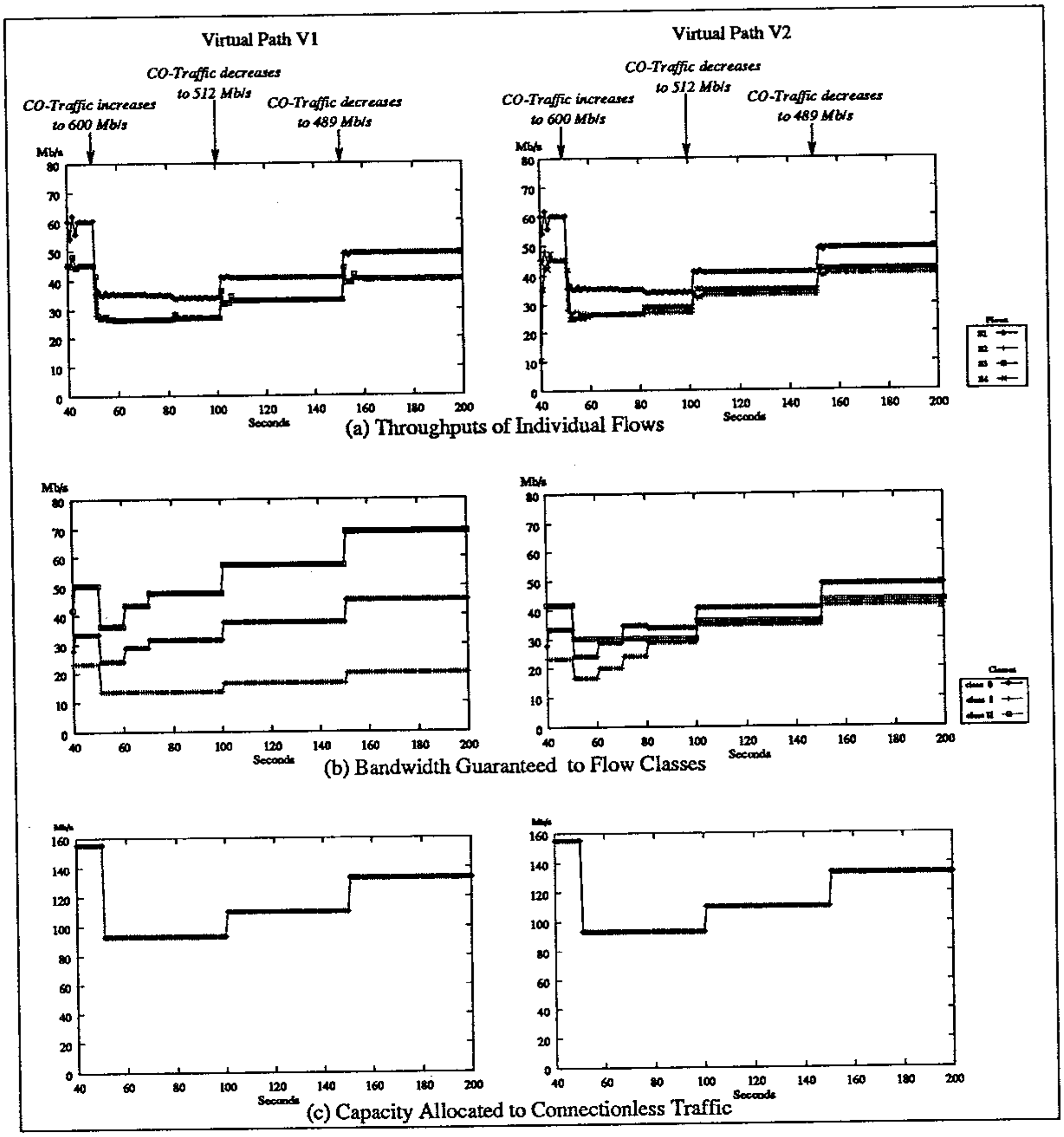


Figure 4.7: Long-term Regulation.

Chapter 5

Conclusions

With the rapid development of digital multimedia technologies, the characteristics of the traffic in computer networks has undergone a dramatic transformation. The introduction of delay-sensitive multimedia traffic to computer networks has added to the complexity of allocating bandwidth to traffic flows in an internetwork. The need to provide performance guarantees to multimedia flows, coupled with the large bandwidth required by such traffic, has shown up shortcomings in the existing best-effort service model. Existing flow and congestion control mechanisms, which have been designed to provide best-effort service to data traffic, are not suitable in a network in which traffic flows with widely different service requirements have to coexist. In particular, multimedia flows in existing data networks such as the Internet, have their service disrupted by the sudden emergence of a data traffic flow. Adopting an explicit resource reservation service model would let the network provide hard service guarantees to multimedia traffic, but would create compatibility problems with existing network applications. In addition, resource reservation will not allow efficient statistical multiplexing of traffic. Multi-level bandwidth allocation mechanisms allows the network to isolate the different classes of traffic, preventing data traffic from disrupting multimedia traffic, and vice-versa, while maintaining the best-effort service paradigm of traditional packet-switched networks.

In this thesis, we have presented two bandwidth allocation mechanisms, that regulate flows belonging to different traffic classes in, first, a traditional packet-switched internetwork, and second, in an ATM network supporting connectionless traffic along with connection-oriented traffic. Both mechanisms impose a resource regulation mechanism on top of an underlying connectionless, best-effort service. Bandwidth allocation is performed at multiple levels, allowing the total available bandwidth to be fairly distributed, first among traffic classes, and second, among flows belonging to a particular traffic class. Simple network protocols have been developed to implement each of the two mechanisms in a distributed manner, with a minimum of overhead. Simulation results have shown that both protocols achieve maximal bandwidth allocation while satisfying fairness criteria for both traffic classes and individual traffic flows. Our results indicate that a multi-level bandwidth allocation mechanism will allow both data and multimedia traffic to coexist on the same network, without causing undue disruptions in service.

In §5.1, we highlight the contributions of this thesis by chapter and in §5.2, discuss directions for future work.

5.1 Summary of Contributions

In Chapter 1, we introduce the problem of regulating flows with timing requirements, e.g., flows from multimedia or real-time applications, in a network where delay-sensitive flows must coexist with traditional bursty data traffic. The existing best-effort paradigm is incapable of guaranteeing delay and jitter bounds to multimedia traffic, absent any resource reservation or admission control mechanisms. Resource reservation mechanisms, coupled with admission control procedures, can guarantee timing constraints to multimedia flows, but they are not suitable for regulating bursty data traffic, which is unable to predict its service requirements in advance. Further, resource reservation will not allow the network to statistically multiplex flows in order to efficiently utilize the available bandwidth.

In Chapter 2, we survey the techniques proposed for bandwidth regulation of flows in the context of two different network models, the traditional packet-switched internetwork, and the newly developed Asynchronous Transfer Mode (ATM). While several schemes have been proposed to regulate either traffic classes, or individual traffic flows in a network, none of the schemes address the issue of simultaneously regulating bandwidth to both traffic classes and individual flows. We also observe that end-to-end rate control mechanisms, of the type that we have proposed, are more efficient in congestion control in high-speed networks than mechanism which enforce regulation based on purely local information.

In Chapter 3, we present two bandwidth regulation mechanisms. We first define a bandwidth regulation mechanism for traffic flows in a traditional packet-switched internetwork. The regulation mechanism satisfies two fairness conditions: *inter-class* fairness, for allocating bandwidth among the various flow classes, and *intra-class* fairness for allocating bandwidth among flows belonging to the same flow class. The second bandwidth regulation mechanism is intended for use in an ATM network that supports connectionless traffic by means of connectionless servers. As before, the connectionless traffic consists of multiple traffic classes. Bandwidth regulation is achieved at three different levels. *Short-term* bandwidth regulation is similar to the mechanism developed for the packet-switched internetwork. *Medium-term* regulation is concerned with monitoring and allocating so-called *class guarantees* to the flow classes. Lastly, *long-term* regulation is concerned with bandwidth allocation to connectionless traffic as a whole.

In Chapter 4, we present two sets of protocols, which implement the abovementioned regulation mechanisms. The so-called p–protocol performs bandwidth regulation in the context of a packet-switched internetwork. It enforces both inter-class and intra-class fairness conditions. The c–protocol is designed to regulate connectionless traffic in an ATM network. It performs bandwidth regulation at the three levels discussed above. We have performed simulation experiments which verify that our protocols adapt to changing load levels in the network, and rapidly settle at allocation levels, which have been verified to be the maximum possible allocations under the network loads.

5.2 Future Work

The present thesis has a number of avenues for future research.

Several issues have to be addressed in a real-world implementation of the p–protocol. As connectionless traffic sources are unable to predict their traffic load in advance, an implementation of the protocol would have to estimate the traffic load. Among the methods that could be considered, would be to use the backlog of packets at the access gateways as an estimate of the load. Another approach would be to use the the volume of traffic already transmitted by the flow as an estimate of its future load. Exponential averages might be used to smooth minor fluctuations in the traffic load.

Another potential area of investigation would be the interaction of the protocol with the routing mechanisms. The protocol currently assumes fixed routes, which may be acceptable only if routing changes are infrequent. If routing changes are of the order of the update interval of the p–protocol, then the network could be continuously in the convergence phase of the protocol. One possible solution would be to incorporate the share and surplus values obtained by the p–protocol into the routing metric.

Both the p–protocol and the c–protocol assume that the control packets, which contain state information, are never dropped in the network. This condition can be imposed in a real network by using a reliable out-of-band protocol for exchanging state information. An alternative implementation would have each flow periodically transmitting its state to the bottleneck gateways. This solution would maintain only soft-state information at the gateways, and provide robustness in the presence of packet losses.

The c–protocol, which has been developed for an ATM network using CLSs, could be adapted for use in ATM networks that use other mechanisms for supporting connectionless traffic. For example, the c–protocol could be adapted for use in an ATM network that interconnects local area networks using semi-permanent virtual circuits. The protocol would monitor the utilization of the VCs and modify their bandwidth according to the usage.

Another area where the c–protocol may be used is in the context of ABR services. The c–protocol would be used in conjunction with the rate-based congestion control mechanism adopted by the ATM Forum. The latest draft specifications of the ABR service allow sources to specify their desired rates in the Explicit Rate (ER) field of RM cells. Intermediate switches could use the c–protocol to calculate rate assignments for individual flows, which would satisfy fairness criteria. The calculated rates would be transmitted to the sources by using the ER field of control cells. This composite mechanism will give networks with the ability to perform fair bandwidth allocations, while retaining the ability to react rapidly to local congestion.

Bibliography

[1] D.F. Box, D.P. Hong, and T. Suda. Architecture and Design of Connectionless Data Service for a Public ATM Network. In *Proc. IEEE INFOCOM'93*, pages 722–731, 1993.

[2] P. E. Boyer and D. P. Tranchier. A Reservation Principle with Applications to the ATM Traffic Control. *Computer Networks and ISDN Systems*, 24:321–334, 1992.

[3] J. D. Cavanaugh and T. J. Salo. Internetworking with ATM WANs. Technical report, Minnesota Supercomputing Center, December 1992.

[4] G. H. Clapp. LAN Interconnection Across SMDS. *IEEE Network*, 5(5):25–32, September 1991.

[5] P. Crocetti, G. Gallassi, and M. Gerla. Bandwidth Advertising for MAN/ATM Connectionless Internetting. In *Proc. IEEE INFOCOM'91*, pages 1145–1150, 1991.

[6] J. R. Davin and A. T Heybey. A Simulation Study of Fair Queueing and Policy Enforcement. *Computer Communication Review*, 20(5):23–39, October 1990.

[7] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *Proc. ACM Sigcomm '89*, pages 1–12, 1989.

[8] P. Crocetti *et. al.* ATM-based SMDS for LANs/MANs Interconnection. In *Proc. International Switching Symposium*, volume 1, pages 87–91, October 1992.

[9] D. Ferrari and D. C. Verma. A Scheme for Real-Time Channel Establishment in Wide-Area Networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, April 1990.

[10] S. Floyd. Link-Sharing and Resource Management Models for Packet Networks. Technical report, Lawrence Berkeley Laboratory, September 1993. ftp://ftp.ee.lbl.gov/papers/link.ps.Z.

[11] M. Gerla, T. C. Tai, and G. Gallassi. Internetting LAN's and MAN's to B-ISDN's for Connectionless Traffic Support. *IEEE Journal on Selected Areas in Communications*, 11(8):1145–1159, October 1993.

[12] N. Giroux. Specification of the ABR Service. ATM Forum Contribution 93-935, November 1993.

[13] S. J. Golestani. A Stop-and-Go Queueing Framework for Congestion Management. In *ACM SIGCOMM'90 Symposium, Communications Architecture and Protocols*, pages 8–18, September 1990.

[14] E. L. Hahne, C. R. Kalmanek, and S. P. Morgan. Flow Control on a High-Speed Wide-Area Data Network. *Computer Networks and ISDN Systems*, 26(1):29 – 43, September 1993.

[15] J. Y. Hui. Resource Allocation for Broadband Networks. *IEEE Journal on Selected Areas in Communications*, 6(9):1598–1608, December 1988.

[16] ITU-T. CCITT Recommendation I.327: B-ISDN Functional Architecture. Studygroup XVIII. Geneva, March 1991.

[17] ITU-T. ITU-T Recommendation I.121: Broadband Aspects of ISDN. Studygroup XVIII. Geneva, April 1991.

[18] ITU-T. ITU-T Recommendation I.364: Support of Broadband Connectionless Data Service on B-ISDN. Studygroup XVIII. Geneva, March 1994.

[19] V. Jacobsen. Congestion Avoidance and Control. In *Proc. ACM Sigcomm '88*, pages 314–329, August 1988.

[20] J. M. Jaffe. Bottleneck Flow Control. *IEEE Transactions on Communications*, 29(7):954–962, July 1981.

[21] Charles R. Kalmanek, Hemant Kanakia, and Srinivasan Keshav. Rate Controlled Servers for Very High-Speed Networks. In *IEEE Global Telecommunications Conference, San Diego, California*, December 1990.

[22] S. Keshav. REAL: A Network Simulator. Technical Report 88/472, Computer Science Department, University of California, Berkeley, December 1988.

[23] B.G. Kim and P. Wang. ATM Network: Goals and Challenges. *Communications of the ACM*, 38(2):30–38, February 1995.

[24] H. T. Kung, T. Blackwell, and A. Chapman. Credit-Based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multiplexing. In *Proc. ACM SIGCOMM'94*, pages 101–114, September 1994.

[25] T. Van Landegem and R. Peschi. Managing A Connectionless Virtual Overlay Network on Top of An ATM Network. In *Proc. IEEE ICC'91*, pages 988–992, 1991.

[26] C.L. Liu and J.W. Layland. Scheduling Algorithms for Multiprogramming In A Hard Real-time Environment. *Journal of ACM*, 20(1):46–61, January 1973.

[27] L. Mongiovi, M. Farrell, and V. Trecordi. A Proposal for Interconnecting FDDI Networks Through B-ISDN. In *Proc. IEEE INFOCOM'91*, pages 1160–1167, April 1991.

[28] E. Monteiro, F. Boavida, and V. Freitas. A Fairness Analysis of LAN/WAN Protocol Relays. *Computer Networks and ISDN Systems*, 26:379–388, 1993.

[29] P. Newman. ATM Local Area Networks. *IEEE Communications Magazine*, pages 86 – 98, March 1994.

[30] C. O'Neill. Fairness Discarding for Congestion Control in ATM Networks. In *Proc. Australian Broadband Switching and Services Symposium '92*, volume 1, pages 185–192, July 1992.

[31] K. K. Ramakrishnan and R. Jain. A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with A Connectionless Network Layer. In *Proc. ACM SIGCOMM'88*, August 1988.

[32] K. K. Ramakrishnan and R. Jain. A Binary Feedback Scheme for Congestion Avoidance in Computer Networks. *ACM Transactions on Computer Systems*, 8(2):158–181, May 1990.

[33] L. Roberts. Enhanced Proportional Rate Control Algorithm. ATM Forum Contribution 94-0735R1, August 1994.

[34] S. Shenker, D. D. Clark, and L. Zhang. A Scheduling Service Model and a Scheduling Architecture for an Integrated Services Packet Network, 1993. Preprint.

[35] M. Steenstrup. Fair Share for Resource Allocation. Technical report, BBN Systems and Technologies, ftp://clynn.bbn.com/pub/docs/FairShare.draft9312.ps, December 1992.

[36] H. Suzuki and F. Tobagi. Fast Bandwidth Reservation Scheme with Multilink and Multipath Routing in ATM Networks. In *Proc. IEEE INFOCOM'92*, pages 2233–2240, May 1992.

[37] D. Verma, H. Zhang, and D. Ferrari. Guaranteeing Delay Jitter Bounds in Packet Switching Networks. In *Proceedings of Tricomm'91*, April 1991.

[38] M. Yamamoto, T. Hirata, C. Ohta, T. Hode, H. Okada, and Y. Tezuka. Traffic Control Scheme for Interconnection of FDDI Networks through ATM Network. In *Proc. of IEEE INFOCOM'93*, pages 411–420, 1993.

[39] N. Yin and M. G. Hluchyj. On Closed-Loop Rate Control for ATM Cell Relay Networks. In *Proc. IEEE INFOCOM'94*, pages 99–108, June 1994.

[40] N. Yin and M. G. Hluchyj. On Closed-Loop Rate Control for ATM Cell Relay Networks. ATM Forum Contribution 94-0438R2, September 1994.

[41] H. Zhang and S. Keshav. Comparison of Rate-Based Service Disciplines. In *Proc. ACM SIGCOMM'91*, pages 113–121, September 1991.

[42] Lixia Zhang. Virtual CLock: A New traffic Control Algorithm for Packet Switching Networks. In *Proc. ACM SIGCOMM'90*, pages 19–29, September 1990.

[43] M. Zukerman and S. Chan. Fairness in ATM Networks. *Computer Networks and ISDN Systems*, 26(1):109–117, September 1993.

[44] M. Zukerman and S. Chan. Congestion Control by Maintaining Fairness in High Speed Data Networks. In *IEEE GLOBECOM'94*, volume 3, pages 1576–1580, November 1994.