Efficient Shaping of User-specified QoS using Aggregate-flow Control*

Huan Ren and Kihong Park**

Network Systems Lab, Dept. of Computer Sciences, Purdue University, West Lafayette, IN 47907, U.S.A. {renh,park}@cs.purdue.edu

Abstract. Architecting networks capable of providing scalable, efficient, and fair services to users with diverse QoS requirements is a pressing problem. The two principal issues are: design of "good" per-hop behavior and edge control. In previous work [2, 3], we studied aggregate-flow QoS control from a noncooperative resource provisioning context. In [20], the framework was generalized by, one, solving an optimal aggregate-flow per-hop behavior problem, and two, showing how it can be used coupled with end-to-end label control to facilitate scalable and fair QoS when driven by selfish users and service providers. In this paper, we focus on optimal aggregate-flow per-hop control and complement analysis by experimental performance evaluation. We show that user-specified, diverse QoS is efficiently facilitated over the optimal per-hop behavior network substrate using adaptive label control end-to-end.

Keywords: Differentiated services, optimal per-hop behavior, adaptive label control, class-based IP switching

1 Introduction

Architecting networks capable of providing scalable, efficient, and fair services to users with diverse QoS requirements is a challenging problem. The traditional approach uses resource reservation and admission control to provide both *guarantees* and *graded* services to application traffic flows. Analytical tools for computing and provisioning QoS guarantees [7, 8, 17] rely on overprovisioning coupled with traffic shaping/policing to preserve well-behavedness properties across switches that implement a form of GPS packet scheduling. For applications needing guaranteed services, the unconditional protection afforded by per-flow resource reservation and admission control is a necessity. For the population of elastic applications that require QoS-sensitive services but not guarantees, it would be overkill to provision QoS using the mechanisms of per-flow reservation and admission control. In addition to the service mismatch, overhead associated with administering resource reservation and admission control which require per-flow state at routers impedes scalability.

* Supported in part by NSF grants ANI-9875789 (CAREER) and EIA-9972883.

** Additionally supported by NSF grants ANI-9714707, ESS-9806741, and grants from the Purdue Research Foundation, Santa Fe Institute, and Sprint.

Recently, efforts have been directed at designing network architectures with the aim of delivering QoS-sensitive services by introducing weaker forms of protection or assurance to achieve scalability [2, 6, 9, 14]. The differentiated services framework [1, 6, 12] has advanced a set of building blocks comprised of per-hop and access point behaviors with the aim of facilitating scalable services through aggregate-flow resource control inside the network and per-flow traffic control at the edge. By performing a many-to-one mapping from the large space of individual flows to the much smaller space of aggregate flow labels, scalability of per-hop control is achieved at the expense of introducing uncertainty and volatility by flow-aggregation and aggregate-flow packet switching per-hop. Simplified models of Assured Service [12] and Premium Service [13] were presented in [15] and analyzed with respect to their performance when compared with simulations. In [10], an adaptive 1-bit marking scheme is described, and the resulting bandwidth sharing behavior demonstrated via simulations when the priority level is controlled end-to-end. In [9], the authors describe a proportional differentiation model which seeks to achieve robust, configurable service class separation—i.e., QoS differentiation—with the support of two candidate packet schedulers. They use simulation to study the behavioral properties. In previous work [2, 3, 18], we studied aggregate-flow per-hop control mechanisms and end-to-end controls motivated by game theoretic considerations—a router performs class-based label switching which emulates user optimal service class selection with respect to selfishness—without considering the space of all aggregate-flow per-hop controls. The generalization to optimal per-hop behavior design was carried out in [20]. Stability and efficiency of the overall system—when driven by selfish users and service providers—was studied in the game theoretic context. Other related works include [14, 16]. In this paper, we focus on optimal aggregate-flow per-hop control and its properties. We complement the theoretical development by simulation-based performance evaluation of end-to-end QoS provisioning over the optimal aggregate-flow classifier substrate. We show that scalable, user-specified diverse QoS is effectively facilitated through the joint action of end-to-end adaptive label control and optimal aggregate-flow per-hop control.

The rest of the paper is organized as follows. In the next section, we give a description of the QoS provisioning architecture. In Section 3, we present the optimal aggregate-flow classifier and its properties. Section 4 shows performance results of its QoS provisioning properties.

2 Network Architecture

2.1 Overall System Structure

Assume there are n flows or users. A user $i \in [1, n]$ sends a traffic stream at average rate $\lambda_i \geq 0$ (bps). We will assume that λ_i is given and fixed ("fixed bandwidth demand"). The case when λ_i is variable ("variable bandwidth demand") is considered separately. It is also possible to interpret λ_i as data volume (bytes), for example, file size when files are accessed by reliable transport protocols. Let

$\mathbf{x}^i = (x_1^i, x_2^i, \dots, x_s^i)$ denote the vector of end-to-end QoS rendered to user i. For example, x_1^i may represent mean delay, x_2^i packet loss rate, x_3^i delay jitter (e.g., as measured by some second-order statistic), and so forth. We assume that all QoS measures are represented such that a smaller magnitude means better QoS. Each packet belonging to user i is enscribed with a scalar $\eta_i \in \{1, 2, \dots, L\}$ taking on L distinct values. Typically, the number of users is very large vis-à-vis the range of η_i, i.e., $n \gg L$, and per-flow identity—as conveyed by η_i—is lost as soon as a packet enters the network.

2.2 Per-hop Control

Per-hop control consists of a *classifier* and a *packet scheduler*. We assume a GPS packet scheduler with m service classes and service weights $\alpha_k \geq 0$, $\sum_{k=1}^m \alpha_k = 1$, for an output port whose link bandwidth μ is shared in accordance with the service weights. It is not necessary to have GPS as the underlying packet scheduling discipline—e.g., priority queues, multiple copies of RED with different thresholds are alternatives—but we will show that GPS has certain desirable properties when considering the problem of selecting an optimal aggregate-flow per-hop control for differentiated services. The most important component—where decisions about differentiated treatment of packets enscribed by different η values occurs—is the classifier which is given by a map, $\xi : [1, L] \to [1, m]$. That is, n flows—effectively L (or less) flows from the router's perspective since packets are scheduled by their label values only—routed to the same output port on a switch are mapped to m service classes. For *aggregate-flow* control, $n > L$ and $L \geq m$. If $L > m$, this leads to a further aggregation. For some

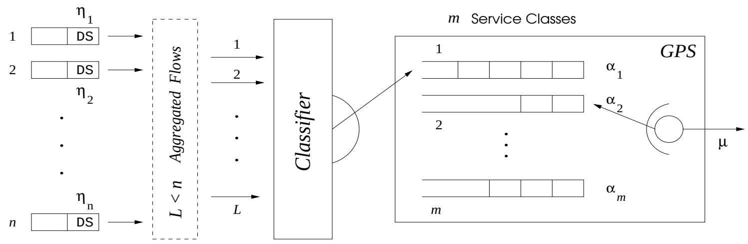

Fig. 2.1. η value in DS field of IP datagram is used by the classifier to select service class in GPS packet scheduler.

choice of classifier and packet scheduler, the QoS received by flow $i \in [1, n]$ at a switch is determined—explicitly or implicitly—by a performance function x^i, $\mathbf{x}^i = x^i(\boldsymbol{\eta}, \boldsymbol{\lambda})$, where $\boldsymbol{\eta} = (\eta_1, \dots, \eta_n)$ and $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n)$. More precisely, flow i's performance, in the aggregate-flow case, is determined by the performance function $x^k(\boldsymbol{\eta}^a, \boldsymbol{\lambda}^a)$ associated with service class $k \in [1, m]$ where

$$k = \xi(\eta_i), \quad \boldsymbol{\eta}^a = (1, 2, \dots, L), \quad \boldsymbol{\lambda}^a = (\lambda_1^a, \lambda_2^a, \dots, \lambda_L^a), \quad \text{and} \quad \lambda_\ell^a = \sum_{j:\eta_j = \ell} \lambda_j.$$

That is, the switch sees only (up to) L "super users" or aggregate flows. The per-hop control structure is depicted in Figure 2.1.

There are three properties of the per-hop control which are relevant from a QoS control perspective. Let $\mathbf{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ denote the unit vector whose i'th ($i \in [1, n]$) component is 1 and 0 for all other components. The properties are:

(A1) for each flow i and configuration $\boldsymbol{\eta}$, $x^i(\boldsymbol{\eta} + \mathbf{e}_i) \leq x^i(\boldsymbol{\eta})$ and $x^i(\boldsymbol{\eta} - \mathbf{e}_i) \geq x^i(\boldsymbol{\eta})$;

(A2) for any two flows $i \neq j$ and configuration $\boldsymbol{\eta}$, $x^j(\boldsymbol{\eta} + \mathbf{e}_i) \geq x^j(\boldsymbol{\eta})$ and $x^j(\boldsymbol{\eta} - \mathbf{e}_i) \leq x^j(\boldsymbol{\eta})$;

(B) for two flows $i \neq j$ and configuration $\boldsymbol{\eta}$, $\eta_i \geq \eta_j$ implies $x^i(\boldsymbol{\eta}) \leq x^j(\boldsymbol{\eta})$.

In the definitions, the range of $\boldsymbol{\eta}$ is such that the perturbations remain in the n-dimensional lattice, i.e., $\boldsymbol{\eta} + \mathbf{e}_i, \boldsymbol{\eta} - \mathbf{e}_i \in [1, L]^n$. Property (A1) states that, other things being equal, increasing the label value of flow i improves the QoS received by flow i. Property (A2) states that increasing η_i will not increase the QoS received by any other flow j. Property (B) states that if flow i has a higher η value than flow j, then the QoS it receives is superior to that of flow j. We call property (B) the *differentiated service* property. (B) has the immediate consequence $x^i(\boldsymbol{\eta}) = x^j(\boldsymbol{\eta})$ iff $\eta_i = \eta_j$. Thus there is no absolute, a priori QoS level attached to the η_i values. It is the magnitude of η_i—*relative* to other flows' label values—that will determine the QoS received by a flow i. Properties (A1) and (B) are desirable from an end-to-end QoS shaping—i.e., label control—point-of-view. Property (A2) is imposed by resource-boundedness.

2.3 Edge Control

The properties exported by per-hop control—if satisfied—are not sufficient by themselves to render end-to-end QoS commensurate with user requirements. End-to-end (or edge) control complements per-hop control by setting the value of η per-flow, either statically—open-loop control—or dynamically—closed-loop control—in accordance with user needs. We assume that the network exercises *access control* at the edge such that users are not permitted to assign η values to their packets arbitrarily—if every user assigns the maximum η value L to their flows, then QoS control via η loses its meaning (degenerates to FIFO-based best-effort service by property (B)).

User i's QoS requirement, in general, can be represented by a *utility function* U_i which has the form $U_i(\lambda_i, \mathbf{x}^i, p_i)$ where λ_i is the traffic rate, $\mathbf{x}^i$ the end-to-end QoS received, and p_i the unit price charged by the service provider. The total cost to user i is given by $p_i \lambda_i$. We assume that U_i satisfies the *monotonicity* properties[1]

$$\partial U_i / \partial \lambda_i \geq 0, \quad \partial U_i / \partial \mathbf{x}^i \leq 0, \quad \text{and} \quad \partial U_i / \partial p_i \leq 0. \tag{2.1}$$

[1] U_i need not be differentiable, nor even be continuous. We use continuous notation here for notational clarity; monotonicity is the only property required.

Other things being equal, an increase in the traffic rate is favourably received by a user, so is an improvement in QoS, but an increase in the price charged by the service provider has a detrimental effect on user satisfaction. These are minimal, weak requirements on the qualitative form of user utility. If η control is allowed to be exercised by the user, then a *selfish* user i can be defined as performing the self-optimization $\max_{\eta_i \in [1,L]} U_i(\lambda_i, \mathbf{x}^i, p_i)$ where η_i influences user i's utility U_i via its effect on the QoS received $\mathbf{x}^i$. We assume $p_i(\mathbf{x}^i)$ is a monotone (nonincreasing) function of $\mathbf{x}^i$—better QoS incurs higher cost—which corresponds to the price function exported by the service provider.

Under the aforementioned assumptions and per-hop control properties, symmetric label control of the form (shown here for threshold utilities)

$$\frac{d\eta_i}{dt} = (-1)^a \varepsilon \, ||\mathbf{x}^i(\boldsymbol{\eta}) - \boldsymbol{\theta}^i|| \tag{2.2}$$

where $a = \mathrm{sgn}(||\boldsymbol{\theta}^i|| - ||\mathbf{x}^i(\boldsymbol{\eta})||)$ and $\mathrm{sgn}(\cdot)$ is 1 if its argument is nonnegative, and 0, otherwise, can be shown to be asymptotically stable. The choice of norm is relevant—the QoS vectors, under the usual ordering relation, form a partially ordered set—and, in general, subtle effects with respect to QoS ordering and resource control can arise [4, 5]. We remark that adaptive label control is a simpler control problem than congestion control since it does not suffer under the latter's unimodal input-output relation [19].

3 Optimal Aggregate-flow Per-hop Control

3.1 Semantics of Optimal Per-hop Behavior

Consider the per-flow control or classifier problem for n users who choose packet labels from $[1, L]$. Technically, per-flow classification means $n = m$, and L is either greater or smaller than n. The range L may be finite or unbounded, and the variable $\eta_i \in [1, L]$ discrete or continuous. When n users mark their flows with a value η_i drawn from the metric space $[1, L]$ (Euclidean distance) with property (A1) satisfied—larger η_i values, other things being equal, result in a greater apportionment of resources and thus better QoS—η_i can be viewed as codifying a user's QoS or resource demand with respect to some measurement unit. If network resources are *infinite*, then a flow's request can be satisfied based on the η_i value specified, without consideration of the needs specified by other flows (except, possibly, for pricing issues). That is, independence or decoupling holds. If, on the other hand, resources are *finite*—an OC-24 link is shared among bandwidth intensive users—then the users' collective resource demand may exceed available bandwidth. In the presence of such *resource contention*, a conflict resolution scheme is needed, including the criteria by which allocation is decided.

Assume available bandwidth is normalized such that total available bandwidth is $\mu = 1$. First, assume $\eta_i \in \mathbb{R}_+$ is a *continuous* variable over the real unit interval $[0, 1]$, expressing user i's normalized bandwidth demand *per unit flow*. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ with $\alpha_i \geq 0$, $\sum_{k=1}^n \alpha_k = 1$, represent the fraction of resources apportioned by the per-flow classifier to $i \in [1, n]$, and let $\omega_i = \alpha_i / \lambda_i$ denote the fraction of resources allocated to i per unit flow. Under the above *semantics*, given $\boldsymbol{\eta}$ and $\boldsymbol{\lambda}$, the optimization

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{n} (\eta_i - \omega_i)^2 \tag{3.1}$$

measures the "goodness" of a resource allocation $\boldsymbol{\omega}$ with respect to users' codified needs $\boldsymbol{\eta}$ in the mean-square sense. Since (3.1) penalizes by the *difference* error, the relative importance of higher η_i values is preserved, and resources are apportioned accordingly. For general $\eta_i \in \mathbb{R}_+$, including the discrete and bounded case $\eta_i \in \{1, \ldots, L\}$ which is of special interest, define the normalization

$$\hat{\eta}_i = \begin{cases} (\eta_i - \eta_{\min})/(\eta_{\max} - \eta_{\min}), & \text{if } \eta_{\max} \neq \eta_{\min}, \\ 1, & \text{otherwise,} \end{cases} \tag{3.2}$$

where $\eta_{\min}$, $\eta_{\max}$ are the minimum and maximum values of $\{\eta_1, \eta_2, \ldots, \eta_n\}$, respectively. Note that $\hat{\eta}_i \in [0, 1]$, and unless all η_i values are equal, $\eta_{\min} = 0$ and $\eta_{\max} = 1$. Similarly, let $\hat{\omega}_i$ denote the normalization of ω_i. Then, given $\boldsymbol{\eta}$, the generalized optimization criterion is

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{n} (\hat{\eta}_i - \hat{\omega}_i)^2. \tag{3.3}$$

(3.3) realizes the same semantics as (3.1), however, generalized by the function or "code" (not 1-1) given by (3.2) to η_i values not restricted to the real unit interval $[0, 1]$. If L is bounded, then the 1-1 function $\hat{\eta}_i = \eta_i / L$ achieves a similar purpose.

Proposition 3.4 *Given $\boldsymbol{\eta}, \boldsymbol{\lambda} \in \mathbb{R}^n_+$, the solution to* (3.3) *is*

$$\alpha_i = (1 - \nu) \frac{\lambda_i \hat{\eta}_i}{\sum_{j=1}^{n} \lambda_j \hat{\eta}_j} + \nu \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j}, \qquad i \in [1, n], \tag{3.5}$$

where $0 \leq \nu \leq 1$ is a parameter defining a continuous family of solutions. The optimal per-flow classifier given by (3.5) *satisfies properties* (A1), (A2), *and* (B).

The parameter ν, which stems from the dimension reduction associated with (3.2), has an appealing interpretation. The second term in (3.5) corresponds to the proportional share achieved by FIFO scheduling, whereas the first term corresponds to proportional share of the corresponding *virtual* flows $\lambda_i \hat{\eta}_i$, which are the original flow rates weighted by their relevancy variable $\hat{\eta}_i$ derived from η_i. Thus (3.5) represents a convex combination of two extreme behavioral modes.

3.2 Optimal Aggregate-flow Classification

Let us consider the aggregate-flow classifier problem where $n > m$. Aggregate-flow control, whether it has many or few labels, must service n flows using $m < n$

service classes which results, in general, in a reduced ability to effectively shape end-to-end QoS with respect to the performance criterion (3.3) when compared to per-flow control. That is, the minimum value of (3.3) achieved by optimal per-hop control is smaller than that of optimal aggregate-flow control.

An *aggregate-flow per-hop control with parameter* (m, L) is a function

$$\Phi_{m,L} : (\boldsymbol{\eta}, \boldsymbol{\lambda}) \mapsto (\xi, \boldsymbol{\alpha}) \tag{3.6}$$

where $\xi : [1, L] \to [1, m]$ is the *classifier* and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ is the vector of service weights assigned to the m service classes. With respect to end users, $\Phi_{m,L}$ induces—explicitly or implicitly—a performance function $\varphi^i_{m,L}$ for each user $i \in [1, n]$, $\varphi^i_{m,L} : (\boldsymbol{\eta}, \boldsymbol{\lambda}) \mapsto \alpha_i$, where $\alpha_i = \varphi^i_{m,L}(\boldsymbol{\eta}, \boldsymbol{\lambda}) \geq 0$ is user i's share of the bandwidth allocated by $\Phi_{m,L}$. With a slight abuse of notation, we use α_i to denote both user i's ($i \in [1, n]$) apportioned resource, as well as the service weight allocated by $\Phi_{m,L}$ to service class i ($i \in [1, m]$). In the per-flow case, they coincide.

Consider a special type of aggregate-flow per-hop control $\Phi_{m,L}$—called *Reduction Classifier*—whose behavior is completely determined by its classifier $\xi : [1, L] \to [1, m]$, in the following sense. Let $S_k = \{i \in [1, L] : \ \xi(i) = k\}$, $k \in [1, m]$, denote the partition of $[1, L]$ induced by ξ. Then $\Phi_{m,L}$ is specified by the following procedure:

<u>$\Phi_{m,L}(\boldsymbol{\eta}, \boldsymbol{\lambda})$:</u>

1. Compute $\lambda^k = \sum_{i \in S_k} \lambda_i$ for each $k \in [1, m]$.
2. Compute $\hat{\eta}^k$ for $k \in [1, m]$ as follows,

$$\hat{\eta}^k = \begin{cases} 0, & \text{if } \exists i \in S_k,\, \hat{\eta}_i = 0; \\ 1, & \text{if } \exists i \in S_k,\, \hat{\eta}_i = 1; \\ \sum_{i \in S_k} \hat{\eta}_i / |S_k|, & \text{otherwise.} \end{cases}$$

3. Use per-flow optimal solution (Proposition 3.4) with new input $\tilde{\boldsymbol{\eta}} = (\hat{\eta}^1, \ldots, \hat{\eta}^m)$, $\tilde{\boldsymbol{\lambda}} = (\lambda^1, \ldots, \lambda^m)$, to solve the reduced per-flow classifier problem consisting of m superusers.

A reduction classifier reduces the L label (or n user) problem to an m user per-flow classification problem by aggregation of component flows and centroid computation, then solves the reduced problem by applying the optimal per-flow classification solution.

Theorem 3.7 *Let $\Phi_{m,L}$ be a reduction classifier represented by its classifier ξ. Then $\Phi_{m,L}$ is an* optimal aggregate-flow per-hop control, *i.e., satisfies* (3.3) *if,* *and only if, ξ is a solution to*

$$\min_{\xi'} \sum_{k\in[1,m]} \sum_{i\in S_k} (\hat{\eta}_i - \hat{\eta}^k)^2 \tag{3.8}$$

where the minimum ranges over all reduction classifiers ξ'.

Theorem 3.7 shows that an optimal aggregate-flow classifier must efficiently cover—in the mean-square sense—the set of label values $\{\hat{\eta}_1, \hat{\eta}_2, \dots, \hat{\eta}_n\}$ using m centroids $\{\hat{\eta}^1, \dots, \hat{\eta}^m\}$. Aggregate-flow per-hop control is, mathematically, an optimal clustering problem. Unlike its many brethren in higher dimensions that are NP-complete [11], the clustering problem given by (3.8) in Theorem 3.7 has a poly-time algorithm; e.g., it can be solved by dynamic programming. When $L = m$, the practically relevant case where there are as many labels as service classes, optimal aggregate-flow classification has a *linear time* algorithm.

3.3 Properties of Optimal Aggregate-flow Classifier

To satisfy user i's QoS requirement, the per-hop control must apportion a fraction $\alpha_i^* \geq 0$ of the available bandwidth. Let α_i^* denote the minimal such bandwidth. We will use $\varphi^i(\cdot)$ to denote the performance function corresponding to $x^i(\cdot)$ which allocates—explicitly or implicitly—a service weight to user i for a given input $\boldsymbol{\eta}$. For user i, let $\mathcal{A}_i = \{\boldsymbol{\eta} : \varphi^i(\boldsymbol{\eta}) \geq \alpha_i^*\}$. Thus $\mathcal{A}_i$ represents the set of configuration where user i's QoS requirement is satisfied. Let $\mathcal{A}^* = \bigcap_{i=1}^n \mathcal{A}_i$. A configuration $\boldsymbol{\eta}$ is *system optimal* if $\boldsymbol{\eta} \in \mathcal{A}^*$, and for all $\boldsymbol{\eta}' \neq \boldsymbol{\eta}$, $\boldsymbol{\varphi}(\boldsymbol{\eta}') > \boldsymbol{\varphi}(\boldsymbol{\eta})$ does not hold. In a system optimal configuration, the users' QoS requirements are met while expending the minimal amount of resources.

Theorem 3.9 *An optimal aggregate-flow per-hop control with parameters $L = m$ satisfies properties* (A1), (A2), *and* (B).

The $L = m$ constraint advanced by Theorem 3.9 coincides with practical considerations that derive from an implementation perspective. For example, assuming four bits from the TOS field in IPv4 are used to encode the label set $\{1, 2, \dots, 16\}$, then we may configure 16 service classes at routers, one for each of the 16 possible label values.

Theorem 3.10 *Let $L = m < n$. $\mathcal{A}^* \neq \emptyset$ if there exists $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ with $\alpha_{min} = 0$, $\alpha_{max} = 1$, $0 \leq \alpha_i \leq 1$, such that for all $i \in [1, n]$,*

$$(1-\nu)\frac{\lambda_i \alpha_i}{\sum_{j=1}^n \lambda_j \alpha_j} + \nu \frac{\lambda_i}{\sum_{j=1}^n \lambda_j} \geq \alpha_i^* + \frac{1-\nu}{L-1}\frac{\lambda_i}{\sum_{j=1}^n \lambda_j \alpha_j}. \tag{3.11}$$

The left-hand-side of inequality (3.11) denotes a valid service weight vector with respect to the optimal aggregate-flow classifier. The second term in the right-hand-side of (3.11) quantifies the loss of power due to coarsification. If $L \to \infty$, then the term drops out. In practice, L is a small finite value (e.g., using 4 bits in the precedence field of IP, $L = 16$). The next result shows that when $n \gg L$ (the raison d'etre of aggregate-flow control) relation (3.11) can be tight even for small L.

Corollary 3.12 *Under the same conditions as* Theorem 3.10, *let* $d_i = \lfloor (L-1)\alpha_i \rfloor$, $i \in [1, n]$. *Then,* $\mathcal{A}^* \neq \emptyset$ *if for all* $i \in [1, n]$,

$$(1-\nu)\,\frac{\lambda_i \alpha_i}{\sum_{j=1}^{n} \lambda_j \alpha_j} + \nu\,\frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j} \geq \alpha_i^* + (1-\nu)\frac{\lambda_i}{\sum_{k=1}^{L-1} k \sum_{j:d_j=k} \lambda_j}. \quad (3.13)$$

For $n \gg L$, we can expect $\frac{\lambda_i}{\sum_{k=1}^{L-1} k \sum_{j:d_j=k} \lambda_j} \ll 1$, and (3.11) gives a tight bound on the existence condition of system optimal configurations. We remark that the normalization $\hat{\eta}_i = \eta_i / L$ is able to achieve a higher efficiency with respect to $\mathcal{A}^*$. The other results hold as well. The optimal efficiency problem, and its relation to the present semantics of optimal aggregate-flow per-hop control, is studied in a separate paper.

4 Performance Results

4.1 Set-up

We study the dynamical properties of scalar QoS control using simulation when the η values are controlled by selfish users at the edge. We use the LBNL Network Simulator *ns* (version 2) as our simulation platform. We have modified *ns* to incorporate various per-hop and end-to-end control mechanisms. *QSim* [5] is a *ns*-based WAN QoS simulation environment used in the performance evaluation experiments. We have benchmarked a number of network topologies including vBNS. Figure 4.1 (left) shows a 4-hop topology that emulates the physical topology of Purdue Infobahn, a private IP-over-SONET QoS testbed comprised of Cisco 7206 VXR routers shown in Figure 4.1 (middle) and (right). Implementation of optimal per-hop behavior in IOS with cooperation from Cisco is work-in-progress.

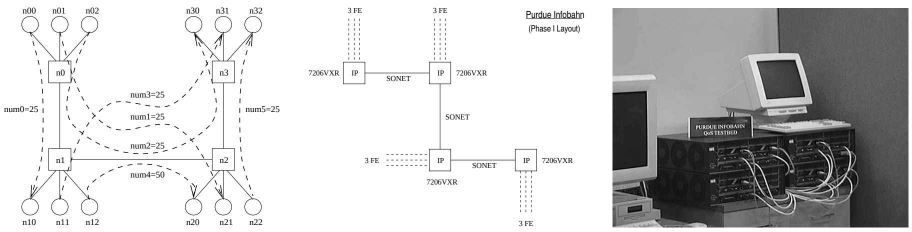

Fig. 4.1. Left: Flow configuration of a 4-hop benchmark topology. Middle: Logical layout of Purdue Infobahn QoS testbed. Right: Physical layout of Purdue Infobahn IP-over-SONET backbone comprised of four Cisco 7206 VXR routers.

4.2 Structural Properties

Figure 4.2 demonstrates the properties (A1), (A2), and (B) satisfied by the optimal aggregate-flow per-hop control which are critical to the facilitation of effective QoS provisioning. The left figure shows packet loss rate across $m = 16$ service classes for fixed label values $\eta = 1, 2, \ldots, 16$, for varying contention levels. We observe that service class separation—property (B)—is satisfied across different contention levels. Figure 4.2 (middle) shows the effect of varying η for a fixed user (here flow 16) on the QoS received by all other users (i.e., groups of users over 16 service classes). That is, the graphs demonstrate $\partial x^i / \partial \eta_{16}$ for $i = 1, 2, \ldots, 16$. We observe that properties (A1) and (A2) hold for the optimal aggregate-flow per-hop control. Figure 4.2 (right) shows QoS separation across 16 service classes for delay. In general, for properties (A1) and (A2) to hold for delay, GPS must be augmented with buffer control.

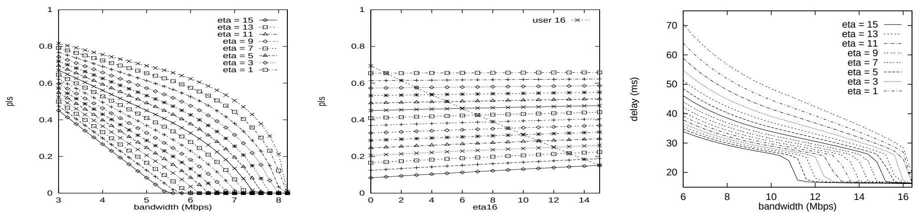

Fig. 4.2. Manifestation of properties (A1), (A2), and (B). Left: Service class QoS separation (packet loss rate) as a function of bottleneck bandwidth (property (B)). Middle: End-to-end QoS as a function of $\eta_1 6$ which is decreased from 1 to 16 (properties (A1) and (A2)). Right: QoS separation for delay.

4.3 Adaptive Label Control

Dynamics Figure 4.3 shows the QoS shaping dynamics of a $n = 25$ user, $L = 16$ label, and $m = 16$ service class configuration for a 2-hop configuration with 7 groups of users with packet loss rate requirements 0.1 (users 0, 1), 0.15 (users 2, 3, 4), 0.2 (users 5, 6, 7), 0.3 (users 8, 9, 10), 0.4 (users 11–14), 0.6 (users 14–18), and best-effort (users 19–24). The latter's η value is fixed at 1. Figure 4.3 (top) shows the packet loss trace for users 0, 5, and 11–14 whose QoS requirements are satisfied. The bottom plots show the corresponding η traces. We observe that adaptive label control is able to achieve the target QoS in a distributed and end-to-end manner over the optimal aggregate-flow per-hop control network substrate. In particular, Figure 4.3 (right) shows the η value trace of users 11, 12, 13, and 14 who happen to possess the same QoS requirement (packet loss rate 0.4). We observe that they converge to the same service class as determined by η. Similar results hold for end-to-end delay.

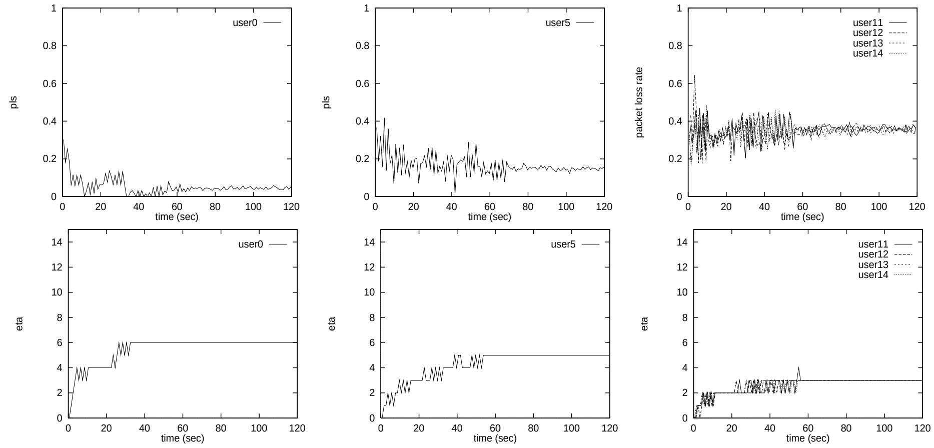

Fig. 4.3. Individual flow dynamics. Top: Packet loss trace for users with QoS requirements 0.1, 0.2, and 0.4. Bottom: Corresponding η label value trace.

End-to-end QoS Provisioning Figure 4.4 (left) shows QoS achieved using adaptive label control over the aggregate-flow per-hop behavior substrate for $n = 25$, $L = m = 16$, and user QoS requirements 0.01, 0.05, 0.1, 0.15, 0.2, 0.3, and 0.4. For small bandwidth, $\mathcal{A}^*$ is empty and no solution exists that satisfies all user requirements. As bandwidth is increased, adaptive label control achieves both QoS differentiation and user-specified QoS shaping. Figure 4.4 (right) shows a 10-fold larger system comprised of $n = 250$ users, $L = m = 16$, and seven levels of QoS requirements. 200 users—spanning all seven QoS requirement levels—are VBR sources (Poisson) and 50 users (also spanning all seven levels) are CBR sources. We observe that when resources are sufficient and $\mathcal{A}^* \neq \emptyset$, then adaptive label control—in conjunction with optimal per-hop behavior—is able to differentiate as well as affect end-to-end QoS commensurate with user requirements.

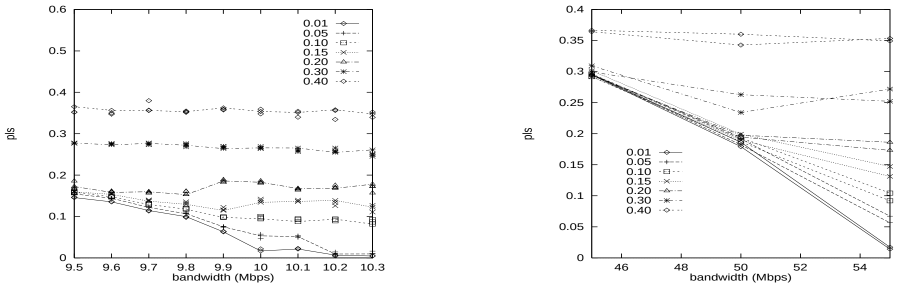

Fig. 4.4. Left: Packet loss rate achieved as a function of bottleneck bandwidth for $n = 25$, $L = m = 16$, and user population with seven QoS requirements. Right: Packet loss rate achieved for $n = 250$, $L = m = 16$, with seven QoS requirements.

References

1. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated service. RFC 2475, 1998.
2. S. Chen and K. Park. A distributed protocol for multi-class QoS provision in noncooperative many-switch systems. In *Proc. IEEE International Conference on Network Protocols*, pages 98–107, 1998.
3. S. Chen and K. Park. An architecture for noncooperative QoS provision in many-switch systems. In *Proc. IEEE INFOCOM '99*, pages 864–872, 1999.
4. S. Chen, K. Park, and M. Sitharam. On the ordering properties of GPS routers for multi-class QoS provision. In *Proc. SPIE International Conference on Performance and Control of Network Systems*, pages 252–265, 1998.
5. Shaogang Chen. *Stratified Best-effort QoS Provisioning in Noncooperative Networks.* PhD thesis, Purdue University, 2000.
6. D. Clark and W. Fang. Explicit allocation of best-effort packet delivery service. *IEEE/ACM Trans. Networking*, 6(4):362–373, 1998.
7. R. L. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE J. Select. Areas Commun.*, 13(6):1048–1056, 1995.
8. G. de Veciana, G. Kesidis, and J. Walrand. Resource management in wide-area ATM networks using effective bandwidths. *IEEE J. Select. Areas Commun.*, 13(6):1081–1090, 1995.
9. C. Dovrolis, D. Stiliadis, and P. Ramanathan. Proportional differentiated services: Delay differentiation and packet scheduling. In *Proc. ACM SIGCOMM '99*, 1999.
10. W. Feng, D. Kandlur, D. Saha, and K. Shin. Adaptive packet marking for providing differentiated services in the Internet. In *Proc. IEEE International Conference on Network Protocols*, pages 108–117, 1998.
11. M. Garey and D. Johnson. *Computers and Intractability.* W. H. Freeman and Company, 1979.
12. J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. RFC 2597, 1999.
13. V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. RFC 2598, 1999.
14. J. MacKie-Mason and H. Varian. Economic FAQs about the Internet. In L. McKnight and J. Bailey, editors, *Internet Economics*, pages 27–63. MIT Press, 1996.
15. M. May, J. Bolot, A. Jean-Marie, and C. Diot. Simple performance models of differentiated services schemes for the Internet. In *Proc. IEEE INFOCOM '99*, pages 1385–1394, 1999.
16. Andrew Odlyzko. Paris Metro Pricing: The minimalist differentiated services solution. In *Proc. IEEE/IFIP International Workshop on Quality of Service*, 1999.
17. A. Parekh and R. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM Trans. Networking*, 2(2):137–150, 1994.
18. K. Park, M. Sitharam, and S. Chen. Quality of service provision in noncooperative networks: heterogeneous preferences, multi-dimensional QoS vectors, and burstiness. In *Proc. 1st International Conference on Information and Computation Economies*, pages 111–127, 1998.
19. Kihong Park. Warp control: a dynamically stable congestion protocol and its analysis. In *Proc. ACM SIGCOMM '93*, pages 137–147, 1993.
20. H. Ren and K. Park. Toward a theory of differentiated services. In *Proc. IEEE/IFIP International Workshop on Quality of Service*, pages 211–220, 2000.