

Bandwidth Regulation of Real-Time Traffic Classes in Internetworks *

Jörg Liebeherr * Ian F. Akyildiz ** Debapriya Sarkar *

* Computer Science Department
University of Virginia
Charlottesville, VA 22903

** School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Abstract

New network applications which involve transmission of continuous media data, such as audio and video conferencing, introduce immense challenges for the design of packet-switching internetworks. Existing flow and congestion control mechanisms have been shown to be ineffective for supporting the real-time requirements of continuous media data transfers. We propose a novel bandwidth regulation mechanism which improves the ability of the network to cope with multiple real-time and non real-time traffic classes. The mechanism achieves regulation of link bandwidth at two levels. At one level, bandwidth is dynamically regulated between different traffic classes. We introduce the concept of *inter-class regulation* which enforces that the bandwidth left unused by some traffic classes is assigned equally to traffic classes with high bandwidth demands. At the second level, bandwidth regulation is enforced on packet flows from the same class. Each end-to-end packet flow from the same class has identical bandwidth constraints if their routes share the link with the smallest capacity for this class. This concept is referred to as *intra-class regulation*. We show that a bandwidth assignment which provides both *intra-class* and *inter-class regulation* without unnecessary waste of bandwidth is uniquely determined. We present a simple distributed protocol that achieves *intra-class* and *inter-class regulation* in a general internetwork. The protocol does not require network gateways to maintain state information on individual traffic flows, and adapts quickly to changes in the traffic load. The effectiveness of the protocol is demonstrated by simulation experiments.

Key Words: Bandwidth Regulation, Fairness, Flow Control, Real-Time Traffic, Packet Switching, Internetworks.

*The work of Jörg Liebeherr and Debapriya Sarkar was supported in part by the National Science Foundation under Grant No. NCR-9309224.

1 Introduction

Until recently, traffic on the Internet was dominated by applications for file transfers, electronic mail, electronic bulletin boards, and remote login [1, 13]. This type of traffic requires reliable transport service at the user level, but is only moderately sensitive to the amount and the variance of end-to-end delays. With the availability of audio/video hardware, numerous applications have been developed which enable the participation in audio and video-conferencing over the Internet. The transmission of audio and video prefers, but does not require a reliable transport service. However, transmission of audio and video data is very sensitive to end-to-end network delays, and to variations of the delays.

There is an ongoing discussion whether traditional packet-switching network, such as the Internet, can cope with the challenges introduced by the new applications with real-time requirements. We briefly review three main positions in this discussion:

1. *Do Nothing*: Obviously, this solution is appropriate if sufficient network resources are always available. Additionally, one may argue that existing congestion control mechanisms have shown to be effective for controlling the pure volume of network traffic. However, recent experiences show that traditional congestion control methods are not satisfactory for controlling traffic with real-time requirements.
2. *Resource Reservation with Admission Control*: This approach argues that the stringent demands of real-time transmissions on network delay, variance of delays, bandwidth and error rate can only be met if the network reserves resources for each *flow*¹. Admission control functions determine if the network has sufficient resources to support a new flow. If the resources are not available, the flow will not be accepted.

The Tenet protocol suite [7, 8] is an example of a set of protocols which includes resource reservation and admission control functions. Resources can be allocated such that the requirements of flows are guaranteed even in worst-case situations. A different approach to resource reservation with admission control in internetworks is given in [4, 16, 20].

Note that resource reservation with admission control, if implemented in the Internet, will have serious implications. First, since network resources are dedicated to a particular flow, the network can no longer be viewed as a shared resource. If access to the Internet remains unrestricted, a malicious user could reserve an unproportional amount of network resources.

¹Throughout this paper, we use the term **flow** to denote an end-to-end, or host-to-host, packet stream. The term **flow class** is used to denote a category of flows. Assignment of flows to a given set of flow classes can be based on the application type, the protocol used, or the location of the traffic source [20].

Thus, one must define a policy that limits the amount of resources that an individual user can reserve. Admission control for flows implies that access to the network can be denied if resources are scarce. Hence, the network is no longer generally accessible to every user at all times.

3. *Resource Regulation 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. The difference between a resource reservation scheme with admission control and a resource regulation scheme is that the former can provide *absolute* performance guarantees to flows, whereas the latter only provides *relative* performance guarantees.

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, the so-called *flow classes*.

In flow class regulation schemes, the network reserves a fixed amount of resources for a flow class, but permits other flow classes to utilize resources that are left unused. A different policy for resource regulation is to enforce fairness conditions for all flows in the network. Ideally, however, a resource regulation mechanism should simultaneously enforce policies for both flow classes and individual flows. Until now, such a resource mechanism has not been proposed.

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.

This study addresses the problem of regulating the use of link bandwidth in an internetwork without admission control functions. In today's internetworks, link bandwidth is the scarcest resource. Buffer overflows at gateways², excessive end-to-end delays and delay variations mainly result from the lack of available link bandwidth. We present a novel approach for regulating link bandwidth for both flow classes and individual flows. The objectives of our approach is to implement specific policies to distribute available bandwidth between flow classes (*inter-class regulation*) and

²**Gateway** is the term used in this paper for switching nodes in an internetwork. The term **access gateways** is used for gateways at the outside boundary of an internetwork.

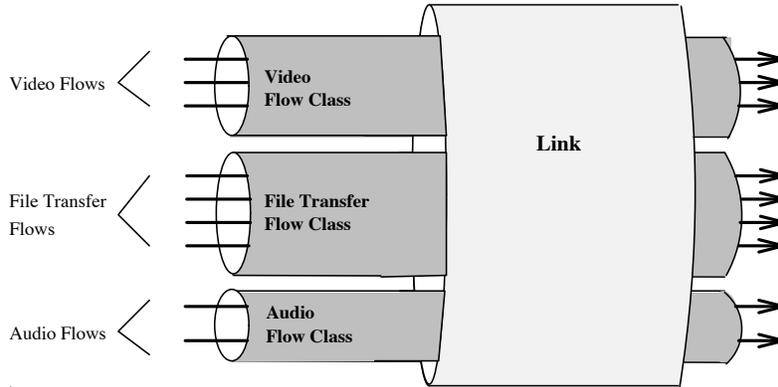


Figure 1: Flows and Flow Classes at a Network Link.

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 Regulation:*

At each link in the network, a flow class may obtain a bandwidth guarantee. If the flows of a flow class do not fully utilize this guarantee, the unused bandwidth, the so-called *surplus*, is made available to other flow classes. In this study we consider that surplus bandwidth is divided evenly among all flow classes which can utilize the bandwidth for transmission.

- *Intra-Class Regulation:*

For each flow class, a so-called *share* at a network 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 flow's route, the *bottleneck link*. Hence, two flows from the same class and with the same bottleneck link have identical end-to-end throughput constraints.

In Figure 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. 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.

To our knowledge, our work is the first attempt to simultaneously regulate link bandwidth at the flow class and the flow level. We show that both regulation concepts must be addressed together, otherwise, usable bandwidth in the network will be wasted.

We present a distributed protocol that implements the above regulation policies. The overhead of the protocol consists of a few counters at each gateway, a control protocol that periodically disseminates the values of the counters to all gateways at the boundary of the network, so-called *access gateways*. We also require a rate control mechanism at the traffic sources. For the regulation protocol, internal gateways need not keep state information on individual flows, and traffic sources need not transmit their bandwidth requirements to network gateways or to other traffic sources. We will show that the protocol quickly stabilizes after changes of the network load.

The remaining sections are structured as follows. In Section 2 we review previous work on resource regulation for packet-switching networks. In Section 3 we formally introduce our notion of intra-class and inter-class bandwidth regulation. We show that there is a unique solution to a bandwidth regulation scheme that implements the abovementioned inter-class and intra-class regulation policies without unnecessary waste of bandwidth. In Section 4 we present a protocol which implements the bandwidth regulation mechanism. We use simulation experiments to demonstrate the effectiveness of the protocol. In Section 5 we conclude our results.

2 Related Work

The problem of regulating link bandwidth in a packet-switching network has been addressed previously. Objectives of existing bandwidth regulation algorithms are to either reach some notion of fairness between flows within a single flow class, or to control link bandwidth allocation to flow classes without considering individual flows. So far, no regulation mechanism has been proposed that, at the same time, regulates bandwidth for individual flows *and* for flow classes in a general network.

First results on bandwidth regulation were obtained in the 1980s for traditional packet-switching networks with connection-oriented service. Currently, research on bandwidth regulation mechanisms is conducted in two domains. In the previous section, we have discussed the need for new traffic control methods in internetworks. The second domain of research is B-ISDN, where new bandwidth control algorithms are needed for efficient implementations of connectionless data services [2].

One approach to bandwidth regulation is based on scheduling algorithms at the gateways. Fair Queueing and its variations [5, 6, 18], and Round-Robin [11], have shown to satisfy certain fairness criteria for either individual flows or flow classes, however, not for both. A disadvantage of regulation methods that are exclusively implemented at the network gateways, e.g., by scheduling algorithms, is that they can control usage of bandwidth only by dropping packets. However, if a packet is dropped at a gateway which is not located close to the flow source, the packet consumes bandwidth at all links between the source and the gateway which drops the packet. To overcome

this drawback, Hahne et. al. [12] proposed to support the Round-Robin discipline with a window based flow control mechanism.

A different type of bandwidth control regulates the traffic rate at the flow sources [10, 15]. In these studies, the objective of the regulation mechanisms is to ensure fairness conditions for individual flows, similar to our concept of *intra-class* regulation. However, regulation of bandwidth at the flow class level is not addressed.

A number of studies considers bandwidth regulation of flow classes, without providing mechanisms that regulate the bandwidth consumption of flows from the same class. In these studies, the objective of the regulation mechanisms is referred to as *link sharing*. Link sharing approaches provide some notion of inter-class regulation, but do not address at all bandwidth regulation of flows from the same class (intra-class regulation). For example, Steenstrup [21] proposes a hierarchical structure of flow classes with bandwidth guarantees for each class. Guarantees can be allocated statically or dynamically. Regulation of link bandwidth is performed exclusively at the gateways. Traffic measurements are used to adapt the throughput guarantees to actual transmitted traffic. Another approach to hierarchical link sharing is presented in [20]. Flows at the higher levels of the flow class hierarchy require an admission control entity. A third hierarchical and highly flexible approach to link sharing is presented by Floyd [9]. A drawback of the link sharing approaches is that they cannot control link bandwidth for individual flows unless there is only one flow in each class [9, 21], or admission control functions are used [20].

3 Bandwidth Allocations with Intra-class and Inter-class Regulation

We consider an arbitrary network of gateways which are connected by point-to-point links. Hosts access the network by connecting to so-called *access gateway*, i.e., gateways at the outside boundary of the network. 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).

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

At each link, flow class p may have a *bandwidth guarantee* of $G_{l_p} > 0$ with $\sum_{p \in \mathcal{P}} G_{l_p} < C_l$. Let \mathcal{P}_l denote the set of classes with a positive guarantee at link l , that is, $\mathcal{P}_l = \{p \in \mathcal{P} \mid G_{l_p} > 0\}$. If a class- p flow i has link l on its route, i.e., $i \in \Delta_{l_p}$, 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 $G_{l_0} = C_l - \sum_{p \in \mathcal{P}} G_{l_p}$.

Let the *surplus* of a flow class, ϕ_{l_p} be the maximum bandwidth that a class can utilize at a link in excess of its guarantee G_{l_p} . 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 fully utilize its guarantee.

Let the *share* of a class- p flow i at a link l , $\alpha_{i_p}(l)$ denote the maximum bandwidth that flow i can receive at a link l on its route. The share $\alpha_{i_p}(l)$ may be different at each link along the route of a flow, and may be different for flows from the same class that share the same link. The *bottleneck link* for a flow i , l_i^* , is the link on the route that has the smallest share, i.e., $\alpha_{i_p}(l_i^*) = \min_{l \in \mathcal{R}_i} \alpha_{i_p}(l)$.

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 Λ which contains the λ_i as elements. The throughput of all flows is given by the throughput set Γ which contains the γ_i as elements. Finally, we define the *regulation set* Φ as the set which contains the tuples $\{(\alpha_{i_p}(l), \phi_{l_p}) \mid l \in \mathcal{R}_i\}$ 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 1 *Given a network topology \mathcal{T} with offered load set Λ , throughput set Γ , and regulation set Φ . A bandwidth allocation is a relation $\Omega \subseteq \Lambda \times \Gamma \times \Phi$ such that*

1. $\gamma_i \leq \min(\lambda_i, \alpha_{i_p}(l_i^*))$ for all $i \in \mathcal{F}$.
2. $\sum_{p \in \mathcal{P} \cup \{0\}} \sum_{i \in \Delta_{l_p}} \gamma_i \leq C_l$ for all $l \in \mathcal{L}$.
3. $\sum_{i \in \Delta_{l_p}} \gamma_i \leq G_{l_p} + \phi_{l_p}$ for all $p \in \mathcal{P}_l$.

The first condition enforces that the throughput of a flow cannot exceed its load or the share at its bottleneck link. The second condition enforces that the total throughput from all flows at a link is limited by the capacity of the link. The third condition enforces that the throughputs from the flows of the same class cannot exceed the bandwidth guarantee by more than the surplus.

Next we introduce bandwidth allocations which provide *inter-class* regulation. 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. Note that a flow class may not utilize its guarantee at a link for three reasons. First, the total load of the class can be less than its guarantee. Second, the sum of the flows' shares from this class can be less than the guarantee. Third, the throughput of class- p flows is limited due to restrictions at other links. A bandwidth allocation with inter-class regulation assigns the unused bandwidth equally among flow classes which can take advantage of the additional capacity. Thus, the maximum bandwidth at link l that a class p can 'borrow' from the guarantees of other classes is identical for all classes, and we obtain for the surplus values that $\phi_l \equiv \phi_{lp}$ for all classes $p \in \mathcal{P}_l$.

The following definition provides a formal definition of inter-class regulation. In the definition, C_{lp} is used to denote the *available bandwidth* of flow class p at link l with $C_{lp} \geq \sum_{j \in \Delta_{lp}} \gamma_j$.

Definition 2 *A bandwidth allocation is said to provide inter-class regulation 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., $\phi_l \equiv 0$, provides inter-class regulation. However, such an allocation results in a waste of link bandwidth. In Lemma 1 we state that by selecting ϕ_l as large as possible, one can make the entire link bandwidth available for transmission.

Lemma 1 *Given a bandwidth allocation with inter-class regulation. 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 $q \in \mathcal{P}_l$.

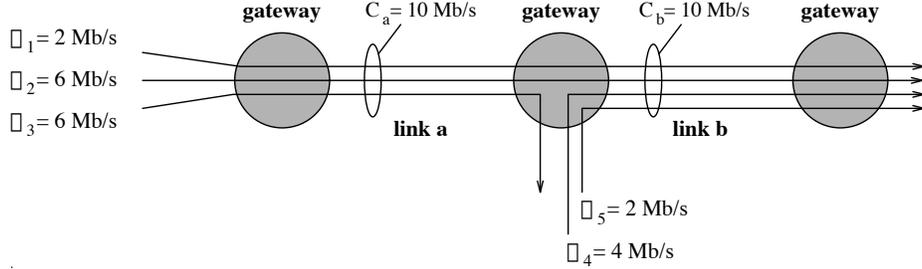


Figure 2: Intra-Class Fairness in a Network with Two Links.

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

Next we discuss bandwidth allocations with intra-class regulation. For the special case of only one flow class the regulation policy is similar to [15]. *Intra-class regulation* is concerned with distributing C_{lp} , the bandwidth available to a flow class p at a link l , to the flows from this class. Recall that a bandwidth allocation defines for each flow i with link l on its route a share $\alpha_{ip}(l)$ that gives the maximum bandwidth available to this flow at this link. Intra-class regulation enforces that the shares of flows from the same class are identical, i.e., for each flow $i \in \Delta_{lp}$ we have $\alpha_{ip}(l) \equiv \alpha_p(l)$. As a result, if two flows i and j of the same flow class have the same bottleneck link, i.e., $l_i^* = l_j^*$, then both flows have the identical throughput constraints. Bandwidth allocations with intra-class regulation are formally defined as follows.

Definition 3 A bandwidth allocation is said to provide intra-class regulation 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 $i \in \mathcal{F}_p$

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

As an example of intra-class regulation, consider the network in Figure 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 as follows:

$\lambda_1 = 2 \text{ Mb/s}$	$\lambda_2 = 6 \text{ Mb/s}$	$\lambda_3 = 6 \text{ Mb/s}$	$\lambda_4 = 4 \text{ Mb/s}$	$\lambda_5 = 2 \text{ Mb/s}$
------------------------------	------------------------------	------------------------------	------------------------------	------------------------------

Setting the share values to

$$\alpha_a = 5\text{Mb/s} \quad \text{and} \quad \alpha_b = 3\text{Mb/s},$$

respectively, for link a and link b , we obtain the following throughput values from Definition 3:

$\gamma_1 = 2 \text{ Mb/s}$	$\gamma_2 = 3 \text{ Mb/s}$	$\gamma_3 = 5 \text{ Mb/s}$	$\gamma_4 = 3 \text{ Mb/s}$	$\gamma_5 = 2 \text{ Mb/s}$
-----------------------------	-----------------------------	-----------------------------	-----------------------------	-----------------------------

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 $\gamma_3 = \min(\lambda_3, \alpha_a) = \alpha_a$.

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 $\alpha_b < 3 \text{ Mb/s}$, or will violate the constraints for a bandwidth regulation with intra-class regulation, e.g., if $\alpha_b > 3 \text{ Mb/s}$. 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 *maximal shares*. In Lemma 2 we give the condition that must hold if the shares in a network with multiple flow classes are maximal.

Lemma 2 *The values of the shares in a bandwidth allocation with intra-class regulation 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 $\sum_{j \in \Delta_{l_i^* p}} \gamma_j < G_{l_i^* p}$, 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

The given definitions of bandwidth regulation are concerned with allocating bandwidth to flows of the same flow class (*intra-class regulation*), and to entire flow classes (*inter-class regulation*). Indeed, inter-class and intra-class regulation are two independent concepts. One can easily imagine bandwidth allocations that provide inter-class regulation but do not offer intra-class regulation, and vice versa. In particular, all proposals for hierarchical link sharing [9, 20, 21] provide some regulation for flow classes (different from the presented inter-class regulation), but do not solve the regulation problem for flows from the same class.

We can follow from Lemma 1 that a bandwidth allocation with intra-class regulation but without maximal shares can result in a waste of available bandwidth. Likewise, Lemma 2 implies that a bandwidth allocation with inter-class regulation but without maximal surplus values may leave bandwidth unused. Therefore, one is interested in finding bandwidth allocations which offer inter-class regulation with maximal surplus values, and intra-class regulation with maximal shares. In Theorem 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 1 *Given a network with topology \mathcal{T} with offered load set Λ . Then there exists a unique bandwidth allocation that provides intra-class regulation with maximal shares $\alpha_p^*(l)$ and inter-class regulation with maximal surplus values ϕ_l^* . The maximal shares and the maximal surplus values are determined by a solution of the following equation system ³.*

$$\alpha_p^*(l) = \begin{cases} \infty & \text{if } \mathbf{O}_{lp} = \emptyset \\ \frac{G_{lp} + \phi_l^* - \Theta_{lp}}{|\mathbf{O}_{lp}|} & \text{otherwise} \end{cases} \quad (1)$$

and

$$\phi_l^* = \begin{cases} \infty & \text{if } \bigcup_{q \in \mathcal{P}_l} \mathbf{O}_{lq} = \emptyset \\ \frac{C_l - \sum_{\mathbf{O}_{lq} \neq \emptyset} G_{lq} - \sum_{\mathbf{O}_{lq} = \emptyset} \Theta_{lq}}{|\{q \in \mathcal{P}_l \mid \mathbf{O}_{lq} \neq \emptyset\}|} & \text{otherwise} \end{cases} \quad (2)$$

subject to the side conditions:

$$G_{lp} + \phi_l^* - \Theta_{lp} \geq 0 \quad (3)$$

$$C_l - \sum_{\mathbf{O}_{lq} \neq \emptyset} G_{lq} - \sum_{\mathbf{O}_{lq} = \emptyset} \Theta_{lq} \geq 0 \quad (4)$$

where:

$$\Theta_{lp} = \sum_{i \in \mathbf{U}_{lp}} \lambda_i + \sum_{k \in \mathcal{L}} |\mathbf{R}_{lp}(k)| \cdot \alpha_p^*(k) \quad (5)$$

and the sets \mathbf{U}_{lp} , \mathbf{R}_{lp} , and \mathbf{O}_{lp} are defined for all $p \in \mathcal{P}_l$ as:

$$\mathbf{U}_{lp} = \{i \in \Delta_{lp} \mid \alpha_p^*(l) \geq \lambda_i, i \notin \bigcup_{k \in \mathcal{L}} \mathbf{R}_{lp}(k)\} \quad (6)$$

$$\mathbf{O}_{lp} = \{i \in \Delta_{lp} \mid l = l_i^*, \alpha_p^*(l) < \lambda_i\} \quad (7)$$

$$\mathbf{R}_{lp}(k) = \{i \in \Delta_{lp} \mid k = l_i^*, \alpha_p^*(k) < \lambda_i\} \quad \text{for } k \neq l \quad (8)$$

³In equations (1) and (2), $|X|$ denotes the cardinality of a set X .

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 $\mathbf{R}_{lp}(k)$ ($k \in \mathcal{R}_i$). \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 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 share at the respective bottleneck link, each restricted flow at link l is overloaded at some other link on its route.

Proof Idea: The complete proof of the theorem is too lengthy to be presented here [19]. Therefore, we will only discuss the main steps of the proof.

It can be shown that a solution to the equation system in (1) – (8) can be effectively constructed. The construction of the solution is performed with a nested iteration over the number of flow classes and the number of links. The uniqueness of the solution can be derived from the properties of the iterative algorithm. This part of the proof requires considerable effort. One can show that any bandwidth allocation which satisfies the equation system in (1) – (8), provides inter-class and intra-class regulation. Also, one can verify that the shares as calculated in (1), and the surplus values as calculated in (2) are maximal. It can be proven that any bandwidth allocation which offers intra-class regulation with maximal shares and inter-class regulation with maximal surplus values, is also a solution of (1) – (8). Finally, the uniqueness of the solution implies the unique existence of the desired bandwidth allocations. \square

An important implication of Theorem 1 is that inter-class and intra-class regulation cannot be addressed separately, unless one accepts the waste of bandwidth caused by not selecting maximal shares $\alpha_p^*(l)$ as in equation (1), or maximal surplus values ϕ_l^* as in equation (2). Note that the computation of the maximal shares at a link in (1) requires knowledge of the surplus value in (2). On the other hand, the surplus value at a link in (2) is dependent on the values of the shares in (1). Results similar to our Theorem 1 can be developed for different bandwidth regulation definitions, in particular, for hierarchical link sharing schemes. Thus our theorem indicates that neglecting bandwidth control of individual flows as in the link sharing schemes will result in waste of bandwidth.

In the next section, we use Theorem 1 to derive a protocol that implements inter-class and intra-class regulation with maximal shares and maximal surplus values. We will show that the complexity of the desired bandwidth allocation can be achieved with a relatively simple protocol.

4 A Protocol for Inter-Class and Intra-Class Bandwidth Regulation

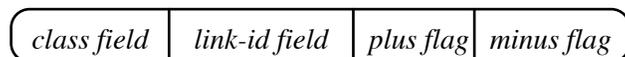
We present a protocol that is able to implement the mathematically developed inter-class and intra-class bandwidth regulation with maximal shares and surplus values from the previous section. The protocol is completely distributed and can be implemented with little overhead. A main advantage of our protocol is that it does not require any network entity to keep global state information. The computational overhead at network gateways is limited to maintaining a few counters. We present a simulation experiment to show that the achieved bandwidth regulation quickly converges to the theoretically correct values after load changes in the network.

For the sake of a clear presentation we make some assumptions for the network and the protocol. We assume that information on the offered load of a flow is available at its source. Also, the protocol does not address reliability issues. After the presentation of the protocol and the simulation experiment, we discuss how these assumptions can be relaxed.

4.1 Protocol Description

(a) Extensions to Packet Header

For the 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:



In the following, we will use ‘+’ to indicate a set *plus flag* in a packet header, ‘-’ to indicate a set *minus flag*, and ‘.’ to indicate that a flag is not set.

(b) Update Intervals and Rate Control at Sources

The 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 the following information content to all access gateways:

p	$gw:li$	$Share_p$
-----	---------	-----------

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 $Share_p(gw:li)$. After receiving the control packets, the access gateway which is closest to the source of a class- p flow, say flow i , calculates

$$Quota[i] = \min(Share_p(gw:li) \mid gw:li \text{ is on the route of class-}p \text{ flow } i) \quad (9)$$

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 $Load[i] \leq Quota[i]$, then flow i is *underloaded*.

For *underloaded* flows, the source sets the header of each packet to

p	$underloaded$	\cdot	\cdot
-----	---------------	---------	---------

.

- If $Load[i] > Quota[i]$, and $Quota[i]$, as calculated in equation (9), is such that $Quota[i] = Share_p(gw:li)$, then flow i is ‘*overloaded at gw:li*’.

In this case the source of the flow sets all packet headers to

p	$gw:li$	\cdot	\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* \implies *overloaded at gw:li*.

In this case, the source sets the header of the flow’s next packet to

p	$gw:li$	$+$	\cdot
-----	---------	-----	---------

.

- *overloaded at gw:li* \implies *underloaded*.

Then, the source sets the header of the next packet to

p	$gw:li$	\cdot	$-$
-----	---------	---------	-----

.

- *overloaded at gw1:li1* \implies *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

p	$gw:li2$	$+$	\cdot
-----	----------	-----	---------

,

and in the immediately following packet, the header is set to

p	$gw:li1$	\cdot	$-$
-----	----------	---------	-----

.

(d) Operations at the Gateways

Next we discuss the functions performed by a gateway, say gateway \mathbf{gw} . Each outgoing link of the gateway, say $\mathbf{gw:li}$, is assigned a capacity $\text{Cap}(\mathbf{gw:li})$ 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 $\mathbf{gw:li}$, denoted by $\text{Guar}_p(\mathbf{gw:li})$, gives the transmission guarantee of flow class p during an update interval. The gateway maintains two counters, $\text{Rate}_p(\mathbf{gw:li})$ and $\text{OL}_p(\mathbf{gw:li})$, and two variables, $\text{Share}_p(\mathbf{gw:li})$ and $\text{Surplus}_p(\mathbf{gw:li})$ for each flow class with $\text{Guar}_p(\mathbf{gw:li}) > 0$. The counters and variables are mandatory for default class ‘0’.

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

$$\begin{array}{c} \boxed{p \quad \text{underloaded} \quad \cdot \quad \cdot} \quad \text{or} \quad \boxed{p \quad \mathbf{gw:li1} \quad \cdot \quad \cdot} \\ \text{with} \\ \mathbf{gw1:li1} \neq \mathbf{gw:li} \end{array}$$

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

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

(e) Calculation of Share and Surplus Values

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

$$\text{Share}_p(\mathbf{gw:li}) = \begin{cases} \text{infinity} & \text{if } \text{OL}_p(\mathbf{gw:li}) = 0 \\ \frac{\text{Guar}_p(\mathbf{gw:li}) + \text{Surplus}_p(\mathbf{gw:li}) - \text{Rate}_p(\mathbf{gw:li})}{\text{OL}_p(\mathbf{gw:li})} & \text{otherwise} \end{cases} \quad (10)$$

and

$$\begin{aligned} \text{Surplus}_p(\mathbf{gw:li}) &= \\ &= \begin{cases} \text{infinity} & \text{if } \text{OL}_p(\mathbf{gw:li}) = 0 \text{ for all } p \\ \frac{\text{Cap}(\mathbf{gw:li}) - \sum_{\text{OL}_q(\mathbf{gw:li}) > 0} \text{Guar}_q(\mathbf{gw:li}) - \sum_{\text{OL}_q(\mathbf{gw:li}) = 0} \text{Rate}_q(\mathbf{gw:li})}{|\{q \mid \text{OL}_p(\mathbf{gw:li}) > 0\}|} & \text{otherwise} \end{cases} \quad (11) \end{aligned}$$

The results for the new values for $\text{Share}_p(\text{gw:li})$ and $\text{Surplus}_p(\text{gw:li})$ are sent to all access gateways in the abovementioned control packets with content $\boxed{p \mid \text{gw:li} \mid \text{Share}_p}$. Finally, the gateway resets its counters $\text{Rate}_p(\text{gw:li})$ to zero.

Note that equations (10) and (11) are based on our Theorem 1. In equations (10) and (11), infinity is chosen such that $\text{infinity} \gg \text{Cap}(\text{gw:li})$. Both equations can be computed for all flow classes without information on the share or surplus values at other gateways. By setting $\text{gw:li} \equiv l$, and by neglecting that Theorem 1 is expressed in terms of data rates, we obtain the following relation between equations (10) – (11) and Theorem 1 :

$$\begin{aligned}
 \text{Share}_p(\text{gw:li}) &= \alpha_p^*(l) \\
 \text{Surplus}_p(\text{gw:li}) &= \phi_l^* \\
 \text{Cap}(\text{gw:li}) &= C_l \\
 \text{Guar}_p(\text{gw:li}) &= G_{lp} \\
 \text{OL}_p(\text{gw:li}) &= |\mathbf{O}_{lp}| \\
 \text{Rate}_p(\text{gw:li}) &= \Theta_{lp}
 \end{aligned}$$

4.2 Simulation Experiment

To provide insight into the dynamics of our bandwidth regulation 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 [17]. 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 1250 Bytes. Propagation delays are small and set to $10\mu\text{s}$. Each source 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 3, 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 100 Mb/s. We simulate the behavior of five flows from three different flow classes: θ , I , and II . The bandwidth guarantees of the flow classes are identical at all links, and denoted by G_θ , G_I , and G_{II} . The guarantees are set to:

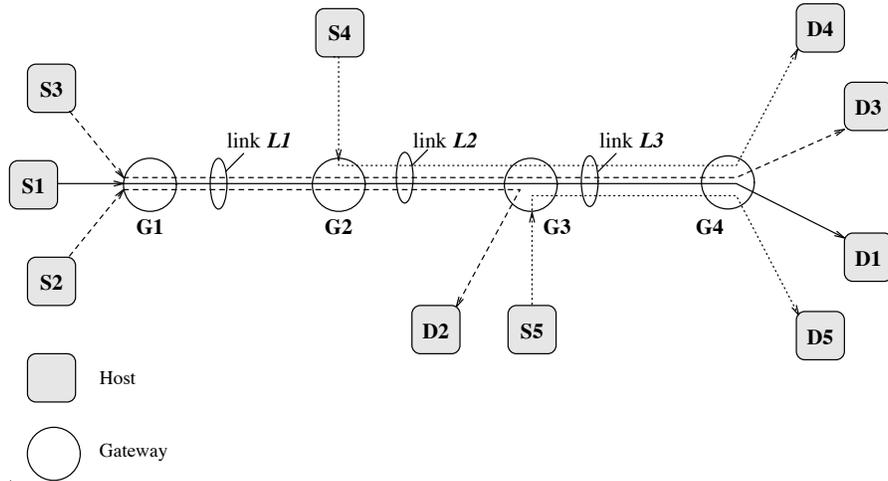


Figure 3: Simulated Network.

Flow (Source Host)	Destination Host	Route	Class	Offered Load	Start Time (<i>in s</i>)
S_1	D_1	(L_1, L_2, L_3)	0	10 Mb/s	$t = 0$
S_2	D_2	(L_1, L_2)	II	40 Mb/s	$t = 20$
S_3	D_3	(L_1, L_2, L_3)	II	70 Mb/s	$t = 40$
S_4	D_4	(L_2, L_3)	0	70 Mb/s	$t = 90$
S_5	D_5	(L_3)	I	60 Mb/s	$t = 140$

Table 1: Flow Parameters.

$$G_0 = 15 \text{ Mb/s for class } 0,$$

$$G_I = 30 \text{ Mb/s for class } I,$$

$$G_{II} = 55 \text{ Mb/s for class } II.$$

The parameters of the five flows in Figure 3, that is, source host, destination host, route, flow class membership, offered load, and time of first packet transmission, are summarized in Table 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. 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 θ 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 transmission 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 follows⁴:

	$\alpha_0(L1)$	$\alpha_I(L1)$	$\alpha_{II}(L1)$	ϕ_{L1}	γ_{S1}	γ_{S2}	γ_{S3}
Link $L1$	–	–	50	35	10	40	50

In Figure 4 it can be seen that the protocol quickly settles at the predicted values.

- At $t = 90$, flow $S4$ from class θ starts transmission on links $L2$ and $L3$ with an offered load of 70 Mb/s. Then, both classes θ and II require their respective bandwidth guarantees 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 θ 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}	γ_{S1}	γ_{S2}	γ_{S3}	γ_{S4}
Link $L2$	20	–	35	15	10	35	35	20

Within class θ , flow $S1$ is underloaded and $S4$ is overloaded at link $L2$. Note in Figure 4 that the throughputs of $S2$ and $S3$ drop to 35 Mb/s.

⁴The data in the tables is given in Mb/s. For clarity, we substituted the symbol ‘ ∞ ’ by ‘–’.

- 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 θ), 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 θ and flow $S5$ from class I can borrow the unused class- II guarantee on link $L3$. Note from Figure 4 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}
1	38.3	38.3	13.3	38.3

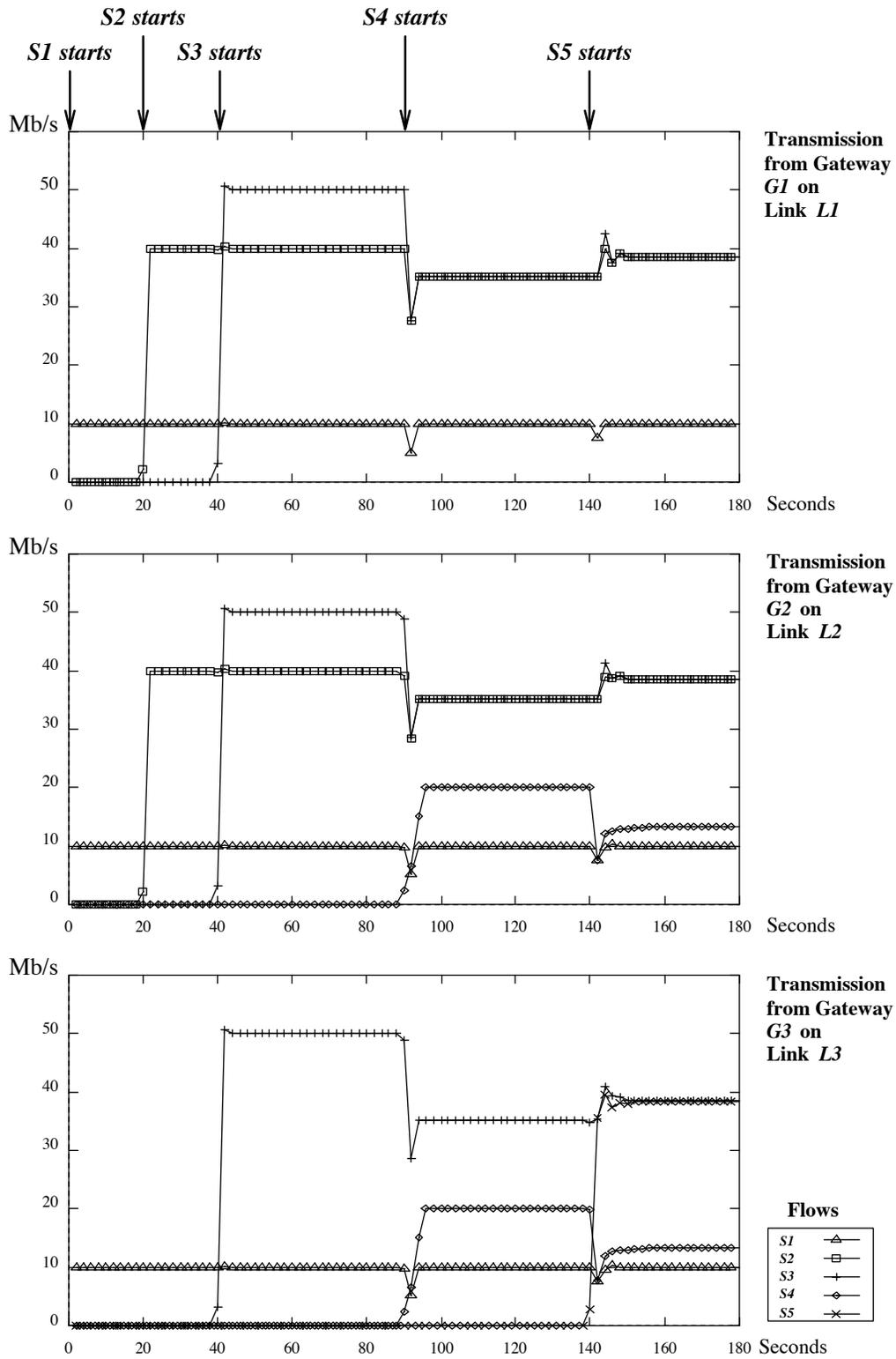


Figure 4: Simulation Results.

4.3 Discussion

For the protocol, we made a number of assumptions which must be addressed in any ‘real-world’ implementation. Here, we discuss how these assumptions can be relaxed.

- *Flow Sources*

Our protocol assumes that for each flow the desired traffic load, `Load[i]`, is available. This assumption can be relaxed by using the backlog of untransmitted packets at the rate controller as indicator of the load.

- *Scheduling*

We do not assume a particular scheduling algorithm for the gateways. In fact, for our simulation experiment we used FIFO scheduling at all gateways with excellent results. A better scheduling algorithm, such as Fair Queueing [6], could support the bandwidth regulation protocol. However, one should avoid complex scheduling algorithms which require state information on individual flows at the gateways.

- *Robustness in the Presence of Errors*

The protocol, as described, is sensitive to packet losses which contain information on state transitions on a flow, i.e., packets which have the *plus flag* or the *minus flag* set. There are a number of ways to increase the robustness of the protocol. An obvious solution is to use a reliable out-band protocol for sending information on state transitions to gateways. In another solution, gateways keep information on the identity of overloaded flows. This solution does not require *plus* and *minus flags*, since the state of a flow can be obtained by inspecting the *link-id field* of packet headers. However, maintaining information on overloaded flows at the gateways burdens gateways with considerable processing overhead. In yet another approach, each overloaded flow source periodically transmits its state to the bottleneck gateways, and the gateways periodically reset their information on overloaded flows. This solution, if properly implemented, keeps only *soft-state* information at gateways [3], and provides robustness in the presence of gateway failures.

- *Selection of Update Periods and Sensitivity During State Transitions*

The stability of the bandwidth regulation scheme is sensitive to the size of the update interval. The sensitivity should be similar to the selection of update periods in routing protocols.

A related issue is the sensitivity of the protocol towards state changes of flows. A single flow which constantly flip-flops between *underloaded* and *overloaded* states can prevent the entire network from converging to a stable bandwidth assignment. This problem can be reduced by making flows less adaptive to changes of the load, `Load[i]`, or the quota, `Quota[i]`. For example, by using exponential moving averages as in [21], load and state changes will not have immediate consequences

in the network.

If new flows start transmission, the throughput of existing flows can degrade temporarily for the duration of one update interval. This effect is due to the asynchronous nature of our protocol, and can be prevented by slowly increasing the transmission rates of new flows, similar to *slow start* in TCP [14].

- *Non-Cooperative Sources and Gateways*

Our protocol assumes that all sources are well-behaved in that they properly adjust their rate control parameters. Also we assume that all network gateways execute the bandwidth regulation protocol. So far, we have not explored the mechanisms needed to maintain a proper bandwidth regulation if some sources and gateways do not conform to our protocol.

- *Routing Issues*

In the entire study, we have assumed fixed routes for all flows. This is an acceptable assumption if route changes occur only infrequently. If this is not the case, each change of a route may result in a different values for shares and surplus, which in turn will result in a convergence phase of the bandwidth regulation mechanism.

Ideally, the routing protocol should cooperate with the bandwidth regulation protocol. For example, the routing procedures could include the values of $\text{Share}_p(\text{gw}:1)$ into the routing metric.

5 Conclusions

We have proposed a bandwidth regulation mechanism for controlling link bandwidth in internetworks. We have given two bandwidth regulation objectives for traffic in an internetwork, referred to as *inter-class regulation* and *intra-class regulation*. Inter-class regulation describes how different traffic classes, for example, video and file transfer traffic classes, share link bandwidth without considering the number of end-to-end traffic streams, so-called *flows*, in each class. Intra-class regulation enforces rules for dividing link bandwidth to flows from the same class. We have developed a theoretical framework for bandwidth regulation schemes in a general network, and formally showed the existence of a unique bandwidth assignment which simultaneously satisfies inter-class and intra-class regulation, and, in addition, does not waste link bandwidth. These results have been applied for the development of a distributed control protocol that achieves the desired bandwidth regulation. We have presented a simulation experiment and showed that the protocol quickly adapts to changes in the network load. We have discussed several extensions of our protocol which, for example, can provide robustness in the presence of errors or gateway failures.

References

- [1] R. Caceres, P. B. Danzig, S. Jamin, and D. J. Mitzel. Characteristics of Wide-Area TCP/IP Conversations. In *Proc. ACM Sigcomm '91*, pages 101–112, September 1991.
- [2] CCITT. CCITT Draft Recommendation I.c1s: Support of Broadband Connectionless Data Service on B-ISDN, December 1991.
- [3] D. D. Clark. The Design Philosophy of the DARPA Internet Protocols. In *Proc. Sigcomm '88*, pages 106–114, August 1988.
- [4] D. D. Clark, S. Shenker, and L. Zhang. Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanisms. In *Proc. Sigcomm '92*, pages 14–26, August 1992.
- [5] 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.
- [6] A. Demers, S. Keshav, and S. Shenker. Analysis and Simulation of a Fair Queueing Algorithm. In *Proc. Sigcomm '89*, pages 1–12, 1989.
- [7] D. Ferrari, A. Banerjea, and H. Zhang. Network Support for Multimedia - A Discussion of the Tenet Approach. Technical Report TR-92-072, International Computer Science Institute, Berkeley, October 1992.
- [8] 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.
- [9] S. Floyd. Link-Sharing and Resource Management Models for Packet Networks, September 1993. Draft.
- [10] E. M. Gafni and P. Bertsekas. Dynamic Control of Session Input Rates in Communication Networks. *IEEE Transactions on Automatic Control*, 29(11):1009–1009, November 1984.
- [11] E. L. Hahne. Round-Robin Scheduling for Max-Min Fairness in Data Networks. *IEEE Journal on Selected Areas in Communications*, 9(7):1024–1039, September 1991.
- [12] 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.
- [13] H. Heimlich. Traffic Characterization of the NSFNET National Backbone. In *Proc. 1990 Winter USENIX Conference*, January 1990.

- [14] V. Jacobsen. Congestion Avoidance and Control. In *Proc. Sigcomm '88*, pages 314–329, August 1988.
- [15] J. M. Jaffe. Bottleneck Flow Control. *IEEE Transactions on Communications*, 29(7):954–962, July 1981.
- [16] S. Jamin, S. Shenker, L. Zhang, and D. D. Clark. An Admission Control Algorithm for Predictive Real-Time Service. In *Proc. 3rd International Workshop on Network and Operating Support for Digital Audio and Video*, pages 308–315, November 1992.
- [17] S. Keshav. REAL: A Network Simulator. Technical Report 88/472, Computer Science Department, University of California, Berkeley, December 1988.
- [18] E. Monteiro, F. Boavida, and V. Freitas. A Fairness Analysis of LAN/WAN Protocol Relays. *Computer Networks and ISDN Systems*, 26:379–388, 1993.
- [19] D. Sarkar. Bandwidth regulation in internetworks. *Master's Thesis*. In Preparation. 1994.
- [20] S. Shenker, D. D. Clark, and L. Zhang. A Scheduling Service Model and a Scheduling Architecture for an Integrated Services Packet Network, 1993. Preprint.
- [21] M. Steenstrup. Fair Share for Resource Allocation, December 1992. Draft.