A New Protocol for Bandwidth Regulation of Real-Time Traffic Classes in Internetworks*

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

† Department of Computer Science, University of Virginia, Charlottesville, VA 22903
‡ School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332

Abstract

A novel bandwidth regulation mechanism is proposed which improves the ability of a packet-switching network to cope with multiple real-time and non-real-time traffic classes. The mechanism achieves regulation of link bandwidth at two levels. At the first level, bandwidth is dynamically regulated between different traffic classes. The concept of 'inter-class regulation' is introduced which enforces that the bandwidth left unused by a traffic class is divided among traffic classes with high bandwidth demands. At the second level, bandwidth regulation is enforced on end-to-end traffic streams, so-called flows, *such that flows from the same class with identical routes have the same throughput constraints. This concept is referred to as 'intra-class regulation'. A simple distributed protocol is presented that achieves intra-class and inter-class regulation in a general internetwork. The effectiveness of the protocol is demonstrated by simulation experiments.*

1 Introduction

Until recently, traffic on the Internet was dominated by applications for file transfers, electronic mail, electronic bulletin boards, and remote login. 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:

Add Bandwidth: Since excessive network delays and delay variations only occur in the presence of network congestion, a network that is equipped with sufficient network resources will be congestion-free and, thus, can support delay sensitive real-time applications. However, due to the burstiness of traffic, in particular from applications that involve the transmission of compressed video, the amount of network resources needed to avoid congestion conditions can be considerable.

Resource Reservation with Admission Control: This approach [1, 4] 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. Note that resource reservation with admission control, if implemented in the Internet, will have serious implications, since 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.

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 [5, 8, 12]. In general, resource regulation schemes do not dedicate resources to individual flows and do not provide admission control. Rather, the network enforces policies to distribute available resources to the flows. As a result, the resources available to a flow decreases if the number of flows increases.

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.

Throughout this study, we regard an internetwork as consisting of a collection of gateways that are connected by transmission links with fixed capacity, as shown in Figure 1. We distinguish between *internal gateways* and *access gateways*: internal gateways are connected exclusively to gateways, while access gateways are also linked to host systems, typically via a local area network. Hosts access the network via access gateways and each host can transmit to any other host connected to the network. Any unidirectional traffic stream between two host systems is called a *flow*.

We address the problem of regulating the use of link

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

[1] Throughout this paper, we use the term flow to denote an end-to-end, or host-to-host, packet stream. Each flow belongs to one traffic class, and the assignment of flows to traffic classes is based on the application type, the protocol used, or the location of the traffic source [11].

1063-6927/95 $4.00 © 1995 IEEE

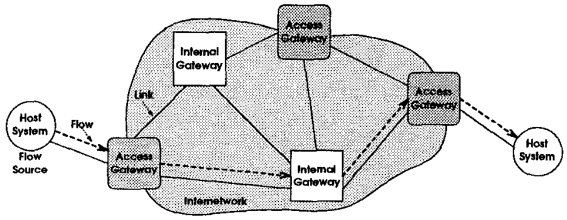

Figure 1: Internetwork.

bandwidth in an internetwork such as the one in Figure 1 without relying on admission control functions. In todays internetworks, link bandwidth is the scarcest resource. Excessive end-to-end delays, long delay variations, and packet losses mainly result from the lack of available link bandwidth. We present a novel approach for regulating the use of link bandwidth for both traffic classes and individual flows. Our objective is to implement two policies for regulating the use of link bandwidth in the network. One policy, referred to as *inter-class regulation*, regulates the bandwidth consumption of different traffic classes; the other policy, referred to as *intra-class regulation*, controls the bandwidth use of flows from the same class:

- *Inter-Class Regulation:* At each network link, traffic classes are statically assigned bandwidth guarantees. The guarantee of a class at a link is a lower bound on the total bandwidth available to all flows from this class. If the flows of a traffic class do not fully utilize the guarantee, the unused bandwidth is made available to other traffic classes. The network dynamically calculates a so-called *surplus* for a link. The surplus specifies a limit on the bandwidth that a single traffic class with high bandwidth demand can 'borrow' from other classes. Inter-class regulation does not specify how the bandwidth available to a class is distributed to the flows in this class.
- *Intra-Class Regulation:* A network with intra-class regulation enforces throughput limits for each flow at each network link, referred to as *shares*. At each link there is one share value for each traffic class. The maximum end-to-end throughput of a flow is limited by the link with the smallest share, 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 2 we illustrate the relation between flows, shown as arrows, and traffic classes, shown as pipes, for a single link. Inter-class regulation is concerned with allocating link bandwidth to the traffic classes, i.e., video, file transfer, and audio traffic classes in Figure 2. Intra-class regulation is concerned with distributing bandwidth within a single traffic class. For example, for the video traffic class, intra-class regulation determines the fraction of video-class bandwidth that is made available to a single video flow.

The problem of regulating link bandwidth in a packet-switching network has been addressed previously. One approach to bandwidth regulation is based on scheduling algorithms at the gateways [2, 3, 7]. A disadvantage of these methods is that they control usage of bandwidth exclusively by dropping packets. A different type of bandwidth control regulates the traffic rate at the flow sources [6, 8, 13]. In these studies, the objective is to ensure fairness conditions for individual flows, similar to our concept of intra-class regulation; however, regulation of bandwidth at the traffic class level is not addressed. A number of studies on *link sharing* considers bandwidth regulation of traffic classes without providing mechanisms that regulate the bandwidth consumption of flows from the same class. Link sharing approaches provide some notion of inter-class regulation, but typically do not address bandwidth regulation of flows from the same class [5, 11, 12]. So far, no regulation mechanism has been proposed that, at the same time, regulates bandwidth for individual flows *and* for traffic classes in a general network.

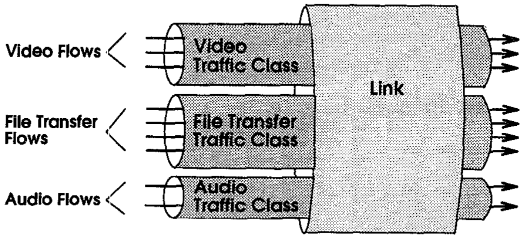

Figure 2: Flows and Traffic Classes at a Network Link.

To our knowledge, our work is the first proposal for a scheme that can regulate link bandwidth simultaneously at the traffic class and the flow level. We present a protocol that implements the policies of inter-class and intra-class regulation:in a distributed fashion. With our protocol, internal gateways need not keep state information on individual flows. 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 formally characterize a bandwidth regulation scheme with inter-class and intra-class regulation. In Section 3 we present a protocol which implements the bandwidth regulation mechanism. In Section 4 we use simulation experiments to demonstrate the effectiveness of the protocol. In Section 5 we conclude our results.

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

We consider an arbitrary network of gateways as shown in Figure 1, where hosts access the network via so-called *access gateways*. We assume that each flow, that is, a unidirectional traffic stream between two host systems, is carried over a fixed route of network gateways. The network distinguishes different *traffic classes* and may provide bandwidth guarantees for traffic 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*, denoted by λ_i for a flow i. The rate of actual data transmission is called the *throughput* of the flow, denoted by γ_i.

The network has a set $\mathcal{L}$ of unidirectional *network links* which connect internal or access gateways. The capacity of link $l \in \mathcal{L}$ is denoted by C_l and expressed in bits per second. We use $\mathcal{P}$ to denote the set of traffic classes that are recognized by the network. All traffic that does not belong to one of the classes in $\mathcal{P}$ is assigned to the *default class* '0'. So, the total set of traffic classes is given by $\mathcal{P}_0 = \mathcal{P} \cup \{0\}$. We use $\mathcal{F}$ to denote the set of end-to-end flows in the network, and $\mathcal{F}_p$ to denote the set of flows with traffic from class p ($\mathcal{F} = \bigcup_{p \in \mathcal{P}_0} \mathcal{F}_p$). The fixed route of a flow $i \in \mathcal{F}$ 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, traffic class p can obtain a *bandwidth guarantee* of G_{lp} with $\sum_{p \in \mathcal{P}} G_{lp} \leq C_l$. 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 ($G_{lp} = 0$), then flow i is assigned to default class '0' at this link. The bandwidth guarantee of class 0 at link l is given by $G_{l0} = C_l - \sum_{p \in \mathcal{P}} G_{lp}$. Let $\mathcal{P}_l$ denote the set of classes with a guarantee at link l including default class '0', that is, $\mathcal{P}_l = \{p \in \mathcal{P} \mid G_{lp} > 0\} \cup \{0\}$.

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. We refer to the *surplus*, denoted by ϕ_{lp}, as the maximum bandwidth that a class can borrow in excess of its guarantee G_{lp}. We assume that for each class-p flow i there is a throughput limit at each link on the flow's route. We refer to the throughput limits as *shares*, and denote the share of a class-p flow i at a link l on its route by $\alpha_{ip}(l)$. The share $\alpha_{ip}(l)$ 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, denoted by l_i^*, is the link on the route that has the smallest share, i.e., $\alpha_{ip}(l_i^*) = \min_{l \in \mathcal{R}_i} \alpha_{ip}(l)$. 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 and a set of flows with offered loads $\{\lambda_i \mid i \in \mathcal{F}\}$, share values $\{\alpha_{ip}(l) \mid i \in \mathcal{F}_p, l \in \mathcal{R}_i\}$, and surplus values $\{\phi_{lp} \mid l \in \mathcal{L}, p \in \mathcal{P}_l\}$. A* bandwidth allocation *maps the above parameters into throughput values $\{\gamma_i \mid i \in \mathcal{F}\}$ such that the following conditions hold:*

1. $\gamma_i \leq \min(\lambda_i, \alpha_{ip}(l_i^*))$ *for all flows* $i \in \mathcal{F}$.

2. $\displaystyle\sum_{p \in \mathcal{P}_0} \sum_{i \in \Delta_{lp}} \gamma_i \leq C_l$ *for all links* $l \in \mathcal{L}$.

3. $\displaystyle\sum_{i \in \Delta_{lp}} \gamma_i \leq G_{lp} + \phi_{lp}$ *for all traffic classes* $p \in \mathcal{P}_l$.

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

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 traffic classes to borrow unused bandwidth from other traffic 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. The proof of Lemma 1 is given in [10].

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 traffic class $q \in \mathcal{P}_l$.

Next, we discuss bandwidth allocations with intra-class regulation. For the special case of only one traffic class, the regulation policy is similar to [8, 13]. *Intra-class regulation* is concerned with distributing C_{lp}, the bandwidth available to a traffic 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 traffic class have the same bottleneck link, i.e., $l_i^* = l_j^*$, then both flows have 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 $i \in \mathcal{F}_p$*

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

We refer to the maximum values for shares, that do not leave available capacity unused if the total offered load exceeds the capacity as *maximal shares.* In Lemma 2, proven in [10], we give the condition that must hold if the shares in a network with multiple traffic 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.

The given definitions of bandwidth regulation are concerned with allocating bandwidth to flows of the same traffic class (*intra-class regulation*), and to entire traffic 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 [5, 11, 12] provide some regulation for traffic classes (different from the presented inter-class regulation), but do not solve the regulation problem for flows from the same class.

We can conclude 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 we state that such a bandwidth allocation can be effectively constructed for general networks. The proof of Theorem 1 is given in [10].

Theorem 1 *Given a network and a set of flows with offered loads $\{\lambda_i \mid i \in \mathcal{F}\}$, there exists a 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.*

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

and

$$\phi_l^* = \begin{cases} \infty & \textit{if } \bigcup\limits_{q \in \mathcal{P}_l} \mathbf{O}_{lq} = \emptyset \\ \dfrac{C_l - \sum\limits_{\mathbf{O}_{lq} \neq \emptyset} G_{lq} - \sum\limits_{\mathbf{O}_{lq} = \emptyset} \Theta_{lq}}{|\{q \in \mathcal{P}_l \mid \mathbf{O}_{lq} \neq \emptyset\}|} & \textit{otherwise} \end{cases} \tag{2}$$

subject to the side conditions:

$$G_{lp} + \phi_l^* - \Theta_{lp} \geq 0 \tag{3}$$

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

where:

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

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

$$\begin{aligned} \mathbf{U}_{lp} &= \Big\{ i \in \Delta_{lp} \mid \alpha_p^*(l) \geq \lambda_i \ , \ \ i \notin \textstyle\bigcup_{k \in \mathcal{L}} \mathbf{R}_{lp}(k) \Big\} \\ \mathbf{O}_{lp} &= \Big\{ i \in \Delta_{lp} \mid l = l_i^* \ , \ \ \alpha_p^*(l) < \lambda_i \Big\} \\ \mathbf{R}_{lp}(k) &= \Big\{ i \in \Delta_{lp} \mid k = l_i^* \ , \ \ \alpha_p^*(k) < \lambda_i \Big\} \ (k \neq l) \end{aligned} \tag{6}$$

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.

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 Theorem 1 can be developed for different bandwidth regulation definitions, in particular, for hierarchical link sharing schemes [5, 12]. Thus, Theorem 1 indicates that neglecting bandwidth control of individual flows as in the link sharing schemes will result in waste of bandwidth.

3 Protocol Issues for Bandwidth Regulation

In this section, we present a set of protocol mechanisms that enable an implementation of the mathematically developed inter-class and intra-class bandwidth regulation with maximal shares and surplus values from the previous section. The presented protocol is completely distributed, that is, no network entity is required to keep global state information.

In Section 4 we will present a simulation experiment to show that the presented protocol mechanisms can enforce fast convergence of the bandwidth regulation scheme after load changes in the network. For the sake of a clear presentation we make some simplifying assumptions for the network and the protocol. For example, we assume that information on the offered load of a flow is available at its source. Also, the protocol does not address reliability issues. In [10] we discuss how these assumptions can be relaxed.

3.1 Design Concepts

The protocol mechanisms presented here are intended as extensions to an existing network layer protocol. Even though bandwidth regulation is applicable to both connectionless and connection-oriented networks, we will assume a connectionless network which uses protocols such as IP or CLNP at the network layer.

We distinguish three protocol entities: *flow sources*, *internal gateways*, and *access gateways* (see Figure 1). A flow source is the origin of a flow and assumed to be running on a host computer system. Flow sources access the internetwork through an access gateway. Gateways, both internal and access gateways, perform switching and routing functions in the network and are interconnected via fixed-capacity links. Internal gateways are only connected to gateways, and access gateways are also connected to flow sources.

The following list summarizes the main features of the protocol for enforcing inter-class and intra-class bandwidth regulation:

- Each end-to-end flow in the network is assigned a state: the flow is *underloaded* or *overloaded* at a particular link on its route. An underloaded flow can satisfy its bandwidth demand, while an overloaded flow has a bandwidth demand that exceeds its throughput. The state of a flow is kept only at flow sources. Each flow source tags the packets of the flow with the current state.
- An internal gateway maintains, for each of its outgoing links, a set of counters which are updated every time a packet arrives at the gateway. The update operations depend exclusively on the tagging of the packet. Internal gateways do not keep state information on individual flows.
- After fixed time intervals (*update intervals*) a gateway uses its counters to calculate *share values* for each outgoing link. The share values correspond to the values $\alpha_p(l)$ from Section 2 and denote throughput limits at links. The share values are disseminated to the access gateways in control packets (*link state packets*).
- An access gateway that has received link state packets calculates from the share values the throughput limits of the flow sources connected to this access gateway. The throughput limit is forwarded to the flow sources for a reevaluation of their respective states.

In the following subsections we give a more detailed description of the protocol mechanisms.

3.2 Extensions to Packet Header

For the implementation of the bandwidth regulation scheme we require each packet to carry a limited amount of control information. The control information is carried in the header of a packet[2]. We require three additional fields in the packet header, referred to as *class field*, *bottleneck field*, and *flag*. The *class field* contains information on the traffic class of the packet. The *bottleneck field* identifies the link on the flow's route which limits the throughput of the flow, i.e., the bottleneck link. In the following we assume that links are identified by a pair '`gw:li`' where '`gw`' is a gateway in the network, and '`li`' identifies a network interface at gateway `gw`. If a flow does not have a bottleneck link, the bottleneck field is set to '`NIL`'. The *flag field* takes one of three values: '+', '–', or '.'; value '+' indicates a *plus flag*, '–' indicates a *minus flag*, and '.' to indicate that no flag is set. In the following, we will use the extended header fields to represent a packet. For example, we will write "$\boxed{p}\boxed{\texttt{gw:li}}\boxed{+}$" to denote a packet from class p with bottleneck link `gw:li` and a set *plus flag*.

3.3 Link State Packets and Rate Control at Sources

At the end of each *update interval*, an internal gateway sends, for each of its outgoing links, a *link state packet* to the access gateways of the network. (The length of the update interval should be of the same order as update periods in routing protocols.) A link state packet contains information on the maximum data that a flow can transmit on this link during the duration of an update interval. For a gateway `gw` with an outgoing link `gw:li`, the information that is sent in the link state packet consists of the tuple `<p,gw:li,Share`$_p$`>`, where `Share`$_p$ is the maximum number of bytes that any class-p flow can transmit on link `gw:li` during an

[2] In protocols such as IP, the additional fields can be accommodated in option fields of the packet header.

update interval. Below, in Subsection 3.5 we will discuss how a gateway calculates the value of $\mathtt{Share}_p$.

After receiving the link state packets, the access gateway which is connected to the source of a class-p flow, say flow i, calculates

$$\mathtt{Quota[i]} = \min\Big(\mathtt{Share}_p \mid \mathtt{<p,gw{:}li,Share}_p\mathtt{>} \text{ received and } \mathtt{gw{:}li} \text{ is on the route of flow } i\Big) \tag{7}$$

The link for which the minimum is achieved in equation (7) is the *bottleneck link* of flow i. The access gateway communicates the value of `Quota[i]` and the name of the bottleneck link to flow i's flow source. The flow source maintains a rate control mechanism which limits the data that flow i can transmit during an update interval to `Quota[i]`. We ignore the details of the rate controller and assume only that it does not permit excessive traffic bursts.

3.4 States of Flows

Each flow source has knowledge on the flow's bandwidth demands, denoted by `Load[i]` for flow i. Also, a flow source maintains information on the state of the flow. A flow is either *underloaded*, or *overloaded* at its bottleneck link. A flow source tags each data packet of the flow with information on its state.

> **Underloaded flow:** A flow is *underloaded* if $\mathtt{Load[i]} \leq \mathtt{Quota[i]}$, where `Quota[i]` is as calculated in equation (7). In each packet of an *underloaded* class-p flow, the flow source sets the header fields to $\boxed{\mathrm{p} \mid \mathtt{NIL} \mid \cdot}$.
>
> **Overloaded flow:** A flow is '*overloaded* at link `gw:li`', if $\mathtt{Load[i]} > \mathtt{Quota[i]}$ and link `gw:li` is the bottleneck link of the flow. In this case, the source of a class-p flow i sets the extended header fields of each packet to $\boxed{\mathrm{p} \mid \mathtt{gw{:}li} \mid \cdot}$.

A flow can change its state due to changes of the bandwidth demand `Load[i]` or due to changes of `Quota[i]`. If a flow changes its state, the flow source notifies all gateways on the flow's route by setting a *flag* in a packet header. Three types of state transitions can occur:

- **underloaded $\Longrightarrow$ overloaded at `gw:li`**: In this case, the flow source sends a single packet with packet header fields set to: $\boxed{\mathrm{p} \mid \mathtt{gw{:}li} \mid +}$. The *plus flag* indicates to gateway `gw` that the flow is now overloaded at link `gw:li`.
- **overloaded at `gw:li` $\Longrightarrow$ underloaded**: In this case, the flow source sends a single packet with packet header fields set to $\boxed{\mathrm{p} \mid \mathtt{gw{:}li} \mid -}$. The *minus flag* will be read by gateway `gw` and indicates that the flow is no longer overloaded at the outgoing link `gw:li`.
- **overloaded at `gw1:li1` $\Longrightarrow$ overloaded at `gw2:li2`**: This state transition occurs if the bottleneck link has moved from link `gw1:li1` to link `gw2:li2`. Then, the extended header fields of the next two packets after the state transition are set to $\boxed{\mathrm{p} \mid \mathtt{gw2{:}li2} \mid +}$ and $\boxed{\mathrm{p} \mid \mathtt{gw1{:}li1} \mid -}$. The first packet indicates to gateway `gw2` that the flow is now overloaded at the outgoing link `gw2:li2`. The second packet informs `gw1` that the flow is no longer overloaded at link `gw1:li1`.

3.5 Operations at the Gateways

The bandwidth regulation protocol requires each gateway to maintain a set of counters for each outgoing link. The counters are updated when a new packet arrives at the gateway. Next we discuss the operations performed by some gateway, say gateway `gw`, for one of its outgoing links, say link `gw:li`. Link `gw:li` has two constants `Cap` and $\mathtt{Guar}_p$ which denote the total capacity of the link and the capacity guaranteed to class p, respectively. Both `Cap` and $\mathtt{Guar}_p$ are measured in bytes transmitted per update interval.

For each outgoing link a gateway maintains two counters $\mathtt{Rate}_p$ and $\mathtt{OL}_p$ for each traffic class p. $\mathtt{Rate}_p$ is used to count the number of bytes transmitted on link `gw:li` from all flows that are either underloaded or overloaded at some link `gw1:li1` with $\mathtt{gw1{:}li1} \neq \mathtt{gw{:}li}$.

$\mathtt{OL}_p$ counts the number of flows that are overloaded at link `gw:li`. $\mathtt{OL}_p$ is updated only if a packet arrives that has either a *plus flag* or a *minus flag* set. More precisely, if a packet arrives with header fields set to $\boxed{\mathrm{p} \mid \mathtt{gw{:}li} \mid +}$ then $\mathtt{OL}_p$ is incremented by one. If a packet arrives where the header fields are set to $\boxed{\mathrm{p} \mid \mathtt{gw{:}li} \mid -}$ then $\mathtt{OL}_p$ is decremented by one.

At the end of an update interval, a gateway calculates for each of its outgoing links and for each traffic class p a share value $\mathtt{Share}_p$ and a surplus value $\mathtt{Surplus}_p$. The calculations are based on Theorem 1 from Section 2 and involve the following computations:

$$\mathtt{Share}_p = \begin{cases} \mathtt{infinity} & \text{if } \mathtt{OL}_p = 0 \\ \dfrac{\mathtt{Guar}_p + \mathtt{Surplus}_p - \mathtt{Rate}_p}{\mathtt{OL}_p} & \text{otherwise} \end{cases} \tag{8}$$

and

$$\mathtt{Surplus}_p = \begin{cases} \mathtt{infinity} & \text{if } \mathtt{OL}_p = 0 \text{ for all } p \\ \dfrac{\mathtt{Cap} - \sum\limits_{\mathtt{OL}_q>0} \mathtt{Guar}_q - \sum\limits_{\mathtt{OL}_q=0} \mathtt{Rate}_q}{|\{q \mid \mathtt{OL}_p > 0\}|} & \text{otherwise} \end{cases} \tag{9}$$

In equations (8) and (9), `infinity` is chosen such that $\mathtt{infinity} \gg \mathtt{Cap}$. Note that both equations can be computed for all traffic classes without information on the share or surplus values from other gateways.

As soon as the values for $\mathtt{Share}_p$ and $\mathtt{Surplus}_p$ are calculated for link `gw:li`, gateway `gw` creates a link state packet with content $\mathtt{<p,gw{:}li,Share}_p\mathtt{>}$ and sends the packet to all access gateways. Then, the gateway resets counter $\mathtt{Rate}_p$ to zero.

REMARK: By neglecting for a moment that Theorem 1 is expressed in terms of data rates, we obtain

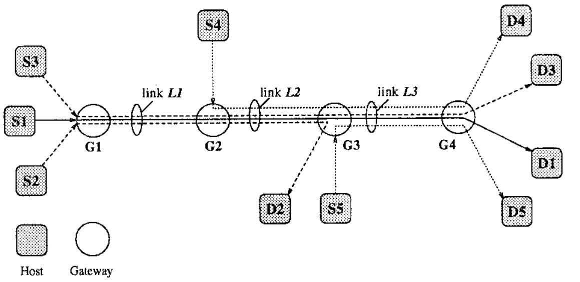

Figure 3: Simulated Network.

the following relation between equations (8) – (9) and Theorem 1:

$$\begin{array}{lcllcl} \texttt{Share}_p & \equiv & \alpha_p^*(l) & \texttt{Surplus}_p & \equiv & \phi_l^* \\ \texttt{Cap} & \equiv & C_l & \texttt{Guar}_p & \equiv & G_{lp} \\ \texttt{OL}_p & \equiv & \lvert O_{lp} \rvert & \texttt{Rate}_p & \equiv & \Theta_{lp}. \end{array}$$

4 Simulation Experiment

To provide insight into the dynamics of the bandwidth regulation protocol outlined in Section 3 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 [9]. 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 $10\mu s$. Each flow source 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 traffic classes: *0*, *I*, and *II*. The bandwidth guarantees of the traffic classes are identical at all links, and denoted by G_0, G_I, and G_{II}. The guarantees are set to $G_0 = 15$ Mb/s for class *0*, $G_I = 30$ Mb/s for class *I*, and $G_{II} = 55$ Mb/s for class *II*.

The parameters of the five flows in Figure 3, that is, source host, destination host, route, traffic 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 transmits 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.

From	To	Route	Class	Offered Load	Start Time
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 1: Flow Parameters.

- 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 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.

- 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 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*.

- 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 that the protocol requires a few iterations before settling at the correct bandwidth allocation.

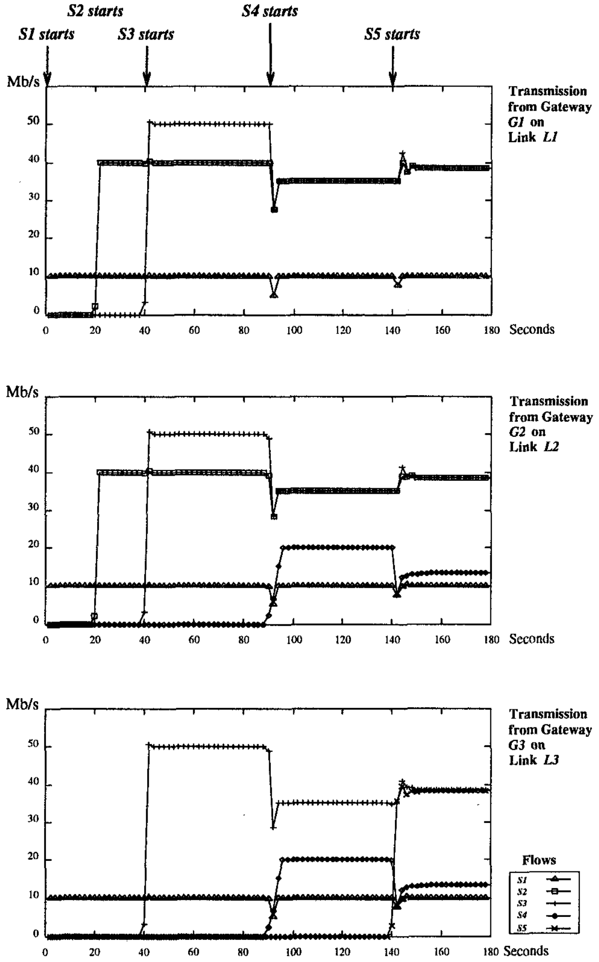

Figure 4: Simulation Results.

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 classes, share link bandwidth without considering the number of end-to-end traffic flows in each class. Intra-class regulation enforces rules for dividing link bandwidth among flows from the same traffic class. We have presented a distributed protocol that enforces inter-class and intra-class regulation of bandwidth in a general network. We have presented a simulation experiment and showed that the protocol quickly adapts to changes in the network load.

References

[1] 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.

[2] 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.

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

[4] 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.

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

[6] 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.

[7] 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.

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

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

[10] J. Liebeherr, I.F. Akyildiz, and D. Sarkar. Bandwidth Regulation of Real-Time Traffic Classes in Internetworks. Technical Report CS-94-24, University of Virginia, Department of Computer Science, June 1994.

[11] S. Shenker, D. D. Clark, and L. Zhang. A Scheduling Service Model and a Scheduling Architecture for an Integrated Services Packet Network. Technical report, Xerox PARC, Palo Alto, California, ftp://ftp.parc.xerox.com/pub/net-research/archfin.ps, August 1993.

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

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