A QoS Architecture for Quantitative Service Differentiation

Nicolas Christin and Jörg Liebeherr, University of Virginia

ABSTRACT

For the past decade, a lot of Internet research has been devoted to providing different levels of service to applications. Initial proposals for service differentiation provided strong service guarantees, with strict *per-flow* bounds on delays, loss rates, and throughput, but required high overhead in terms of computational complexity and memory, both of which raise scalability concerns. Recently, the interest has shifted to *class-based* service architectures with low overhead. However, these newer service architectures only provide weak service guarantees, which do not always address the needs of applications. In this article we introduce a service architecture that supports strong per-class service guarantees, can be implemented with low computational complexity, and only requires maintainance of a little state information. A key mechanism of the proposed service architecture is that service rate allocation to classes is adaptive, and combined with buffer management. Furthermore, instead of using admission control or traffic policing, the proposed architecture exploits explicit congestion notification for the purpose of regulating the traffic entering the network.

INTRODUCTION

Since its creation in the early 1970s, the Internet has adopted a best effort service, which relies on the following three principles:

- No traffic is denied admission to the network.
- All traffic is treated in the same manner.
- The only guarantee given by the network is that traffic will be transmitted in the best possible way given the available resources; that is, no artificial delays will be generated, and no unnecessary losses will occur.

Best effort service is adequate as long as the applications using the network are not sensitive to variations in losses and delays (e.g., electronic mail), and if the load on the network is small. These conditions were true in the early days of the Internet, but do not hold anymore due to the increasing number of different applications using the Internet.

The solutions for providing different levels of services in the network are summarized as *quality of service* (QoS). Several have argued that increasing the capacity of the backbone network makes QoS obsolete. Indeed, the core of the Internet is currently overprovisioned, and supports low latency and low loss service for all its traffic. On the other hand, increasing the capacity of the Internet backbone has merely shifted the capacity bottleneck to the edge of the backbone networks, and the service experienced by demanding applications remains inadequate. As a result, mechanisms for service differentiation are urgently needed in the access networks that connect end users to the Internet backbone.

In fact, the explosion of link capacity in the network, instead of alleviating the need for service guarantees, has put more stringent requirements on QoS architectures. Routers at the edges of the Internet backbone now have to serve millions of concurrent flows at gigabit per second rates, which induces scalability requirements. First, the state information kept in the routers for providing QoS must be small. Second, the processing time for classifying and scheduling packets according to their QoS guarantees must be small as well, even with the advent of faster hardware.

A number of service architectures for packet networks that have tried to provide a solution to the service differentiation problem have been devised in the last decade. Building on the initial work of the Tenet group at the University of California at Berkeley [1], and the work by Clark, Shenker, and Zhang [2], the Internet Engineering Task Force (IETF) proposed the integrated services (IntServ) architecture [3] as a QoS architecture for IP networks. IntServ, developed in the early and mid-1990s, provides the ability to give individual flows *absolute* QoS guarantees on packet delays (delay bounds), and packet losses (no loss), as long as the traffic of each flow conforms to a prespecified set of parameters. At each router, each incoming packet has to be inspected to determine which service guarantees it must receive (per-flow classification). Each router therefore has to keep per-flow state information, which raises concerns regarding the scalability of the IntServ architecture, particularly in light of the over-

This work is supported in part by the National Science Foundation through grants ANI-9730103 and ANI-0085955.

0163-6804/03/$17.00 © 2003 IEEE

head of the associated reservation mechanisms, and has prevented IntServ from gaining wide acceptance.

In the second half of the 1990s, the interest in QoS shifted to architectures that make a distinction between operations performed in the network core, and operations performed at the edges of the network. The basic idea is that the amount of traffic in the network core does not permit complex QoS mechanisms, and that most of the QoS mechanisms should be executed at the network edge, where the volume of traffic is smaller. These recent efforts resulted in the differentiated services (DiffServ) architecture [4], which bundles flows with similar QoS requirements in classes of traffic. The mapping from individual flows to *classes* of traffic is determined at the edges of the network, where computational resources are less scarce than in the core. In the core, scheduling primitives only work with a few classes of traffic, and can thus remain relatively simple. DiffServ currently offers two different types of service in addition to best effort: assured forwarding (AF) and expedited forwarding (EF). AF only provides isolation between different classes of traffic, and *qualitative* loss differentiation between so-called *drop precedence levels* within each class. In times of congestion, packets are dropped with a probability function of their drop precedence. Such a qualitative service architecture can be implemented with simple algorithms and does not require per-flow classification or signaling. However, since the AF service only provides qualitative relative guarantees, service assurance is limited. Conversely, EF offers absolute service guarantees on delay variations for classes of traffic. In essence, providing the EF service to a flow is equivalent to providing a virtual leased line to this flow, which leads to low resource utilization. Thus, it is envisioned that the EF service can only apply to a very limited number of flows [5]. More recently, some research studies have aimed to strengthen the service assurance provided by qualitative relative service models such as AF service. For instance, the *proportional differentiated services* model [6] defines a service model with no admission control where the ratios of loss rates and packet delays between successive priority classes remain roughly constant (*proportional* service guarantees).

From these previous research efforts, it appears that there is a trade-off between simplicity of implementation and strength of service guarantees, as shown in Fig. 1. On one hand, IntServ and EF may be too complex to be realized for large flow populations. On the other hand, AF only supports weak QoS guarantees. As shown in Fig. 1, an "ideal" service architecture (i.e., a service architecture that can be deployed on a large network) should provide strong service guarantees with limited complexity. To this date, the most significant advance in devising such an ideal service is probably the SCORE/CSFQ architecture [5]. SCORE tries to reconcile the strength of the IntServ guarantees with the simplicity of the DiffServ architecture, by moving the state information needed to provide IntServ-like service guarantees from network routers to IP packets. Unfortunately,

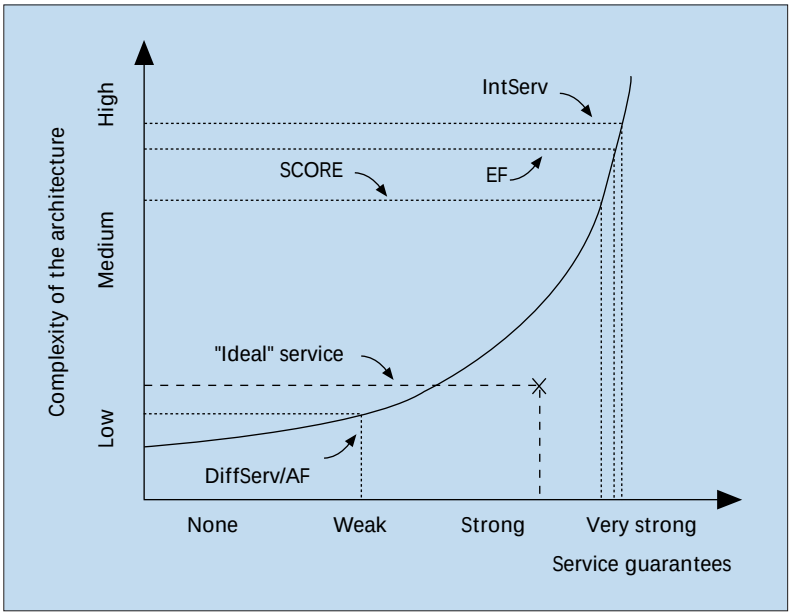

Figure 1. *The trade-off between strength of service guarantees and complexity of the implementation. IntServ provides very strong service guarantees at the price of very high complexity, while DiffServ only provides limited service assurance, but has low complexity. The ideal service should be able to provide strong service differentiation with low complexity.*

SCORE does not alleviate the need for packet classification, which can be computationally expensive, and requires a change in the format of the IP header, which raises questions on the efforts needed to deploy the service.

To achieve the combined goal of providing strong service guarantees with low complexity, we believe one needs to revisit the current tenets of Internet QoS. In previous papers [7–9], we provided technical descriptions of several components (scheduling, buffer management, traffic regulation) that can be used as building blocks of a class-based service architecture with strong differentiation and low complexity. In this article we present these components in a larger context and, for the first time, describe how all components can be combined to build a complete service architecture.

The proposed service architecture relies on the following key concepts. First, service rate allocation to traffic classes is adaptive. The adaptive service rate allocation is conditioned by the instantaneous backlog of traffic classes, service guarantees, and availability of resources. In practice, such an adaptive service rate allocation is realized by an algorithm that jointly performs buffer management and rate allocation. Second, traffic regulation relies on the feedback capabilities of TCP traffic [10] and explicit congestion notification (ECN) [11]. There is no signaling, admission control, or policing of traffic.

The remainder of this article is organized as follows. We first provide an overview of the service architecture we envision for providing a scalable service with strong guarantees. We next describe the Joint Buffer Management and Scheduling (JoBS) framework, which is central to our architecture. Then, we outline how traffic can be regulated using ECN. Last, we draw brief conclusions.

A per-hop, per-class service architecture can be used to build end-to-end service guarantees, for instance if the end applications are in charge of dynamically selecting which class of traffic they require.

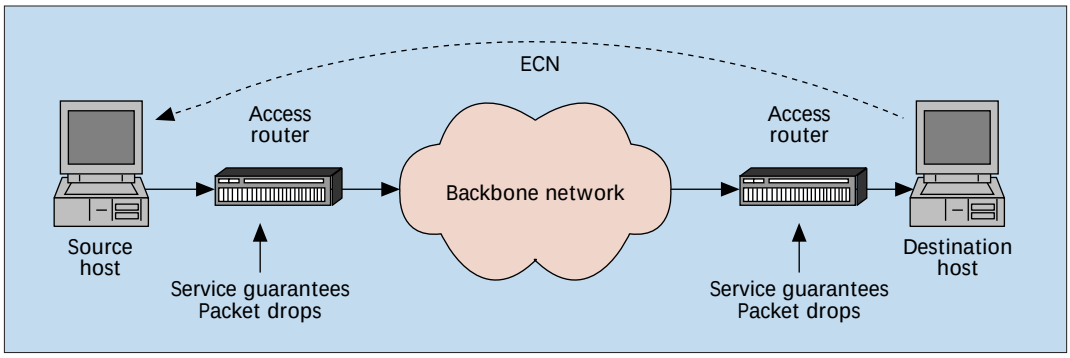

Figure 2. *Deployment of the proposed service in a network. Routers are in charge of transmitting and dropping packets according to the available resources and desired QoS. Routers set the regulation signals (ECN), which are used by the end hosts to regulate their traffic.*

SERVICE ARCHITECTURE

In this section we sketch our solution to the problem of providing strong service differentiation in a scalable manner. We illustrate how our proposed service architecture is deployed in a network in Fig. 2. In this simplified representation, traffic is sent from a source host to a destination host. The source host is connected to the backbone via an access router that supports local per-class service guarantees. Likewise, an access router connects the destination host to the backbone. Based on the service guarantees and available resources, each router dynamically allocates service rates to traffic classes. Packet scheduling at each router directly follows from the service rate allocation. The volume of traffic in the network is controlled by discarding traffic at routers, and sending feedback from the destinations to the traffic sources to reduce the volume of traffic. There is no communication (i.e., signaling) between the different routers, the rate allocation is independent at each router, and the service guarantees provided are also independent at each router. Therefore, the service architecture can be incrementally deployed. Each router that supports the proposed service improves the QoS observed in the entire network. Furthermore, while the example of Fig. 2 assumes that QoS is only needed at access links, the service can also be implemented in routers in the network core. The proposed service architecture indeed allows for incremental deployment, and never degrades the service provided in the network. We next discuss in more detail the service guarantees, packet scheduling and dropping, and traffic regulation.

Service Guarantees — The service we propose consists of per-hop per-class guarantees. These guarantees do not immediately translate into end-to-end service guarantees. However, a per-hop per-class service architecture can be used to build end-to-end service guarantees, for instance, if the end applications are in charge of dynamically selecting which class of traffic they require.

Our goal is to provide a set of service guarantees that can encompass all of AF, proportional differentiated services, and other class-based services. More generally, we want to be able to enforce any mix of absolute and proportional guarantees at each participating router. The service guarantees are independent at each participating router. We refer to this service as *quantitative assured forwarding* (QAF) service [8]. Absolute guarantees apply to loss rates, delays, or throughput, and define a lower bound on the service received by each class. Proportional guarantees apply to loss rates and queuing delays. As an example of the guarantees in the QAF service for three classes of traffic, one could specify service guarantees of the form "Class-1 Delay ″ 2 ms," "Class-2 Delay 4 · Class-1 Delay," "Class-2 Loss Rate £ 1 percent," "Class-3 Loss Rate 2 · Class-2 Loss Rate," and "Class-3 Service Rate ≥ 1 Mb/s" at a given router, and other values at another router. Note that, contrary to the AF service, which provides three levels of drop precedence within a class of traffic, QAF only provides one drop level per class. However, it can be shown that the QAF service can be used to emulate the AF service by assigning each AF drop level to a separate QAF class. Since the QAF service supports absolute guarantees on delays, it can also emulate the EF service. Hence, our proposed service model can implement and interoperate with DiffServ networks, with the possible addition of remarking primitives at the boundaries between DiffServ and QAF domains.

Scheduling and Dropping — The desired service guarantees are realized independently at each router by scheduling and dropping algorithms. Scheduling is based on a service rate allocation to classes of traffic, which share a common buffer. The rate allocation adapts to the traffic demand from different classes. The rates are set so that the per-hop service guarantees are met. If this is not feasible, traffic is dropped. In practice, rate allocation and buffer management are combined in a single algorithm, JoBS, which recomputes the service rate allocation to classes of traffic at the same time it makes dropping decisions. The service rate allocation is independent at each router, and there is no coordination among different routers, which allows for low computational overhead.

Regulating Traffic Arrivals — An end-to-end mechanism has to be in charge of controlling the amount of traffic entering the network, in order to ensure that service guarantees can be met. Traditional approaches to

QoS use a combination of admission control and per-flow traffic policing. Due to the significant overhead involved in keeping track of the number of flows and the traffic submitted by each flow, these mechanisms are not generally usable in a large-scale network. In our architecture, we completely abandon admission control and policing. Instead, we regulate the amount of traffic that enters the network by relying on the congestion control algorithms of TCP used in conjunction with ECN. The traffic volume of UDP traffic is controlled by dropping traffic at routers.

JOINT BUFFER MANAGEMENT AND SCHEDULING

In this section we present the concept of JoBS [8, 9], which is the core of our service architecture. For the discussion here, we assume that no traffic regulation is performed, and that at times when not all service guarantees can be met, some service guarantees are temporarily relaxed. We defer the discussion of traffic regulation to a later section. We first introduce a framework for reasoning about per-class service guarantees and then formulate the realization of these service guarantees in terms of an optimization problem. We then turn to a brief description of a heuristic approximation of the optimization problem that can be implemented in high-speed routers.

A FRAMEWORK FOR SCALABLE SERVICE GUARANTEES

Consider a specific router in the network and a specific output link of this router, where per-hop per-class service differentiation is desired. We assume that all traffic arriving at the transmission queue of the considered output link is marked to belong to one of N classes.

In our framework, service guarantees are enforced independently for each busy period, where a busy period is a time period where the output queue is backlogged. Providing service differentiation only within a busy period requires little state information, and therefore, keeps the implementation overhead limited. Furthermore, it allows the output link to quickly react to changes in traffic load. As a disadvantage, at times of low load, when busy periods are short, providing service differentiation only with information on the current busy period can be unreliable. However, at underloaded links transmission queues are mostly idle, and all service classes receive high-grade service.

We now provide a set of definitions to express our service differentiation objectives. At the time of a packet arrival, the arrival curve is defined as the cumulative amount of traffic that has arrived to the transmission queue of the considered output link since the beginning of the current busy period. The input curve is the cumulative amount of traffic that has been entered into the transmission queue since the beginning of the current busy period. The output curve is the cumulative amount of traffic that has been transmitted since the beginning of the current busy period. In Fig. 3, we illustrate the concepts of arrival curve, input curve, and output curve for class i traffic. At any time, the service rate, r_i, is the slope of the output curve. The service rate of class i is set to zero if no class i traffic is backlogged. In the figure, the service rate is adjusted at n_1, n_2, and n_3, where n_1, n_2, n_3 denote times since the beginning of the current busy period.

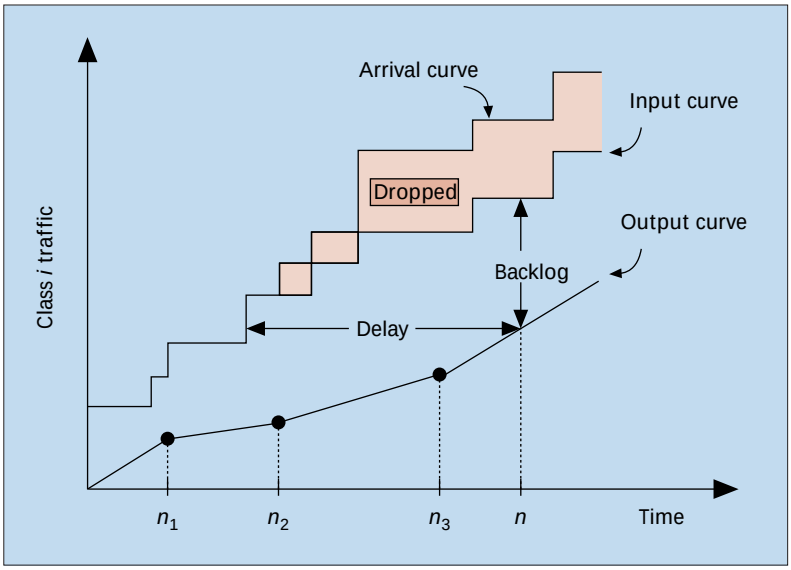

Figure 3. *Delay and backlog at the transmission queue of an output link.*

As illustrated in Fig. 3, at time n the vertical and horizontal distance between the input and output curves, respectively, denote the class i backlog and delay. We define the loss rate of class i to be the ratio of dropped traffic to arrivals, averaged over the time since the beginning of the current busy period.

Service Guarantees — With these metrics, we can express the service guarantees of the QAF service. An absolute delay guarantee on class i imposes that all class i delays be smaller than a given delay bound d_i. Similarly, an absolute loss rate bound for class i imposes that the loss rates constantly remain below a bound L_i. An absolute rate guarantee for class i ensures that the service rate of class i remains above a minimum bound f_i at any time there is a backlog of class i traffic in the output queue. Proportional guarantees on delay (resp. losses) require that the ratios of the delays (resp. loss rates) of two classes with successive indices are constant. The constants are arbitrarily chosen and quantify the proportional differentiation desired. We refer to [8, 9] for a detailed specification of the service guarantees.

SERVICE RATE ALLOCATION AND PACKET DROPPING: AN OPTIMIZATION PROBLEM

Viewing arrivals and service of a traffic class as in Fig. 3, the desired service guarantees can be enforced by selecting the service rate allocation for classes and making decisions to drop traffic at appropriate times. In [9] we showed that the service rate allocation and packet drops could be viewed in terms of a nonlinear optimization

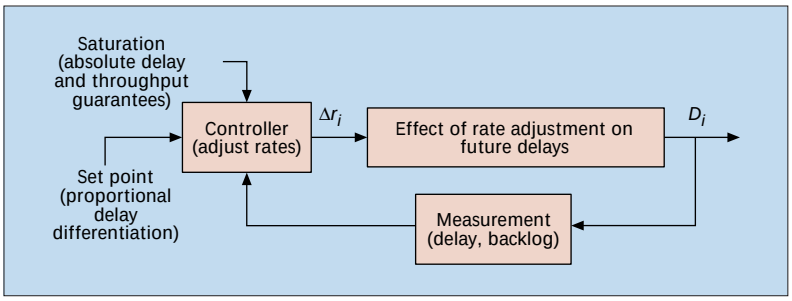

Figure 4. *An overview of the feedback-based approach. This is an overview of the delay feedback loop for class* i*. There is one such loop per class.*

problem, which we summarize here. We refer the reader to [9] for a formal specification of the optimization problem.

The optimization variable x_n is a vector containing the service rates $r_i(n)$ and the amounts of traffic to be dropped from each class, $l_i(n)$, where n denotes the time of an arrival. The optimization problem is posed as minimizing the value of an objective function $F(x_n)$, subject to a set of constraints on x_n.

Even though the choice of the objective function is a policy decision, we select two specific objectives that we believe have general validity:

- As much as possible, the buffer management scheme should avoid dropping traffic.
- The scheduler should, as much as possible, avoid making changes to the current service rate allocation.

The first objective ensures that traffic is dropped only if there is no alternative way to satisfy the constraints. The second objective tries to hold on to a feasible service rate allocation as long as possible. We give the first objective priority over the second objective. Provided that the constraints can be satisfied, the chosen objective function F will select a solution for x_n. The optimization at time n is done with knowledge of the arrival and output curves up to time n.

The constraints of the optimization problem are system constraints, which describe physical limitations (e.g., finite buffer size) and properties of the output link (e.g., work-conserving link), and QoS constraints, which result from the service guarantees offered, as described above. QoS constraints for classes that are not backlogged are simply ignored. While the absolute loss and throughput guarantees can be directly used as QoS constraints, the delay guarantees need to be adapted. Indeed, the delay metric we chose in Fig. 3 characterizes the delay of traffic *leaving* the transmission queue at time n; thus, there is no direct relationship between the delay at time n and the service rate allocated at time n. To include the delay guarantees in the optimization problem, the scheduler projects future delays of traffic backlogged at time n. If not all constraints can be satisfied at the same time, as can be the case without traffic regulation, a precedence order on the "importance" of the constraints is used to temporarily relax some constraints.

The structure of constraints and objective function makes the optimization process in JoBS a *nonlinear optimization problem*, which can be solved with available numerical algorithms.

A Feedback-Based Heuristic Algorithm

The optimization model described above is computationally expensive. Hence, this solution is impractical for an implementation at high speeds. To provide a scheme that can be implemented at high data rates, we devised feedback control algorithms that approximate the optimization described above [8]. As in the optimization model, the service rates $r_i(n)$ and the amount of dropped traffic $l_i(n)$ are adjusted upon each arrival to attempt to meet the desired service guarantees. We next characterize the service rate allocation and dropping algorithm as feedback control problems.

Service Rate Allocation — We have one feedback loop for each class i with proportional delay guarantees. Instead of directly computing the service rate $r_i(n)$, we compute the adjustment $r_i(n)$ to the service rate, compared to its previous value $r_i(n-1)$. $r_i(n)$ is selected such that the constraints of proportional delay differentiation are satisfied at time n. By monitoring the delay of class i traffic leaving the router, we can determine the deviation from the desired proportional differentiation, resulting from past service rate allocations, and infer the adjustment to the service rate needed to attenuate this deviation. Using control theory terminology, the function that maps a monitored delay to a given rate adjustment is the *controller*.

We illustrate the feedback loop in Fig. 4. The set point of the controller is defined by the proportional delay differentiation desired, and the action of the controller is limited by saturation constraints that translate limitations on the rate allocated to class i due to system constraints, and absolute delay and rate guarantees. We showed in [8] that a set of first-order approximations around an operating point allowed for designing a simple controller, implementable at high speeds, consisting of a multiplication between a time-dependent proportional coefficient and the difference between the observed delay and the delay required for proportional delay differentiation. The time-dependent proportional coefficient is chosen so that saturation constraints are respected and the loop is stable.

Packet Dropping — Traffic must be dropped, from either a new arrival or the current backlog, at times when no feasible service rate allocation for meeting all delay guarantees exist, or the transmission queue is full. To satisfy proportional loss guarantees, traffic is dropped from classes according to a drop priority order, defined as follows. For each class, the difference between the loss rate needed for perfect proportional differentiation and the observed loss rate defines an error. The larger the error of a class, the higher its drop priority. For each class i, we stop dropping traffic when either:

- The loss guarantee L_i is reached
- The buffer overflow is resolved or a feasible rate allocation for absolute guarantees exists and there is no need to drop traffic anymore

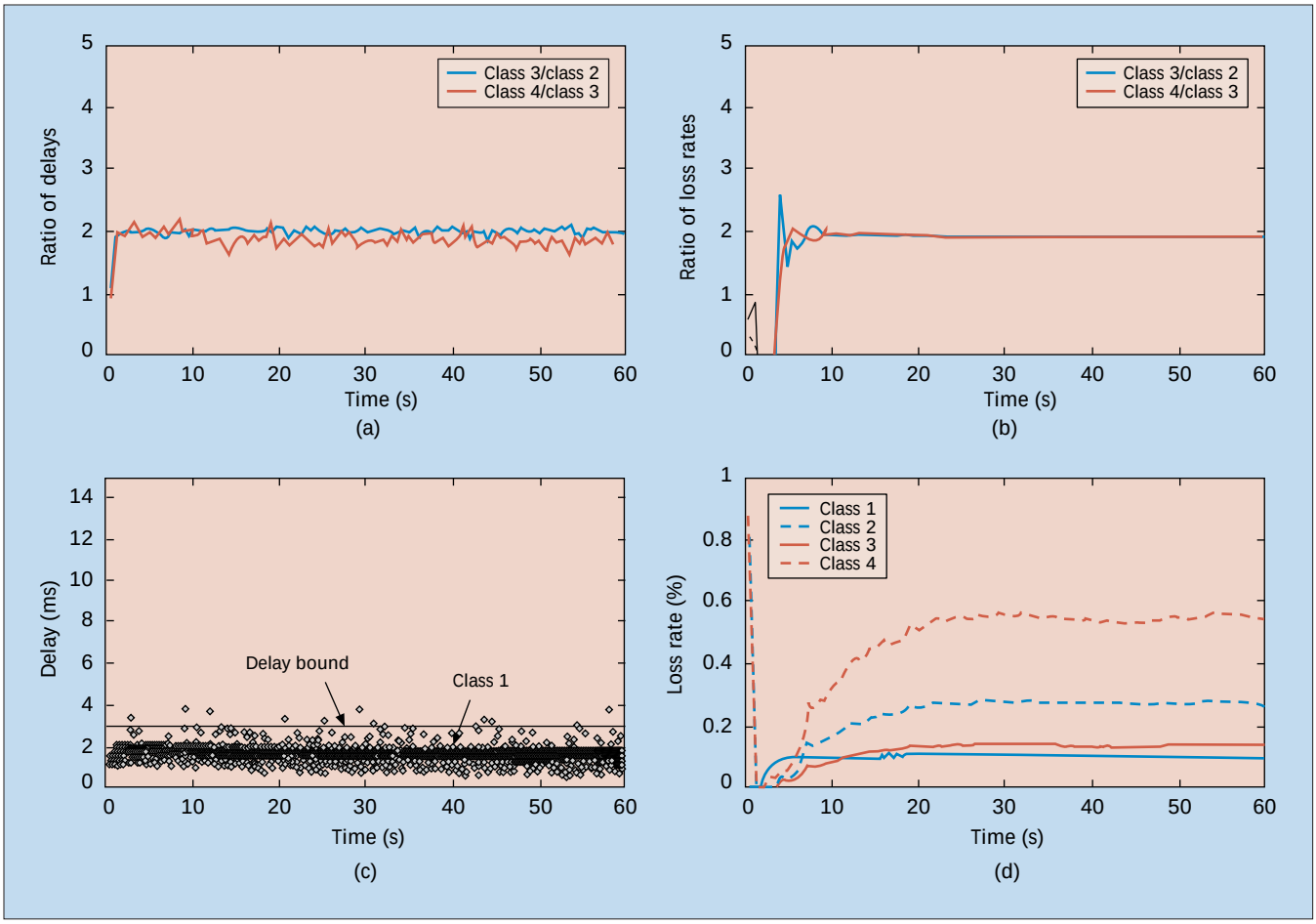

Figure 5. *Performance evaluation of the feedback-based algorithm, on a testbed of FreeBSD PC-routers. Results are averaged over a moving window of size 0.1 s, with the exception of c), which represents the delay of* each *class 1 packet. a) Ratios of delays; b)ratios of loss rates; c) class 1 delays (individual) ; d) loss rates.*

EVALUATION

Simulation experiments using highly bursty Pareto sources [9] indicate that a heuristic such as the feedback-based algorithm achieves an accurate approximation of the optimization-based algorithm under highly variable loads oscillating between 70 and 140 percent of the link capacity. Here, we summarize experimental results gathered on a testbed of PC routers that show the feedback-based algorithm is efficient at providing the desired service guarantees.

We implemented the feedback-based algorithm in PC routers running FreeBSD 4.3 and `altq-3.0` [12]. Our implementation is now available as part of the standard `altq-3.1` distribution. The results presented in Fig. 5 were obtained by running a single-node experiment on a FastEthernet network (100 Mb/s) with four classes of traffic. Class 1 consists of Pareto on-off UDP sources, while the three other classes consist of nonsynchronized greedy TCP traffic sources. The load is almost constant and equal to 100 percent of the link capacity. All classes initially contribute roughly the same amount to the traffic mix. The objective is to provide absolute delay and loss bounds to class 1, and proportional delay and loss differentiation, with a factor of 2, between classes 2, 3, and 4.

Figures 5a and b illustrate that the proportional differentiation objectives are met. Figure 5c shows that, except for a small number of packets (< 1 percent) that present deadline violations, JoBS is able to satisfy all delay constraints at the same time. Figure 5d indicates that the chosen loss rate bound of 0.1 percent on class 1 is consistently respected. When meeting all service guarantees at the same time is not feasible, JoBS relaxes delay guarantees in favor of loss guarantees, which explains why a small fraction of packets miss their deadline. Additional plots presented in [8] and the references therein include more complex topologies and load, and verify that the aggregate throughput of all classes is close to the link capacity of 100 Mb/s, and no class is starved. A large-scale simulation experiment is available in [9], and demonstrates the efficiency of the scheme at realizing the service objectives in a complex multihop network with multiple bottlenecks and a relatively large number of sources.

The experimental results outlined here indicate that, even in the absence of traffic regulation, JoBS manages to provide service differentiation close to what is desired. When the set of constraints resulting from the service guarantees is infeasible, JoBS temporarily relaxes some constraints. Because routers operate independently, processing times at the transmission queue are the key factor in the scalability of

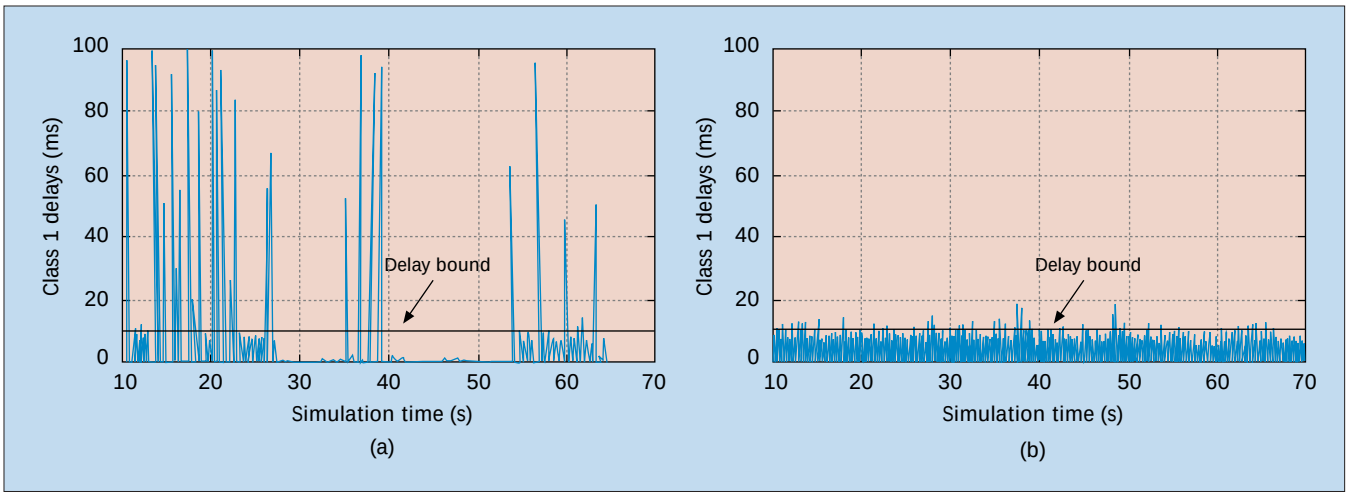

Figure 6. *Comparison between class 1 delays in a) JoBS without traffic reglation and b) JoBS with traffic regulation.*

the architecture. A study of the implementation overhead [8] indicated that a 1 GHz PC can enqueue and dequeue more than 50,000 packets/s in the case of a relatively stringent set of constraints, and if the service rates are adjusted upon *each* packet arrival. Considering that the average size of a packet on the Internet is approximately 450 bytes [13], this results in a maximum throughput of at least 180 Mb/s. To be able to forward packets at higher speeds, one can perform service rate adjustments only once in *n* packet arrivals. Such sampling comes at the expense of the accuracy of the service differentiation, but we did not observe much degradation in service differentiation for sampling intervals up to approximately 10–20 packets, which would be appropriate for gigabit-per-second links. Thus, we believe that our approach is viable at high speeds.

Traffic Regulation Using ECN

We next turn to the description of the approach we envision for traffic regulation. Without traffic regulation, it may be impossible to satisfy a set of absolute service guarantees at a given time, in which case JoBS selectively relaxes some service guarantees. This situation occurs when an absolute loss bound and an absolute delay bound are conflicting, that is, when traffic must be dropped to satisfy a delay bound, but cannot be dropped without violating a loss rate bound. This situation is illustrated in Fig. 5c, which shows that some packets miss their deadline.

To remedy this problem, we propose using ECN at each router to regulate traffic arrivals, instead of admission control or traffic policing. With ECN, the destination of a TCP flow checks if the Congestion Experienced codepoint of the TCP header is set. If so, the destination marks the acknowledgment with the Congestion Echo codepoint. When the source receives an acknowledgment marked with the Congestion Echo codepoint, the source reduces its TCP congestion window by half, ultimately resulting in a decrease in the sending rate. The solution we propose for avoiding conflicting service guarantees is to proactively mark packets with the ECN Congestion Experienced codepoint before conflicts between service guarantees occur. A complete study of the use of ECN for traffic regulation and drop avoidance can be found in [7].

In the context of service differentiation, we propose to separate TCP and non-TCP traffic into different classes. Non-TCP traffic cannot be regulated by packet marking; therefore, we allow relaxation of service guarantees for non-TCP traffic classes. For TCP classes, the input curves can be projected in the future by tracking the congestion window sizes, round-trip times, and maximum segment sizes of all backlogged TCP flows. Using these projections, we can predict future conflicts between service guarantees and proactively mark packets with the ECN Congestion Experienced codepoint to limit the traffic arrivals at the router and avoid such impending conflicts.

Unfortunately, the proposed method requires maintenance of per-flow information, which defeats our scalability criteria. However, it has been shown that a very small number of flows, or "heavy hitters," contribute to the majority of traffic [14]. Thus, sampling techniques such as multistage filters can be used to filter out heavy hitters and drastically reduce the amount of state information to be maintained [15].

Note that the proposed technique resembles proactive marking/dropping algorithms such as Random Early Drop (RED) [16]. The major difference with our proposed approach lies in the fact that RED uses probabilistic arguments to avoid maintaining state information, which has led to questioning its configurability. Conversely, our proposed mechanism maintains limited state information, since only long-lived TCP flows are tracked and represent a small fraction of the total number of flows. The algorithm can then use this state information to have relatively accurate estimates of the input curves over a short time interval, and make deterministic marking decisions.

To assess the viability of our algorithm for traffic regulation via ECN, we simulated a bottleneck link where traffic regulation is absolutely needed, because the buffer size at the output

queue (250 kbytes) is small compared to the output link capacity (45 Mb/s), and stringent loss rate bounds (1 percent) and delay bounds (10 ms) are offered to a class of traffic, say class 1. This bottleneck is traversed by over 60 TCP flows with round-trip times uniformly distributed between 44 and 80 ms. About 20 percent of the TCP flows are greedy TCP flows (heavy hitters), while the others are on-off traffic.

Figure 6a shows that with this experimental setup, in the absence of traffic regulation, many class 1 packets miss their deadlines; it also indicates that class 1 traffic may be starved at times (e.g., between t = 30 s and t = 50 s). Conversely, violations are much rarer and of lesser magnitude when traffic regulation is performed, as shown in Fig. 6b. We point the reader to additional measurements available in [7] for a comparison with other algorithms such as RED. Additionally, the results in [7] show that no traffic is ever dropped when traffic regulation is performed, and that the addition of the traffic regulation algorithm does not reduce the link utilization, nor does it defeat service objectives on loss or throughput, or impact other classes. Thus, despite the fact that some violations of the delay bound still remain, their limited magnitude leads us to believe that traffic regulation using ECN can be a viable alternative to admission control and traffic policing, which prevent violations from happening, but may raise scalability and/or underutilization concerns.

CONCLUSIONS

We sketch the design of a scalable service architecture for providing strong service guarantees. We formally define our notion of strong service guarantees by presenting the quantitative assured forwarding service, and then discuss mechanisms to implement the service. A key concept of the architecture is to use an adaptive service rate allocation to traffic classes, realized by an algorithm combining scheduling and buffer management. Based on a reference algorithm that dynamically solves a nonlinear optimization, we introduce a heuristic relying on feedback control theory. We then discuss how to perform traffic regulation by using the feedback capabilities of TCP, used in conjunction with ECN. A prototype implementation in PC routers of the scheduling and buffer management algorithm is distributed as part of the recent `altq-3.1` package, and is available at http://qosbox.cs.virginia.edu.

REFERENCES

[1] D. Ferrari and D. Verma, "A Scheme for Real-time Channel Establishment in Wide-area Networks," *IEEE JSAC* vol. 8, no. 3, Apr. 1990, pp. 368–79.

[2] D. Clark, S. Shenker, and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism," *Proc. ACM SIGCOMM '92*, Baltimore, MD, Aug. 1992, pp. 14–26.

[3] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," IETF RFC 1633, July 1994.

[4] S. Blake *et al.*, "An Architecture for Differentiated Services," IETF RFC 2475, Dec. 1998.

[5] I. Stoica and H. Zhang, "Providing Guaranteed Services Without Per-flow Management," *Proc. ACM SIGCOMM '99*, Boston, MA, Aug. 1999, pp. 81–94.

[6] C. Dovrolis and P. Ramanathan, "A Case for Relative Differentiated Services and the Proportional Differentiation Model," *IEEE Network*, vol. 13, no. 5, Sept. 1999, Special Issue on Integrated and Differentiated Services on the Internet, pp. 26–34.

[7] N. Christin and J. Liebeherr, "Marking Algorithms for Service Differentiation of TCP Traffic," Tech. rep. CS-2003-04, Univ. of VA, Feb. 2003, tp://ftp.cs.virginia.edu/pub/techreports/CS-2003-04.pdf.

[8] N. Christin, J. Liebeherr, and T. F. Abdelzaher, "A Quantitative Assured Forwarding Service," *Proc. IEEE INFOCOM 2002*, New York, NY, June 2002, vol. 2, pp. 864–73.

[9] J. Liebeherr and N. Christin, "Rate Allocation and Buffer Management for Differentiated Services," *Comp. Nets.*, vol. 40, no. 1, Sept. 2002, pp. 89–110.

[10] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," IETF RFC 2581, Apr. 1999.

[11] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," IETF RFC 3168, Sept. 2001.

[12] K. Cho, "A Framework for Alternate Queuing: Towards Traffic Management by PC-UNIX based Routers," *Proc. USENIX '98 Annual Tech. Conf.*, New Orleans, LA, June 1998, pp. 247–58.

[13] K. Claffy, G. Miller, and K. Thompson, "The Nature of the Beast: Recent Traffic Measurement from an Internet Backbone," *Proc. INET '98*, Geneva, Switzerland, July 1998.

[14] W. Fang and L. Peterson, "Inter-AS Traffic Patterns and Their Implications," *Proc. IEEE GLOBECOM '99*, Rio de Janeiro, Brazil, Dec. 1999, pp. 1859–68.

[15] C. Estan and G. Varghese, "New Directions in Traffic Measurement and Accounting," *Proc. ACM SIGCOMM '02*, Pittsburgh, PA, Aug. 2002, pp. 323–36.

[16] S. Floyd and V. Jacobson, "Random Early Detection for Congestion Avoidance," *IEEE/ACM Trans. Net.*, vol. 1, no. 4, July, 1993, pp. 397–413.

BIOGRAPHIES

NICOLAS CHRISTIN (nicolas@cs.virginia.edu) received an engineering degree from Ecole Centrale Lille, France, in 1999, and a Master's in computer science from the University of Virginia in 2000. In 2002-2003 he worked in Nortel Networks' Advanced Technology group. He is currently pursuing a Ph.D. in computer networks at the University of Virginia, which he expects to complete in 2003. His research interests include scalable service architectures, implementation aspects, and network measurements.

JÖRG LIEBEHERR [M] (jorg@cs.virginia.edu) received a Ph.D. degree in computer science from Georgia Tech in 1991. He is currently an associate professor and a Faculty Fellow in the Department of Computer Science at the University of Virginia. He serves on the editorial boards of *ACM/IEEE Transactions on Networking*, *IEEE Network*, *Computer Communications*, *Real-Time Systems Journal*, *ACM/Springer Multimedia Systems*, and *Cluster Computing*. He served as Editor-in Chief of *IEEE Network* in 1999 and 2000. He was elected to the Board of Governors of the IEEE Communications Society for 2003–2005.

Despite the fact that some violations of the delay bound still remain, their limited magnitude leads us to believe that traffic regulation using ECN can be a viable alternative to admission control and traffic policing.