ELSEVIER

Computer Networks and ISDN Systems 28 (1996) 855-872

COMPUTER NETWORKS and ISDN SYSTEMS

Bandwidth regulation of real-time traffic classes in internetworks

Ian F. Akyildiz[a], Jörg Liebeherr[b,*], Debapriya Sarkar[c]

[a] *Broadband and Wireless Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*

[b] *Computer Science Department, University of Virginia, Charlottesville, VA 22903, USA*

[c] *Hughes Network Systems, Germantown, MD 20876, USA*

Accepted 4 May 1995

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.

Keywords: Bandwidth regulation; Fairness; Flow control; Real-time traffic; Packet switching; Internetworks

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.

[*] Corresponding author. E-mail: jorg@virginia.edu.

SSDI 0169-7552(95)00063-1

There is an ongoing discussion whether traditional packet-switching networks, 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. 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 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. Resource regulation can be enforced at the level of *traffic classes* or at the level of individual flows. In class-level regulation schemes, the network reserves resources for a particular traffic class, and may permit other classes to utilize resources that are left unused. A class-level regulation scheme is not concerned with the distribution of the bandwidth reserved for a class to the flows in the class. On the other hand, a flow-level regulation scheme limits the maximum amount of resources that can be used by a single flow. Ideally a resource regulation mechanism simultaneously controls the consumption of resources at both the level of traffic classes and the level of flows. Until now, such a resource control 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.

Throughout this study, we regard an internetwork as consisting of a collection of gateways that are con-

[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 [20].

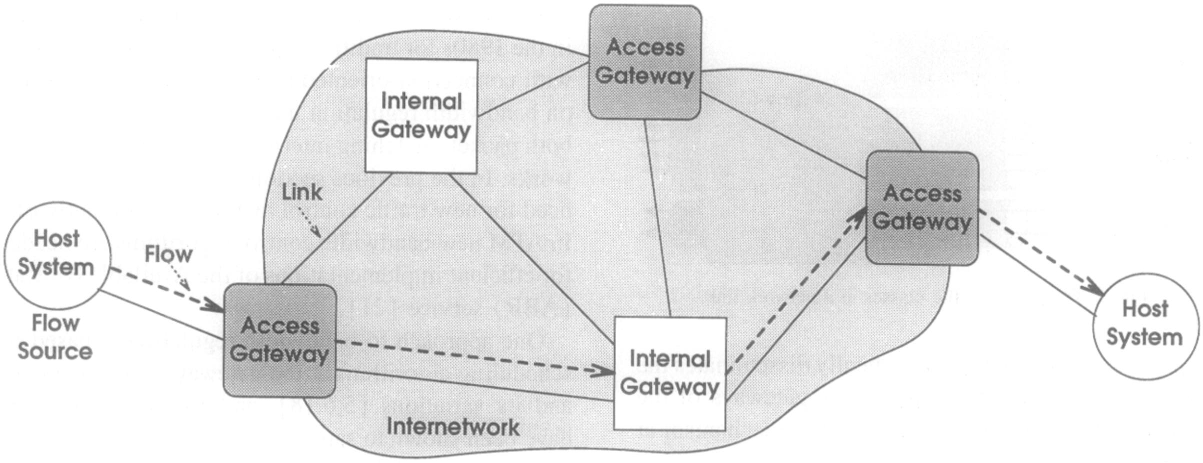


Fig. 1. Internetwork.

nected by transmission links with fixed capacity, as shown in Fig. 1. We distinguish 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 so-called *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 bandwidth in an internetwork such as the one in Fig. 1 without relying on admission control functions. In todays internetworks, link bandwidth is the scarcest resource. Excessive end-to-end delays, and long delay variations, and packet losses, mainly result from the lack of available link bandwidth. We present a novel approach for regulating 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 consumptions between 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 class at each network link, referred to as *shares*. 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 Fig. 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 Fig. 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.

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 distributed protocol that implements the above regulation policies. The overhead of the protocol consists

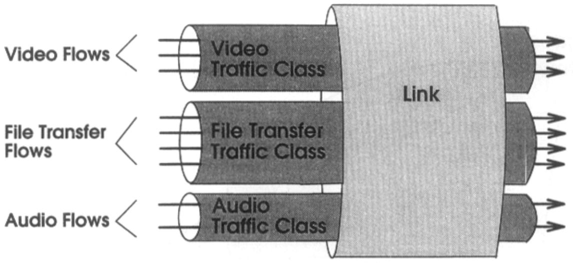


Fig. 2. Flows and traffic classes at a network link.

of a control protocol that periodically disseminates the values of the counters to the access gateways of the network. We also require a rate control mechanism at the flow 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 solution to a bandwidth regulation scheme that implements the above-mentioned 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. In Section 5 we use simulation experiments to demonstrate the effectiveness of the protocol. In Section 6 we discuss extensions of the presented protocol, and in Section 7 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 either to reach some notion of fairness between flows within a single traffic class, or to control link bandwidth allocation to traffic 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 traffic 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 both packet-switching internetworks and B-ISDN networks. In the previous section, we have discussed the need for new traffic control methods in internetworks. In ATM new bandwidth control algorithms are needed for efficient implementations of the Available Bit Rate (ABR) service [21].

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 been shown to satisfy certain fairness criteria for either individual flows or traffic 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 [10,15,23]. 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 traffic class level is not addressed. Other disadvantages are that control algorithms either require global knowledge of the network state [23], or require flow sources to communicate their bandwidth needs to each other [15].

A number of studies considers bandwidth regulation of traffic 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 [22] proposes a hierarchical structure of traffic 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 traffic 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 [9,22], or admission control functions are employed [20].

3. Bandwidth allocations with intra-class and inter-class regulation

We consider an arbitrary network of gateways as shown in Fig. 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.

We consider a network with 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 given 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 (see Fig. 1), and $\mathcal{F}_p$ is the set of flows with traffic from traffic 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 $1 \le k \le K$. 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 *bandwidth guarantee* of G_{lp} with $\sum_{p\in\mathcal{P}} G_{lp} \le C_l$. 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 ($G_p = 0$), 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 use 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) $\gamma_i \le \min\big(\lambda_i, \alpha_{ip}(l_i^*)\big)$ for all flows $i \in \mathcal{F}$.

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

(3) $\displaystyle\sum_{i\in\Delta_{lp}} \gamma_i \le G_{lp} + \phi_{lp}$ 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 $\sum_{j \in \Delta_{lp}} \gamma_j \le C_{lp}$.

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 $p \in \mathcal{P}_l$

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

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 3 we state that by selecting ϕ_l as large as possible, one can make the entire link bandwidth available for transmission.

Lemma 3. *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 $q \in \mathcal{P}_l$.

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

Next, we discuss bandwidth allocations with intra-class regulation. For the special case of only one traffic class, the regulation policy is similar to [15,21,23]. *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 4. 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 $i \in \mathcal{F}_p$

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

As an example of intra-class regulation, consider the network in Fig. 3 with two links, denoted by "a" and "b", and one traffic 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:

$\lambda_1 = 2$ Mb/s, $\lambda_4 = 4$ Mb/s,
$\lambda_2 = 6$ Mb/s $\lambda_5 = 2$ Mb/s.
$\lambda_3 = 6$ Mb/s,

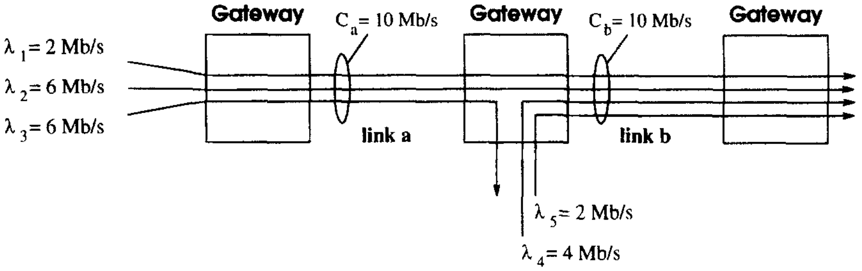


Fig. 3. Intra-class fairness in a network with two links.

By 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 4:

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

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 $\alpha_b < 3$ Mb/s, or will violate the constraints for a bandwidth regulation with intra-class regulation, e.g., if $\alpha_b > 3$ Mb/s. We refer to the maximum values for shares, that do not leave capacity available to a traffic class unused if the total offered load exceeds the capacity as *maximal shares*. In Lemma 5 we give the condition that must hold if the shares in a network with multiple traffic classes are maximal.

Lemma 5. *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. Intra-class regulation with maximal shares is referred to in the literature as max-min fairness [21].

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

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 [9,20,22] 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 3 that a bandwidth allocation with intra-class regulation but without maximal shares can result in a waste of available bandwidth. Likewise, Lemma 5 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 2, 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 6. *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* [2]*:*

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

and

$$\phi_l^* = \begin{cases} \infty & \text{if } \bigcup_{q \in \mathcal{P}_l} \boldsymbol{O}_{lq} = \emptyset, \\ \dfrac{C_l - \sum_{\boldsymbol{O}_{lq} \neq \emptyset} G_{lq} - \sum_{\boldsymbol{O}_{lq} = \emptyset} \Theta_{lq}}{|\{q \in \mathcal{P}_l \mid \boldsymbol{O}_{lq} \neq \emptyset\}|} & \text{otherwise} \end{cases} \tag{2}$$

subject to the side conditions

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

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

where

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

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

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

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

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

Note that each class-p flow i with link l on its route belongs to one of the sets $\boldsymbol{U}_{lp}$, $\boldsymbol{O}_{lp}$, or $\boldsymbol{R}_{lp}(k)$ ($k \in \mathcal{R}_i$). $\boldsymbol{U}_{lp}$ 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. $\boldsymbol{O}_{lp}$ and $\boldsymbol{R}_{lp}(k)$ contain flows i with $\gamma_i < \lambda_i$, that is, the bandwidth demand of the flow is greater than its throughput. $\boldsymbol{O}_{lp}$, the set of *overloaded* class-p flows on link l, contains flows which have link l as the bottleneck. $\boldsymbol{R}_{lp}(k)$, 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 presented in [19]. We will only discuss the main steps of the proof here.

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 traffic classes and the number of links. 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). □

An important implication of Theorem 2 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 Eq. (1), or maximal surplus values ϕ_l^* as in Eq. (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 2 can be developed for different bandwidth regulation definitions, in particular, for hierarchical link sharing schemes [9,22]. Thus, Theorem 2 indicates that neglecting bandwidth control of individual flows as in the link sharing schemes

[2] In Eqs. (1) and (2), $|X|$ denotes the cardinality of a set X.

will result in waste of bandwidth.

In the next section, we use Theorem 2 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. 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 implementation is completely distributed, that is, no network entity is required to keep global state information.

In Section 5 we will present simulation experiments 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. After the presentation of the protocol and the simulation experiments, we will discuss in Section 6 how these assumptions can be relaxed.

4.1. 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 Fig. 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 each time a packet arrives to 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 3 and denotes throughput limits on a link. 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 which use the information to reevaluate their state.

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

4.2. Extensions to packet header

For the implementation of the bandwidth regulation scheme we require each packet, that is, each network layer protocol data unit, to carry a limited amount of control information. The control information is carried in the header of a packet. 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*.

4.3. Link state packets and rate control at sources

Periodically, 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 $\texttt{<p,gw:li,Share}_p\texttt{>}$, where $\texttt{Share}_p$ is the maximum number of bytes that any class-p flow can transmit on link gw:li during an update interval. Below, in Subsection 4.5 we will discuss how a gateway calculates the value of $\texttt{Share}_p$.

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

$$\texttt{Quota[i]} = \min\Big(\texttt{Share}_p \mid \texttt{<p,gw:li,Share}_p\texttt{>} \text{ received and } \texttt{gw:li} \text{ is on the route of flow } i\Big). \qquad (9)$$

The link for which the minimum is achieved in Eq. (9) 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.

The above steps are summarized in Fig. 4 which depicts part of the route of a single flow as a dashed line. The route begins at the flow source shown on the left of the figure, and passes through link gw1:li1 of access gateway gw1, link gw2:li2 of access gateway gw2, and link gw3:li3 of access gateway gw3. At the end of an update interval, gateways gw2 and gw3 send link state packets to access gateway gw1. Then access gateway gw1 calculates the bottleneck link, say link gw2:li2, and the throughput limit Quota[i]. The information is communicated to the flow source which in turn enforces the throughput limit by a rate control mechanism.

4.4. 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 where the bottleneck link. A flow source tags each data packet of the flow with information on its state.

- *Underloaded flow:* A flow is *underloaded* if Load[i] $\leq$ Quota[i], where Quota[i] is as calculated in Eq. (9). In each packet of an *underloaded* class-p flow, the flow source sets the header fields to $\boxed{p}\boxed{\texttt{NIL}}\boxed{\cdot}$.
- *Overloaded flow:* A flow is "*overloaded* at link gw:li", if Load[i] $>$ 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 $\boxed{p}\boxed{\texttt{gw:li}}\boxed{\cdot}$.

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: $\boxed{p}\boxed{\texttt{gw:li}}\boxed{+}$. The *plus flag* indicates to gateway gw that the flow is

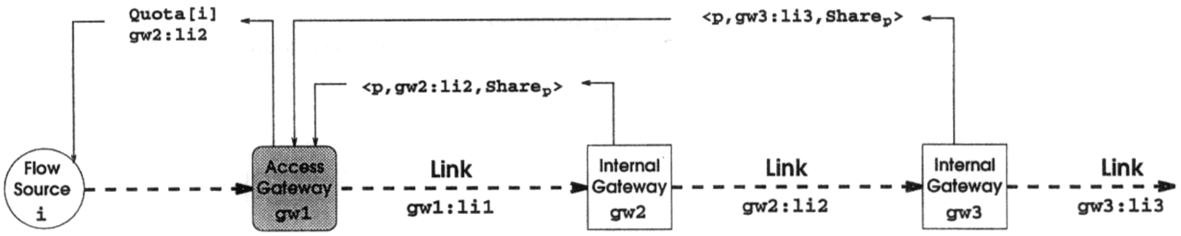


Fig. 4. Transmission and processing of link state packets.

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 $\boxed{p}\boxed{\texttt{gw:li}}\boxed{-}$. 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 $\boxed{p}\boxed{\texttt{gw2:li2}}\boxed{+}$ and $\boxed{p}\boxed{\texttt{gw1:li1}}\boxed{-}$. 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.

4.5. 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 $\texttt{Guar}_p$ which denote the total capacity of the link and the capacity guaranteed to class p, respectively. Both Cap and $\texttt{Guar}_p$ are measured in bytes transmitted per update interval.

For each outgoing link a gateway maintains two counters $\texttt{Rate}_p$ and $\texttt{OL}_p$ for each traffic class p. $\texttt{Rate}_p$ 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, say gw1:li1 with gw1:li1 ≠ gw:li.

$\texttt{OL}_p$ counts the number of flows that are overloaded at link gw:li. $\texttt{OL}_p$ 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 $\boxed{p}\boxed{\texttt{gw:li}}\boxed{+}$, then $\texttt{OL}_p$ is incremented by one. If a packet arrives where the header fields are set to $\boxed{p}\boxed{\texttt{gw:li}}\boxed{-}$, then $\texttt{OL}_p$ is decremented by one.

The update operations for the counters are summarized in Fig. 5. The figure shows the operations performed at gateway gw for a packet that will be routed on outgoing link gw:li. In Figs. 5(a)–(c) we show the operations of $\texttt{Rate}_p$ upon arrival of a packet (with length of Size bytes) that has neither the plus flag nor the minus flag set. Note that no update operation is performed for link gw:li if the header of the arriving packet is set to $\boxed{p}\boxed{\texttt{gw:li}}\boxed{\cdot}$. In Figs. 5(d)–(e) we show the update operations performed upon arrival of a packet with a matching bottleneck field that has a plus or minus flag set.

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 $\texttt{Share}_p$ and a surplus value $\texttt{Surplus}_p$. The calculations are based on Theorem 2 from Section 3 and involve the following computations:

$$\texttt{Share}_p = \\ = \begin{cases} \texttt{infinity} & \text{if } \texttt{OL}_p = 0, \\ \dfrac{\texttt{Guar}_p + \texttt{Surplus}_p - \texttt{Rate}_p}{\texttt{OL}_p} & \text{otherwise} \end{cases} \tag{10}$$

and

$$\texttt{Surplus}_p =$$

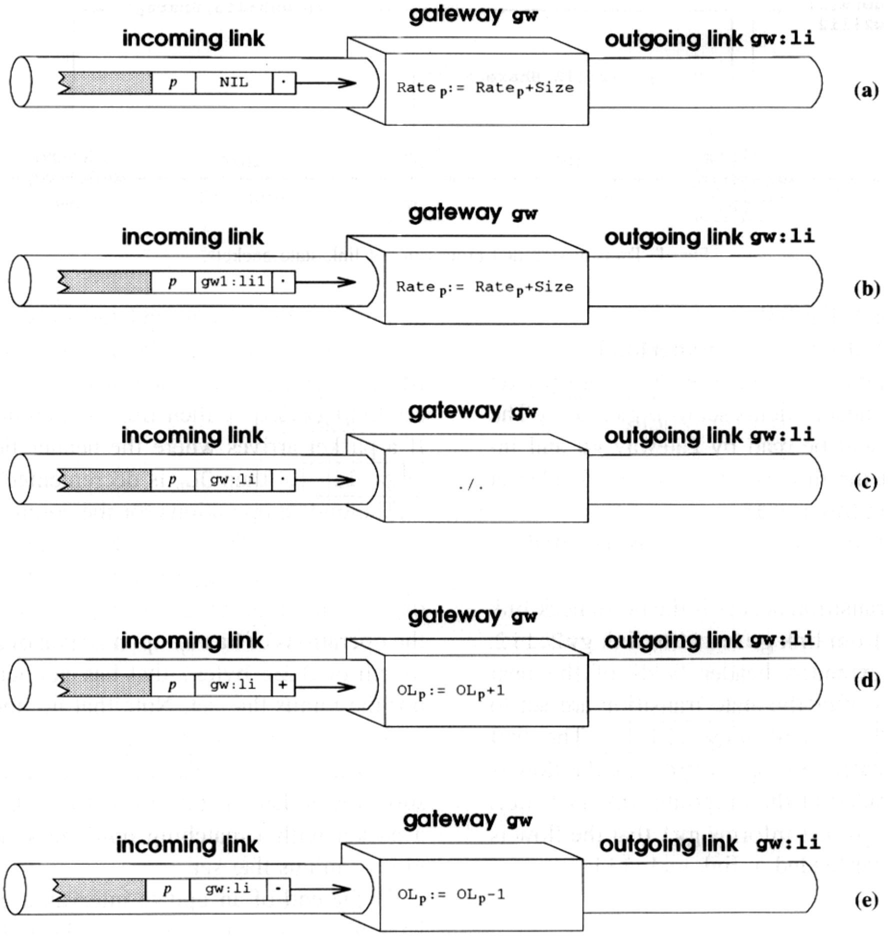


Fig. 5. Update operations performed at gateway gw for link gw:li.

$$= \begin{cases} \mathtt{infinity} & \text{if } \mathtt{OL}_p = 0 \text{ for all } p, \\ \dfrac{\mathtt{Cap} - \sum\limits_{\mathtt{OL}_q>0} \mathtt{Guar}_q - \sum\limits_{\mathtt{OL}_q=0} \mathtt{Rate}_q}{|\{q \mid \mathtt{OL}_p > 0\}|} & \text{otherwise.} \end{cases} \tag{11}$$

In Eqs. (10) and (11), infinity is chosen such that infinity $\gg$ 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 sends a link state packet with content <p,gw:li,$\mathtt{Share}_p$> 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 2 is expressed in terms of data rates, we obtain the following relation between Eqs. (10)–(11) and Theorem 2:

$$\mathtt{Share}_p \equiv \alpha_p^*(l),$$

$$\mathtt{Surplus}_p \equiv \phi_l^*,$$

$$\mathtt{Cap} \equiv C_l,$$

$$\mathtt{Guar}_p \equiv G_{lp},$$

$$\mathtt{OL}_p \equiv |O_{lp}|,$$

$$\mathtt{Rate}_p \equiv \Theta_{lp}.$$

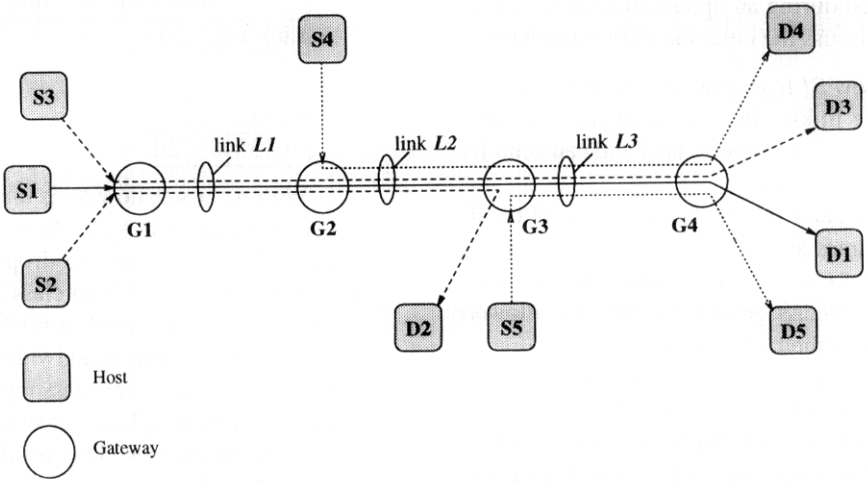


Fig. 6. Simulated network.

5. Simulation experiments

To provide insight into the dynamics of the bandwidth regulation protocol outlined in Section 4 we present simulation experiments 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 μs. 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 Fig. 6, 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

Table 1
Flow parameters

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

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

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

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

The parameters of the five flows in Fig. 6, 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 can transmit on a link during an update interval. The simulation results are summarized in Fig. 7. The figure depicts three graphs which show, separate for each link, the bandwidth (in Mb/s) utilized by each flow. From top to bottom, the graphs show the transmissions by gateway *G1* on link *L1*, by gateway *G2* on link *L2*, and by gateway *G3* on link *L3*. Each data point in the graph corresponds to the amount of data that is transmitted during an update interval of 2 seconds. Next we discuss the outcome of the simulation.

- At $t = 0$, flow *S1* from class *0* starts transmission on all three links. Since no other flow is transmitting, flow *S1* is underloaded and can send its entire load of 10 Mb/s.
- At $t = 20$, class-*II* flow *S2* with a load of 40 Mb/s becomes active on links *L1* and *L2*. Since both flows *S1* and *S2* are underloaded with respect to their class guarantees, they are allowed to transmit at their offered loads.
- At $t = 40$, another class-*II* flow, *S3*, starts to transmit over links *L1*, *L2*, and *L3*, with an offered load of 70 Mb/s. With *S3*, class *II* requires more bandwidth on link *L1* than it is guaranteed. As it is the only such class, inter-class regulation permits class *II* to borrow from the bandwidth guarantees made to other classes. Thus, class *II* obtains 90 Mb/s bandwidth for 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 follows[3]:

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

γ_{S1}	γ_{S2}	γ_{S3}
10	40	50

In Fig. 7 it can be seen that the protocol quickly settles at the predicted values.
- At $t = 90$, flow *S4* from class *0* starts transmission on links *L2* and *L3* with an offered load of 70 Mb/s. Then, both classes *0* and *II* require their respective bandwidth guarantees on link *L2*. Since there is no class-I traffic on link *L2*, inter-class regulation permits the bandwidth guarantee to class *I* to be split between classes *0* and *II*. After $t = 90$, the expected share and surplus values for link *L2*, and the throughputs of flows with traffic on link *L2* are as follows:

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

γ_{S1}	γ_{S2}	γ_{S3}	γ_{S4}
10	35	35	20

Within class *0*, flow *S1* is underloaded and *S4* is overloaded at link *L2*. Note in Fig. 7 that the throughputs of *S2* and *S3* drop to 35 Mb/s.
- At $t = 140$, flow *S5* from class *I* becomes active on link *L3* with a load of 60 Mb/s. Since flow *S5* requires its entire bandwidth guarantee of 30 Mb/s at link *L3*, inter-class regulation forces all other classes to reduce transmissions to their respective guarantees. This results in an interesting shift of bottleneck links. The reduced bandwidth at link *L3* decreases the throughput available to *S4* (from class *0*), and causes a shift of flow *S4*'s bottleneck from link *L2* to *L3*. This in turn, makes bandwidth available for class-*II* flows on link *L2*, yielding a throughput increase for flows *S2* and *S3*. However, since flow *S2* is still restricted at its bottleneck link *L2*, it cannot fully utilize its bandwidth guarantee at link *L3*. Hence, flow *S4* from class *0* and flow *S5* from class *I* can borrow the unused class-*II* guarantee on link *L3*. Note from Fig. 7 that the protocol requires a few iterations before settling at the correct bandwidth allocation. Eventually, the following theoretically expected values are obtained:

	$\alpha_0(L2)$	$\alpha_I(L2)$	$\alpha_{II}(L2)$	ϕ_{L2}
Link *L2*	-	-	38.3	21.7

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

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

6. Discussion

When discussing the implementation issues for a protocol that provides inter-class and intra-class regulation, we made a number of assumptions which must be addressed in any "real-world" implementa-

[3] The data in the tables is given in Mb/s. For clarity, we substituted the symbol "∞" by "-".

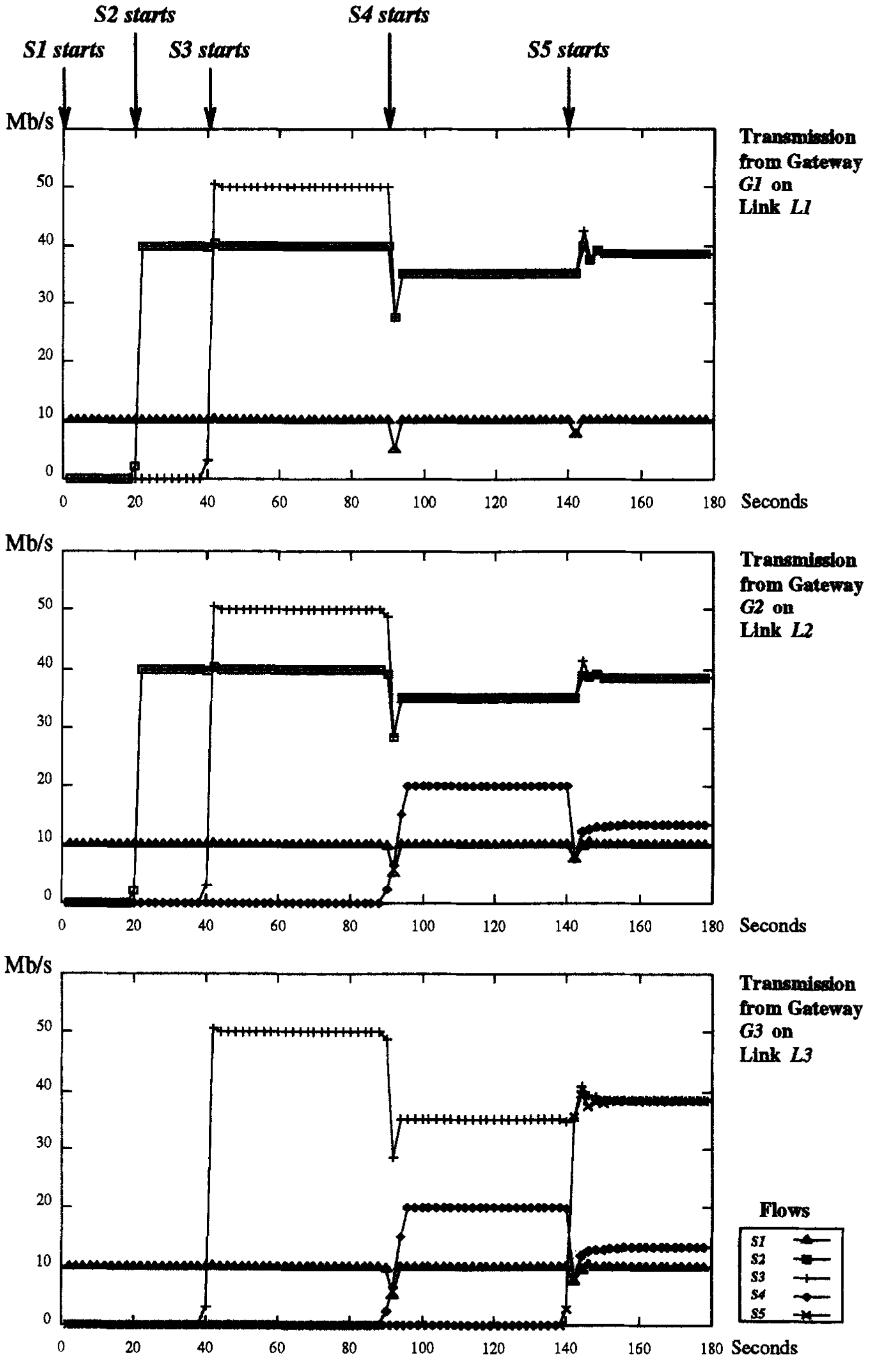


Fig. 7. Simulation results.

tion. 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 with the *plus flag* or *minus flag* set. However, 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 *bottleneck 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 [22], 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.

7. 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 streams, so-called *flows*, in each class. Intra-class regulation enforces rules for dividing link bandwidth to flows from the same traffic class. We have developed a theoretical framework for bandwidth regulation schemes in a general network, and formally showed the existence of a 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.

Acknowledgements

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

References

[1] R. Caceres, P.B. Danzig, S. Jamin, and D.J. Mitzel, Characteristics of wide-area TCP/IP conversations, *Proc. ACM Sigcomm '91*, September 1991, pp. 101–112.

[2] CCITT, CCITT Draft Recommendation I.cls: Support of Broadband Connectionless Data Service on B-ISDN, December 1991.

[3] D.D. Clark, The design philosophy of the DARPA Internet protocols, *Proc. ACM Sigcomm '88*, August 1988, pp. 106–114.

[4] D.D. Clark, S. Shenker, and L. Zhang, Supporting real-time applications in an integrated services packet network: architecture and mechanisms, *Proc. ACM Sigcomm '92*, August 1992, pp. 14-26.

[5] J.R. Davin and A.T Heybey, A simulation study of fair queueing and policy enforcement, *Comput. Comm. Rev.* **20** (5) (1990) 23-39.

[6] A. Demers, S. Keshav, and S. Shenker, Analysis and simulation of a fair queueing algorithm, *Proc. ACM Sigcomm '89*, 1989, pp. 1-12.

[7] D. Ferrari, A. Banerjea, and H. Zhang, Network support for multimedia – a discussion of the Tenet approach, *Comput. Networks ISDN Systems* **26** (10) (1994) 1167–1180.

[8] D. Ferrari and D.C. Verma, A scheme for real-time channel establishment in wide-area networks, *IEEE J. Select. Areas Comm.* **8** (3) (1990) 368–379.

[9] S. Floyd and V. Jacobson, Link-sharing and resource management, *IEEE/ACM Trans. Networking* **3** (4) (1995) 365–386.

[10] E.M. Gafni and P. Bertsekas, Dynamic control of session input rates in communication networks, *IEEE Trans. Automat. Control* **29** (11) (1984) 1009–1009.

[11] E.L. Hahne, Round-robin scheduling for max-min fairness in data networks, *IEEE J. Select. Areas Comm.* **9** (7) (1991) 1024–1039.

[12] E.L. Hahne, C.R. Kalmanek, and S.P. Morgan, Flow control on a high-speed wide-area data network, *Comput. Networks ISDN Systems* **26** (1) (1993) 29–43.

[13] H. Heimlich, Traffic characterization of the NSFNET national backbone, *Proc. 1990 Winter USENIX Conf.*, January 1990.

[14] V. Jacobsen, Congestion avoidance and control, *Proc. ACM Sigcomm '88*, August 1988, pp. 314–329.

[15] J.M. Jaffe, Bottleneck flow control, *IEEE Trans. Comm.* **29** (7) (1981) 954–962.

[16] S. Jamin, S. Shenker, L. Zhang, and D.D. Clark, An admission control algorithm for predictive real-time service, *Proc. 3rd Int. Workshop on Network and Operating Support for Digital Audio and Video*, November 1992, pp. 308–315.

[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, *Comput. Networks ISDN Systems* **26** (1993) 379–388.

[19] D. Sarkar, Distributed bandwidth regulation in mechanisms for multiple traffic classes in wide area networks, Master's Thesis, Computer Science Department, University of Virginia, 1995.

[20] 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, Calif., August 1993. ftp://parcftp.xerox.com/transient/service-model.ps

[21] S. Sathaye, ATM Forum Traffic Management Specification Version 4.0, ATM Forum Contribution 95-0013R8, October 1995.

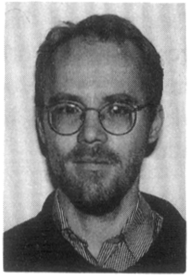
[22] M. Steenstrup, Fair share for resource allocation, Technical Report, BBN Systems and Technologies (Unpublished Memorandum), December 1992. ftp://clynn.bbn.com/pub/docs/FairShare.draft9312.ps

[23] M. Zukerman and S. Chan, Fairness in ATM networks, *Comput. Networks ISDN Systems* **26** (1) (1993) 109–117.

Ian F. Akyildiz received his B.S., M.S., and Ph.D. degrees in Computer Engineering from the University of Erlangen-Nuernberg, Germany, in 1978, 1981 and 1984, respectively. Currently, he is a Full Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology. He has held visiting professorships at the Universidad Tecnica Federico Santa Maria, Chile, Université Pierre et Marie Curie (Paris VI) and Ecole Nationale Supérieure Télécommunications in Paris, France. He has published over hundredfifty technical papers in journals and conference proceedings. He is a co-author of a textbook entitled "Analysis of Computer Systems" published by Teubner Verlag in Germany in 1982. He is an editor for "Computer Networks and ISDN Systems" , for "IEEE Transactions on Computers", and for "ACM-Baltzer Journal of Wireless Networks". He guest-edited several special issues, such as on "Parallel and Distributed Simulation Performance" for "ACM Transactions on Modeling and Simulation"; on "Teletraffic Issues in ATM Networks" for "Computer Networks and ISDN Systems Journal", and on "Networks in the Metropolitan Area" for "IEEE Journal of Selected Areas in Communications".

Dr. Akyildiz is a senior member of IEEE since 1989, member of ACM (Sigcomm) and is a National Lecturer for ACM since 1989. He received the "Don Federico Santa Maria Medal" for his services to the Universidad of Federico Santa Maria in Chile. Dr. Akyildiz is listed on "Who's Who in the World (Platinum Edition)". He received the ACM Best Distinguished Lecturer Award for 1994.

His current research interests are in ATM and wireless networks.



Jörg Liebeherr received the Diplom-Informatiker degree from the University of Erlangen-Nürnberg, Germany, in 1988 and the Ph.D. degree in Computer Science from the Georgia Institute of Technology in 1991. From 1990 until 1991 he did his dissertation work partly at the IBM T.J. Watson Research Center. In 1992 he was a Postdoctoral Fellow in the Computer Science Division of the University of California, Berkeley. Since September 1992 he is an Assistant Professor in the Department of Computer Science at the University of Virginia.

His research interests are broadband networks, multimedia networks, real-time systems and performance evaluation. He is a member of IEEE and ACM (Sigcomm and Sigmetrics).

Debapriya Sarkar received the B.Tech. degree in Computer Science and Engineering from Indian Institute of Technology, Kharagpur, India, in 1993 and the M.S. degree in Computer Science from Department of Computer Science at the University of Virginia in July 1995. He is currently with Hughes Network Systems, Germantown, Md. His research interests are in traffic control of wide-area networks and ATM networks. He is a member of IEEE and ACM (Sigcomm).