

Providing Per-Link Minimum Throughput Guarantees in the Internet

Srisankar S. Kunniyur

Abstract

Traditional Diff-Serv approaches have relied on multiple queues and scheduling mechanisms to provide traffic classes throughput guarantees at links. In this paper¹ we describe a framework to provide throughput guarantees using only a First-In-First-Out (FIFO) queue. We assume that each link can support K classes of traffic. Each link guarantees each class at least a certain fixed fraction of its capacity. We first formulate the problem of providing throughput guarantees as a multi-stage convex optimization problem. We then present a decentralized, scalable solution to this formulation in terms of congestion-controllers at the sources and adaptation algorithms at the links. A possible implementation at the routers is then presented followed by packet level simulations. We then study the stability of the proposed scheme in a general network topology.

I. INTRODUCTION

An explosive growth in business applications using the Internet have resulted in a strong demand for some notion of reliability or quality of service during periods of congestion. During periods of congestion or failure, the quality of service of all flows is degraded. As a result, a strong need for service differentiation in the Internet is felt both by the service providers as well as the customers. Service differentiation helps to protect certain classes of traffic during periods of congestion. Protection of these classes can be in terms of assuring maximum delay guarantees and/or assuring minimum throughput guarantees. For example, a service provider might want to provide two classes of traffic (Class 1 and Class 2) with Class 1 traffic guaranteed a minimum throughput during congestion. As a result, the customer can distribute their traffic among these two classes depending on the priority of the class as well as their budget. For example, a banking firm might like to ensure that in times of congestion, preference is given to bank transaction data. So, the bank might label all their transaction data as Class 1 traffic.

Another example which exemplifies the need for service differentiation arises in a situation of failure. Let us assume there are two classes of traffic in the network (e-commerce traffic and non-commercial traffic). When a link goes down, the IP routing architecture reroutes the traffic on that link. This might cause congestion and losses in other parts of the network. It might be impractical to over-provision the network to account for these failures in the network. As a result, the network might just want to protect the e-commerce traffic in such scenarios by guaranteeing a minimum throughput for such classes of traffic during periods of congestion caused by failure.

To provide quality of service (QoS), the Internet Engineering task Force (IETF) has proposed the Integrated services (IntServ) model (cf. Braden *et al.* [4]) and the Differentiated services (Diff-Serv) model (cf. Bernet *et al.* [2]). The Intserv model provides per-flow QoS guarantees, but does

¹A shorter version of the paper was presented at the Annual Allerton Conference, 2003

not scale well with the number of users. The Diff-Serv model on the other hand provides per-hop behavior based on aggregates (or classes) at the core routers and hence scales well at the core routers. Per flow policing and shaping is done at the edge routers.

Assured forwarding for Diff-Serv was first proposed by Clark and Fang in [6] to provide an “expected capacity” to flows (or classes) specified by the traffic profile. The basic idea was that at the edges, packets are classified as *In* if the traffic conforms to the traffic profile negotiated or *Out* if the traffic violates the negotiated traffic profile. Within the network, the packets are queued using a FIFO queuing discipline and during congestion the router preferentially drops the *Out* packets. RIO (RED with *In* and *Out*) is used at the routers to preferentially drop packets. While RIO and its variants RIO-C, RIO-DC and WRED preferentially drops *Out* packets, the throughput achieved by the *In* packets cannot be determined or set. A service provider would like to set a minimum throughput that can be achieved by the *In* class due to the preferentially marking scheme. Such a flexibility becomes more important when packets can be classified into more than two classes.

Schemes have been proposed in recent literature to create preferential treatment of packets in which packets of different classes are put into different queues. The queues are then serviced according to some scheduling policy to provide a desired quality of service to that class. For example, Weighted Fair Queueing algorithm (cf. Clark, *et al.* [7] and Demers *et al.* [8]), Priority Queueing (cf. Jaiswal [13]) Weighted Round Robin (cf. Katevenis *et al.* [14]) and Class Based Queueing (cf. Floyd and Jacobson [10]) employs multiple queues at the link and can ensure that each class receives at least some portion of the bandwidth. However, separate queues for separate packet types might cause packet reordering leading to performance degradation in TCP and delay jitter in real time traffic.

Our paper proposes a framework that integrates the virtues of having multiple queues at the routers with the RIO scheme. We wish to preserve the simplicity of the RIO scheme at the routers while at the same time ensuring that classes receives at least a certain share of the bandwidth. The main contributions of the paper are: 1) proposing an analytical framework to study the problem using convex optimization techniques, 2) designing a class of marking algorithms at the routers to achieve minimum throughout guarantees, 3) demonstrating the effectiveness of such algorithms using simulations and 4) studying the stability of such schemes in a general network setting with round-trip delays.

A. Related Work

Recent advances in Active Queue Management (AQM) schemes have been successful in providing a low-loss, low-delay service at high utilizations (cf. the adaptive RED scheme by Floyd [9], the Adaptive Virtual Queue (AVQ) scheme by Kunniyur and Srikant [17], the REM scheme by Athuraliya and Low [1] and the PI controller by Hollot *et al.* [12] to name a few). Schemes like AVQ have demonstrated that a low-loss behavior can be obtained for very small buffer sizes. As a result, one can bound the maximum delay experienced by any flow. However the traffic into the link is not differentiated in these schemes and all packets are treated equally. As a result, one cannot guarantee minimum bounds on throughput for a particular traffic class using such schemes. It would be desirable to implement a differentiated services model that provides minimum throughput guarantees at each link to certain classes while maintaining the low-delay and low-loss behavior using only a FIFO queue.

In this paper we integrate low-loss, low-delay AQM schemes with the Diff-Serv framework to provide minimum guaranteed throughputs per class using a FIFO queue. The framework assumes that each link guarantees at least a fixed fraction of its capacity to its priority classes. We also assume that the links provide congestion feedback to the sources in the form of ECN marks² and that all the sources react to congestion notification from the routers (elastic sources). The structure of the congestion-controllers at the sources can be arbitrary and are modeled using utility functions as in [15]. We use multiple marking functions at a link to guarantee a minimum throughput for each traffic class. The use of multiple marking functions per link in the context of pricing was proposed by Gibbens and Kelly in [11]. Each link has two logical resources with the higher priority traffic using both the logical resources and the lower priority traffic using only one logical resource. As a result, more packets of the high priority traffic are marked compared to the lower priority traffic. This translates into a higher price for the high priority traffic. Due to the pricing implication and the ability of users to adapt, no relative throughputs for different traffic classes need to be set. But the total utilization of the link as well as the utilization of a class is a function of the prices that is set at the link and hence is variable. *In this paper we consider a system in which each class is guaranteed a minimum throughput at each link.* We assume that pricing for each class is already set by the service providers. Due to the minimum throughput restriction, traffic in a class with a guaranteed minimum throughput cannot receive more marks than the traffic in non-guaranteed throughput class. One can also adapt this framework to apply in the context of pricing as in [11]. We also consider a system model that is different from the system model considered in [11].

Recent work by Chait *et al.* [5] considers a two-level marking scheme for providing throughput differentiation. However, the model uses token buckets at edge-routers to guarantee minimum throughputs and assumes that the core routers are able to deliver the guarantees. As a result, the edge routers are responsible for setting the parameters of the token buckets and adapting to changing network load. A similar idea of traffic engineering at the edge routers for traffic classes in which each class is guaranteed a minimum throughput from edge-to-edge is discussed in [19]. **In this paper, we consider a different mode of operation in which the throughput guarantees are provided at each link by guaranteeing each priority class at least a certain fixed fraction of the capacity of the link.** As a result, rerouting due to core failures and protection of certain classes at the core due to such failures can be easily handled using the framework presented in the paper. In addition, the framework can be easily integrated in the AF framework of the Diff-Serv model.

The rest of the paper is organized as follows. We first describe the system model in Section II and formulate the problem of providing a low-loss, low-delay service with minimum throughput guarantees as a convex optimization problem. We then propose a decentralized algorithm to solve the convex optimization problem in Section II-A and provide a possible implementation of the algorithm at the routers in Section III. Detailed simulations using the proposed implementation are presented in Section IV. We then study the stability of such schemes in the presence of round-trip delays by conducting a local stability analysis in Section V. We conclude with discussions and future work in Section VI.

²The results presented in the paper segue smoothly with some modifications when dropping instead of marking is employed at the routers.

II. SYSTEM MODEL

Consider a network with a set \mathcal{L} of links and let C_l and $\gamma_l \leq 1.0$ be the capacity and the desired utilization respectively of each $l \in \mathcal{L}$. By desired utilization, we refer to the ratio of the maximum arrival rate (that can be supported by the link to guarantee a desired small loss probability) to the link capacity. We assume that there are K classes of traffic. *At each link $l \in \mathcal{L}$, class j is guaranteed at least a fraction η_l^j of the total bandwidth at the link.* We assume that the network is provisioned correctly in the sense that $\sum_{j=1}^{K-1} \eta_l^j < \gamma_l$ for all links. Without loss of generality, we assume that class K has no guaranteed minimum bandwidth at the link and forms the best effort class of the network.

We associate a route with each flow and hence we use the terms flow and route interchangeably throughout this paper. Let \mathcal{R} be the set of all flows in the network and let \mathcal{R}^j be the set of all flows in the network that belong to class j . Denote by \mathcal{R}_l the set of all flows that traverse link l . Denote a flow r in class j by f_r^j . Let $\mathcal{L}_r^j \subset \mathcal{L}$ be the set of links through which flow f_r^j traverses. Let flow f_r^j generate traffic at rate x_r^j . Suppose a rate x_r^j has an utility $U_r^j(x_r^j)$ to flow f_r^j . Assume that the utility functions are strictly concave and monotonically increasing functions of x_r over $x_r \geq 0$ and that $U_r^{j'}(x_r^j) \rightarrow \infty$ as $x_r^j \rightarrow 0$ for all f_r^j . Examples of such utility functions include $\log(x)$ [15] and $\frac{-1}{x}$ [20], [16].

The objective of the network is to provide Classes 1 to $K - 1$ the minimum guaranteed throughput at each link. The network would like to first ensure that the traffic in Classes 1 to $K - 1$ gets the assured bandwidth before routing the best-effort traffic with no guarantees. Such a task requires the network to first route the higher class traffic (Classes 1 to $K - 1$) and then allocate the remaining bandwidth at each link to best-effort users. The remaining bandwidth at each link is now a function of the routing matrix of the higher class traffic. The following example illustrates the model more clearly.

Consider a two link network and two classes of traffic as shown in Figure 1. We have two flows of Class 1 (Flow 0 and Flow 1) and two flows of Class 2 (Flow 2 and Flow 3) as shown in Figure 1. We assume that all flows have a $\log(x)$ utility function. Assume that Class 1 traffic is guaranteed at least a fraction η (say, 0.50) of the capacity at each link. We can see that Link 1 acts as a bottleneck link for Flow 0. Even if the entire link 1 capacity of 1 Mbps is given to Flow 0, it will not be able to realize its guaranteed minimum throughput of 5 Mbps at Link 2. As a result, the remaining bandwidth at link 2 for Class 2 flows is a function of the routing matrix of Class 1 flows and the capacities of the links through which Class 1 flows traverse.

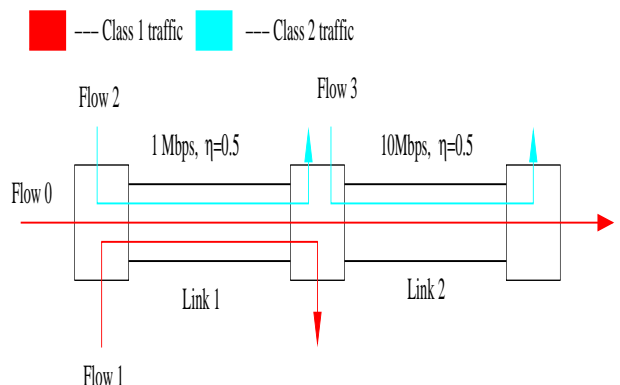


Fig. 1. Example illustrating the system model

In addition to ensuring minimum throughput guarantees, the network has to ensure that flows within the same class are treated fairly. In this paper, we use the utility maximization approach proposed in [15] as a measure of fairness. Such an approach can be used to approximate many notions of fairness like max-min fairness, proportional fairness and minimum potential delay fairness.

We decompose the system³ objective into two steps: In the first step, the system routes the higher class traffic (Class 1 to Class $K - 1$) such that their minimum throughput guarantees are satisfied. For each Class m , $m \in \{1, \dots, K - 1\}$, the step can be written down as:

$$\text{AUX-SYSTEM}(\mathbf{U}^m, \eta^m) \quad (1)$$

$$\begin{aligned} \max \quad & \sum_{f_r^m \in \mathcal{R}^m} U_r^m(y_r^m) \\ \text{subject to} \quad & \sum_{f_r^m: l \in \mathcal{L}_r^m} y_r^m \leq \eta_l^m C_l \quad \forall l \in \mathcal{L} \\ & y_r^m \geq 0 \quad \forall f_r^m \in \mathcal{R}^m. \end{aligned} \quad (2)$$

The system solves the above convex problem for each high priority class. Note that we have a \leq constraint in (2) instead of a \geq sign. Each class (except the best effort traffic class) solves problem (1) which is similar to the system problem defined in [15]. For each $j \in \{0, K - 1\}$, let \hat{y}_r^j solve the above AUX-SYSTEM problem. In the second step, the system tries to allocate the remaining bandwidth at each link to all the users. This step can be written as:

$$\text{SYSTEM}(\mathbf{U}, \gamma) \quad (3)$$

$$\max \sum_{j=1}^{K-1} \sum_{f_r^j \in \mathcal{R}^j} U_r^j(x_r^j + \hat{y}_r^j) + \sum_{f_r^K \in \mathcal{R}^K} U_r^K(x_r^K) \quad (4)$$

subject to

$$\sum_{j=1}^{K-1} \sum_{f_r^j: l \in \mathcal{L}_r^j} (x_r^j + \hat{y}_r^j) + \sum_{f_r^K: l \in \mathcal{L}_r^K} x_r^K \leq \gamma_l C_l, \quad \forall l \in \mathcal{L} \quad (5)$$

$$x_r^j \geq 0 \quad \forall f_r^j \in \mathcal{R}, \quad j \in \{1, \dots, K\}. \quad (6)$$

Since flows in classes 1 to $K - 1$ have already been allocated a flow rate equal to $\{\hat{y}_r^j\}$ (obtained by solving their respective AUX-SYSTEM problem), their utility functions are modified as shown in the objective function in (4). Since $U_r^{K'}(x_r^K) \rightarrow \infty$ as $x_r^K \rightarrow 0$, this forces the network to allocate bandwidth to Class K traffic and prevent starvation of Class K traffic. In other words, the high marginal utility of Class K flows forces the network to allocate some bandwidth to Class K flows. The capacity constraint of the system is given in (5). Note that the capacity constraint is modified to take into account the rates $\{\hat{y}_r^j\}$ that has been allocated to the higher priority traffic. The motivation behind such a framework is that once the minimum throughput guarantees are satisfied, the system objective is to maximize the total utility of all the users subject to capacity constraints and non-negativity constraints. Note that the non-negativity constraints for flows in Classes 1 to $K - 1$ are required to prevent the network from reassigning any bandwidth that was allocated in the AUX-SYSTEM problem. We now show that there exists a unique equilibrium to the problem that the network tries to solve.

Theorem II.1 *There exists a unique solution to the SYSTEM problem defined in (3).*

Proof: Since the utility functions are strictly concave and the constraints are linear, both the system problem in (3) and the auxiliary system problem in (1) have unique solutions. ■

³By system, we refer to both the users (or flows) and the network [15].

The system model shown in (3) is a multi-stage optimization problem in which the solution to the AUX-SYSTEM problem for classes 1 to $K - 1$ is used to solve the SYSTEM problem. A centralized solution to the problem is not feasible in the current Internet. Moreover, with flows arriving and departing, a decentralized solution is necessary for such a model to be successful in the Internet. In the next section we show that the proposed system model in (3) can indeed be solved in a decentralized fashion.

A. Decentralized solution

Denote by λ_l the total flow into link l and by λ_l^j the total Class j flow into link l . Let each link l in the network generate feedback in the form of Explicit Congestion Notification (ECN) marks. Let $p_l(\lambda_l, \tilde{C}_l)$ be the fraction of packets that are marked by the link. As in AVQ [16], we can think of $\{\tilde{C}_l\}$ as the parameter that detects incipient congestion in the links. Hence, the fraction of Class K packets that are marked is given by $p_l(\lambda_l, \tilde{C}_l)$. However in order to provide a minimum throughput guarantee for a Class j flow, the congestion feedback is modified by a function which depends upon the total Class j arrivals as well as a parameter that determines the utilization of Class j flows. Let the fraction of Class j packets that are marked at link l be given by $\max\{0, p_l(\lambda_l, \tilde{C}_l) - q_l^j(\lambda_l^j, \tilde{C}_l^j)\}$, where $q_l^j(\lambda_l^j, \tilde{C}_l^j)$ is the weighting function and \tilde{C}_l^j is the parameter that determines the throughput of class j flows. Since in the current Internet only negative feedback is provided, the $\max(0, z)$ function is employed. We will make the following assumptions regarding the marking functions at each link.

Assumption 1 *The marking functions $q_l^j(u, v)$, $p_l(u, v)$, $u \geq 0, v \geq 0$ and $1 \leq j \leq K - 1$, at each link are assumed to satisfy the following conditions⁴:*

- (i) $q_l^j(u, v) \geq 0$ and $p_l(u, v) \geq 0$ for all u, v
- (ii) $q_l^j(u, v)$ is strictly decreasing in u and strictly increasing in v
- (iii) $q_l^j(u, 0) = 0$.
- (iv) $p_l(u, v)$ is strictly increasing in u and strictly decreasing in v

We assume no feedback delays in this section to describe the decentralized solution. Let each flow f_r^j employ the congestion-control algorithm given by

$$\dot{x}_r^j = \kappa_r^j \left[1 - \frac{1}{U_r^j(x_r^j)} \sum_{l \in \mathcal{L}_r^j} (p_l(\lambda_l, \tilde{C}_l) - q_l^j(\lambda_l^j, \tilde{C}_l^j))^+ \right], \quad (7)$$

where κ_r^j determines the speed of the congestion-controller and the function $(z)^+$ denotes the $\max(0, z)$ function. For Class K , the feedback from link l reduces to $p_l(\lambda_l, \tilde{C}_l)$ since $q_l^K(u, v) = 0$ for all u and v (since Class K has no throughput guarantees). If we assume $U_r(x_r) = \frac{-1}{x}$, we get the familiar TCP congestion controller.

Each link updates \tilde{C}_l and \tilde{C}_l^j using the following algorithm:

$$\dot{\tilde{C}}_l = \begin{cases} \alpha_l(\gamma_l \tilde{C}_l - \lambda_l) & \tilde{C}_l > 0 \\ \max[0, \alpha_l(\gamma_l \tilde{C}_l - \lambda_l)] & \tilde{C}_l = 0 \end{cases}, \quad (8)$$

$$\dot{\tilde{C}}_l^j = \begin{cases} \alpha_l^j(\eta_l^j \tilde{C}_l^j - \lambda_l^j) & \tilde{C}_l^j > 0 \\ \max[0, \alpha_l^j(\eta_l^j \tilde{C}_l^j - \lambda_l^j)] & \tilde{C}_l^j = 0 \end{cases}, \quad j \in [1, K - 1], \quad (9)$$

⁴Such assumptions on marking functions are common and are satisfied by a huge class of marking functions.

where α_l and α_l^j are the step-sizes that determine the speed of adaptation at link l . Note that the routers can employ any arbitrary marking function (p and q) to mark packets. The adaptation at the link is the critical feature. When the total arrival rate (λ_l) is greater than the desired utilization of the link ($\gamma_l C_l$), \tilde{C}_l decreases which increases the marking rate $p(\lambda_l, \tilde{C}_l)$. Similarly, when the total arrival rate is less than the desired utilization, \tilde{C}_l increases which decreases the marking rate $p(\lambda_l, \tilde{C}_l)$. Hence at equilibrium the total arrival rate equals the desired utilization at the link. In addition, if the arrival rate of Class j flows (λ_l^j) falls below the guaranteed throughput at the link ($\eta_l^j C_l$), \tilde{C}_l^j increases which increases $q_l^j(\lambda_l^j, \tilde{C}_l^j)$, which in turn reduces the marking rate for Class j flows. This in turn increases the arrival rate of Class j flows into the link. If at the equilibrium point the total arrival rate of Class j flows at link l is below their minimum guaranteed throughput, then all flows of Class j are either bottlenecked at a different link or has achieved the maximum possible rate (say, due to restrictions on the maximum congestion window size or receiver window size). In this scenario the equilibrium marking rate for Class j flows is zero (i.e., $(p_l - q_l^j)^+ = 0$)⁵. Hence no packets of Class j flows are marked by link l . However, if at the equilibrium point the total arrival rate for Class j flows at link l is greater than the minimum guaranteed throughput for Class j , the equilibrium value of q_l^j is zero. Hence the fraction of Class j packets marked is equal to the fraction of Class K packets marked and is equal to $p_l(\lambda_l, \tilde{C}_l)$. This scenario represents a situation when a normal AQM scheme would lead to a Class j throughput that is greater than the guaranteed throughput. That is, when the level of congestion is not high enough to reduce the throughput of a class below its guaranteed minimum throughput the proposed link adaptation algorithms behave like the AVQ scheme.

In equilibrium, the minimum aggregate rate that is offered to Class j at link l is at least $\eta_l^j C_l$ provided all Class j flows are not bottlenecked elsewhere. Once the minimum throughput guarantees are satisfied, the number of congestion marks that Class j flows receive is proportional to their rate which discourages them from starving the Class K flows. In the event a particular class is unable to use its guaranteed throughput, the unused bandwidth is shared among all users. The congestion controllers determine how the bandwidth is shared by the users. The system comprising the congestion-controllers and the adaptation algorithms at the link can be described by the set of the differential equations given by (7), (8) and (9).

We now state the main result of this section.

Theorem II.2 *The system of congestion-controllers and link adaptation algorithms in (7)-(9) solves the SYSTEM problem in (3). ■*

The above result shows that the invariant point of the system of differential equations describing the congestion-controllers and the link adaptation algorithms solves the proposed SYSTEM problem. Note that the congestion-controllers and the link adaptation algorithms are decentralized in nature and depend only on local parameters. Before proving this result, we first characterize the invariant points of the system of differential equations given by (7)-(9).

⁵See Simulation 2 in Section IV

Definition II.1 A point $(\{x_r^{j*}\}, \{\tilde{C}_l^*\}, \{\tilde{C}_l^{j*}\})$ is an invariant point if

$$\begin{aligned} \frac{dx_r^j}{dt} \Big|_{x_r^j=x_r^{j*}} &= 0 \quad \forall f_r^j \in \mathcal{R} \\ \frac{d\tilde{C}_l}{dt} \Big|_{\tilde{C}_l=\tilde{C}_l^*} &= \begin{cases} 0 & \forall \{l : \tilde{C}_l^* < \infty\} \\ > 0 & \forall \{l : \tilde{C}_l^* = \infty\} \end{cases} \\ \frac{d\tilde{C}_l^j}{dt} \Big|_{\tilde{C}_l^j=\tilde{C}_l^{j*}} &= \begin{cases} 0 & \forall \{l : \tilde{C}_l^{j*} < \infty\} \\ > 0 & \forall \{l : \tilde{C}_l^{j*} = \infty\} \\ < 0 & \forall \{l : \tilde{C}_l^{j*} = 0\} \end{cases}. \end{aligned}$$

■

Lemma II.1 The virtual capacities $(\{\tilde{C}_l^*\}, \{\tilde{C}_l^{j*}\})$ and the arrival rates $(\{\lambda_l^*\}, \{\lambda_l^{j*}\})$ at the invariant points of the system given in (7)-(9) satisfy the following conditions:

$$\begin{aligned} \lambda_l^* &= \gamma_l C_l \quad \forall l \in \{j \in \mathcal{L} : \tilde{C}_l^* < \infty\} \\ \lambda_l^* &< \gamma_l C_l \quad \forall l \in \{j \in \mathcal{L} : \tilde{C}_l^* = \infty\} \\ \lambda_l^{j*} &= \eta_l^j C_l \quad \forall l \in \{j \in \mathcal{L} : \tilde{C}_l^{j*} < \infty\} \\ \lambda_l^{j*} &< \eta_l^j C_l \quad \forall l \in \{j \in \mathcal{L} : \tilde{C}_l^{j*} = \infty\} \\ \lambda_l^{j*} &> \eta_l^j C_l \quad \forall l \in \{j \in \mathcal{L} : \tilde{C}_l^{j*} = 0\}. \end{aligned}$$

■

The Lagrangian of the auxiliary SYSTEM problem in (1) is given by:

$$\mathbb{H}^j(y_r^m, \nu_l^m) = \max_{f_r^m} \sum_{f_r^m} U_r^m(y_r^m) - \sum_{l \in \mathcal{L}} \nu_l^m \left(\sum_{f_r^m \in \mathcal{R}_l} y_r^m - \eta_l^m C_l \right), \quad (10)$$

where $\nu^m = (\nu_l^m : l \in \mathcal{L})$ is the Lagrange multiplier. Therefore, the optimal solution $\{\hat{y}_r^m\}$ for the auxiliary SYSTEM problem in (1) satisfies the KKT conditions [3]:

$$U_r^{m'}(\hat{y}_r^m) - \sum_{l:l \in f_r^m} \nu_l^m = 0 \quad \forall f_r^m \in \mathcal{R}^m \quad (11)$$

$$\nu_l^m = 0 \quad \text{if} \quad \sum_{f_r^m \in \mathcal{L}_r} \hat{y}_r^m < \eta_l^m C_l \quad (12)$$

$$\nu_l^m > 0 \quad \text{if} \quad \sum_{f_r^m \in \mathcal{L}_r} \hat{y}_r^m = \eta_l^m C_l. \quad (13)$$

Now, denote the Lagrangian of the SYSTEM problem in (3) by $\mathbb{J}(x_r^j, \nu_l^j)$. We know

$$\mathbb{J}(x_r^j, \nu_l^j) = \begin{cases} \max_{j=1}^{K-1} \sum_{f_r^j} \sum_{f_r^j} U_r^j(x_r^j + \hat{y}_r^j) + \sum_{f_r^K} U_r^K(x_r^K) + \sum_{j=1}^{K-1} \sum_{f_r^j} \theta_r^j x_r^j \\ - \sum_{l \in \mathcal{L}} \mu_l \sum_{j=1}^{K-1} \sum_{f_r^j \in \mathcal{R}_l} (\hat{y}_r^j + x_r^j) - \sum_{l \in \mathcal{L}} \mu_l \left(\sum_{f_r^K \in \mathcal{R}_l} x_r^K - \gamma_l C_l \right) \end{cases}, \quad (14)$$

where $\mu = (\mu_l : l \in \mathcal{L})$ and $\theta = (\theta_r^j : f_r^j \in \mathcal{R}^j, j \in \{1, 2, \dots, K-1\})$ are the Lagrange multipliers. The optimal solution for the SYSTEM problem in (3) satisfies the KKT conditions [3]:

$$U_r^{K'}(\hat{x}_r^K) - \sum_{l:l \in f_r^K} \mu_l = 0 \quad \forall f_r^K \in \mathcal{R}^K \quad (15)$$

$$U_r^j(\hat{y}_r^j + \hat{z}_r^j) - \sum_{l:l \in f_r^j} \mu_l + \theta_r = 0, \quad \forall f_r^j \in \mathcal{R}^j, j \neq K \quad (16)$$

$$\mu_l = 0 \quad \text{if} \quad \sum_{j=1}^{K-1} \sum_{f_r^j \in \mathcal{L}_r} (\hat{y}_r^j + \hat{z}_r^j) + \sum_{f_r^K \in \mathcal{L}_r} \hat{x}_r^K < \gamma_l C_l \quad (17)$$

$$\mu_l > 0 \quad \text{if} \quad \sum_{j=1}^{K-1} \sum_{f_r^j \in \mathcal{L}_r} (\hat{y}_r^j + \hat{z}_r^j) + \sum_{f_r^K \in \mathcal{L}_r} \hat{x}_r^K = \gamma_l C_l \quad (18)$$

$$\theta_r^j = 0 \quad \text{if} \quad \hat{z}_r^j > 0 \quad (19)$$

$$\theta_r^j = \sum_{l \in f_r^j} (\mu_l - \nu_l^j) \quad \text{if} \quad \hat{z}_r^j = 0. \quad (20)$$

We now restate Theorem II.2 and outline its proof.

Theorem II.3 *The system of congestion-controllers and link adaptation algorithms in (7)-(9) solves the SYSTEM problem in (3).*

Outline of the Proof: From the definition of an invariant point and Lemma II.1, we have for all f_r^j such that $j \neq K$,

$$U_r^{j'}(y_r^{j*}) - \sum_{l:l \in f_r^j} (p_l(\lambda_l^*, \tilde{C}_l^*) - q_l^j(\lambda_l^{j*}, \tilde{C}_l^{j*}))^+ = 0 \quad (21)$$

and

$$U_r^{K'}(x_r^{K*}) - \sum_{l:l \in f_r^K} p_l(\lambda_l^*, \tilde{C}_l^*) = 0 \quad \forall f_r^K \in \mathcal{R}^K. \quad (22)$$

At the invariant points, the following relations also hold.

$$\lambda_l^* < \gamma_l C_l \Rightarrow p_l(\lambda_l^*, \tilde{C}_l^*) = 0 \quad (23)$$

$$\lambda_l^{j*} > \eta_l^j C_l \Rightarrow q_l^j(\lambda_l^{j*}, \tilde{C}_l^{j*}) = 0 \quad (24)$$

$$\lambda_l^{j*} < \eta_l^j C_l \Rightarrow q_l^j(\lambda_l^{j*}, \tilde{C}_l^{j*}) = 1. \quad (25)$$

Substituting $p_l(\lambda_l^*, \tilde{C}_l^*)$ for μ_l and $(p_l(\lambda_l^*, \tilde{C}_l^*) - q_l^j(\lambda_l^{j*}, \tilde{C}_l^{j*}))^+$ for ν_l^j , we can see that the invariant point solves the KKT conditions in (15) - (20). Since the SYSTEM problem has a unique solution, the invariant point solves the SYSTEM problem. \blacksquare

Note that the Lagrange multipliers need not be unique. In the next section we discuss a possible implementation of such algorithms at the routers.

Remarks:

- The utility function of a TCP user is approximately $\frac{-1}{T^2 x}$ where T is the round-trip delay and x is the flow rate of the user [16]. As a result for a TCP user in class j , the congestion-controller in (7) can be written as:

$$\dot{x} = \kappa \left[\frac{1}{T^2} - x(t)x(t) \sum_{l \in \mathcal{L}_r^j} (p_l(\lambda_l, \tilde{C}_l) - q_l^j(\lambda_l^j, \tilde{C}_l^j))^+ \right]. \quad (26)$$

- $x(t) \sum_{l \in \mathcal{L}_r^j} (p_l(\lambda_l, \tilde{C}_l) - q_l^j(\lambda_l^j, \tilde{C}_l^j))^+$ is the fraction of packets marked in the network. Hence, the congestion-controllers at the sources need not be changed.
- Such a scheme can be implemented using only binary feedback from the routers (cf. [1], [17], [12]).

III. IMPLEMENTATION AT THE ROUTERS

In this section we describe a potential way of implementing such class based algorithms at a link. We use the concept of virtual queues and virtual capacity adaptation in [17] to implement such a scheme. One can use any AQM scheme as long as the adaptation parameters and the marking function can be identified. For example, we believe one can modify RED, REM or PI to work in this framework. We call the above implementation the *Class-Based AVQ* algorithm.

An illustration of the class based AVQ scheme with two classes of traffic (high priority and low priority) is shown in Figures 2-5. The high priority traffic is guaranteed a minimum throughput of η times the capacity of the link. The link maintains two virtual queues, a virtual queue for the high priority traffic and a common virtual queue (that is common to both the high priority and low priority traffic). The high priority virtual queue capacity is adapted using (9) to guarantee the minimum throughput for the high priority class and the common virtual queue capacity is adapted using (8) to track the desired utilization of the link.

When a low priority packet arrives at the link, a fictitious packet is enqueued only in the common virtual queue as shown in Figure 2. When a high priority packet arrives, a fictitious packet is enqueued both in the common and high priority virtual queues (see Figure 3). We note that no actual enqueueing and dequeueing of packets is necessary at the virtual queues. We just keep track of the virtual queue lengths. If the incoming packet overflows all the virtual queues to which the fictitious packet is added, it is marked in the real queue and removed from the buffers of the virtual queues. For example, a high priority packet is marked if both the high priority and common virtual queues overflow (see Figure 4). An incoming low priority packet is marked if it makes the common virtual queue overflow (see Figure 5). Note that an incoming low priority packet cannot make the high priority virtual queue overflow since no packet is added to the high priority virtual queue when a low priority packet arrives.

An alternate method to achieve the same result is: for each incoming packet set $\bar{p} = 1$ if the common virtual queue overflows, else set $\bar{p} = 0$. Set $\bar{q}^H = 0$ if the high priority virtual queue overflows, else set $\bar{q}^H = 1$. For each low-priority packet, mark it with the value of \bar{p} . Similarly, for each high-priority packet mark it with the value of $(\bar{p} - \bar{q}^H)^+$. The adaptation of the virtual queues can be implemented using a token bucket scheme as in the AVQ algorithm [17]. Equation (8) can be thought of as a token bucket of size C_l in which tokens arrive at the rate $\alpha_l \gamma_l C_l$ and are drained at a rate of $\alpha_l \lambda_l$ by each arrival. The size of the token bucket is the upper bound on the virtual capacity \tilde{C}_l . Similarly, we can think of Equation (9) as a token bucket of size C_l in which tokens are generated at the rate $\alpha_l^j \eta_l^j C_l$ and are drained at a rate of $\alpha_l \lambda_l^j$ by each Class j arrival. Figure 6 illustrate the adaptation of the virtual capacities. Note that complexity of the class based AVQ algorithm is comparable to the complexity of RIO. RIO also employs multiple virtual queues to preferentially drop packets. The calculation of the average virtual queue size and marking probability in RIO is replaced by the adaptation of the virtual queue in the class based AVQ algorithm.

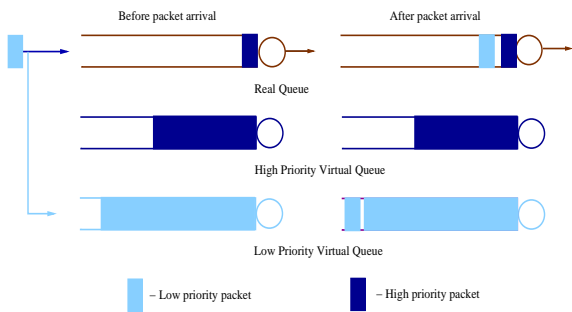


Fig. 2. Class Based AVQ: When a low priority packet arrives

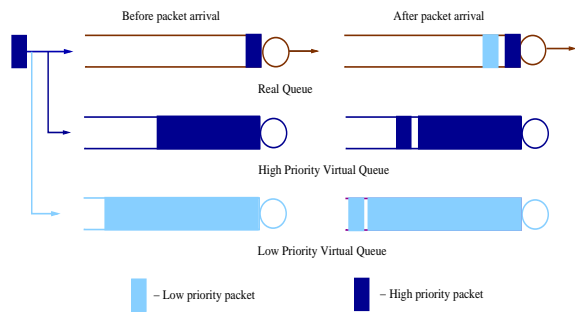


Fig. 3. Class Based AVQ: When a high priority packet arrives

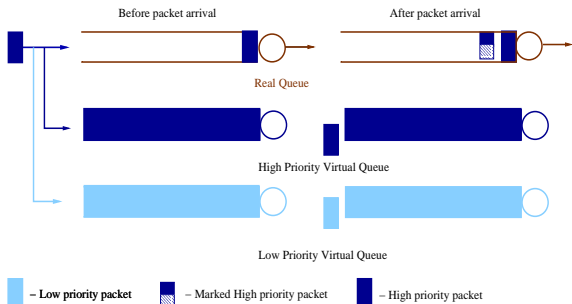


Fig. 4. When the low and high priority queues are full

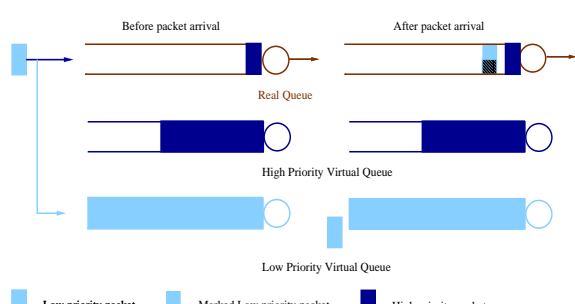


Fig. 5. When the low priority queue is full

IV. SIMULATIONS

We use *ns-2* to carry out a detailed simulation taking into account finite packet sizes, round-trip delay, TCP window-flow control and short flows and is designed to study the effectiveness of the proposed algorithm with packet-level implementations of the TCP congestion controller.

The network used in the simulations is shown in Figure 7. We consider a single bottleneck link of capacity 10 Mbps (link between nodes Core 1 and Core 2 in Figure 7). All other links have a capacity of 100 Mbps. We simulate the class based adaptive virtual queue algorithm on the bottleneck link between nodes core and core2. We let γ , the desired utilization of the bottleneck link, be 0.95. The destination node (*Dest* node in Figure 7) is connected to five edges with round-trip delays of 40ms, 70ms, 100ms, 130ms and 160ms. TCP connections are initiated between the edges and destination. We use TCP-NewReno as the default transport protocol with the TCP data packet size set to 1000 bytes. Each TCP connection is placed in one of three QoS classes: Class 1, Class 2 or Best-effort. Class 1 is assured an utilization of 40% at the link and Class 2 is assured an utilization of 30%. The best-effort class does not have a minimum throughput guarantee at the link. In all the simulations we choose $\alpha = \alpha^j = 0.2$ for all classes.

Simulation 1: In this simulation, we have 20 TCP flows in Class 1, 30 TCP flows in Class 2 and 80 TCP flows in Class 3. The TCP flows are uniformly distributed among all the edges, i.e., 4 Class 1 flows, 6 Class 2 flows and 16 best effort flows are initiated at each edge. The evolution of the total throughput of each class at the bottleneck link is presented in Figure 8. After an initial transient, we can see that Class 1 and Class 2 has an average utilization close to 40% and 30% respectively, as desired. The remaining bandwidth is taken by the best effort TCP connections. We plot the total utilization at the link in Figure 9. We can see that the total utilization at the link hovers around 0.95 as desired. Figures 10 - 12 shows the evolution of the

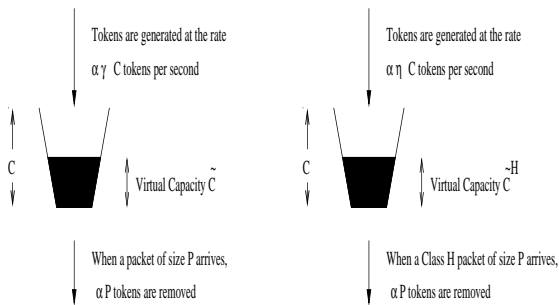


Fig. 6. Token bucket implementation

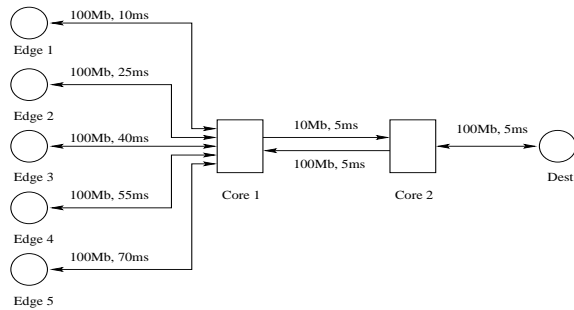


Fig. 7. Network used in simulations

	Edge 1	Edge 2	Edge 3	Edge 4	Edge 5
Class 1 traffic	399 Kbps	228 Kbps	153 Kbps	116 Kbps	97 Kbps
Class 2 traffic	138 Kbps	118 Kbps	96 Kbps	73 Kbps	59 Kbps
Best effort traffic	43 Kbps	30 Kbps	28 Kbps	25 Kbps	25 Kbps

TABLE I

AVERAGE FLOW RATES OF AN USER WHEN MINIMUM THROUGHPUT GUARANTEES ARE EMPLOYED

average flow rate that a typical user gets in each class from Edges 1, 3 and 5. The average is taken over all the users in the class. We can see that Class 1 and Class 2 flows have a higher average flow rate than best effort flows due to the minimum throughput guarantees at the link. Class 1 has a higher average flow rate than Class 2 due to a higher throughput guarantee and smaller number of flows. The fraction of throughput achieved by each class at each edge router is shown in Table I. We see that flows within the same class but from different edges realize different shares of the guaranteed bandwidth. In particular we note that the flows from Edge 1 (which has the smallest round-trip delay) receive a bigger share of the bandwidth. This is due to the well-know round-trip bias of TCP. In the absence of a class-based management scheme, all users from Edge 1 would have received an average throughput of 122 Kbps, all users from Edge 2 would have an average throughput of 76 Kbps, all users from Edge 3 would have an average throughput of 64 Kbps and all users from Edge 4 and Edge 5 would have an average throughput of 54 Kbps and 46 Kbps respectively.

Simulation 2: In this simulation, we simulate a scenario in which all Class 1 flows have a maximum rate of 400 Kbps. This is achieved by limiting the maximum window size of the connections. Class 1 flows are guaranteed at least 6 Mbps at the bottleneck link. We ignore Class 2 flows in this simulation. Initially there are 10 Class 1 flows (two Class 1 flows per edge) and 80 best-effort flows (16 flows per edge). Between 100s and 110s 5 new Class 1 flows (one flow per edge) are introduced. The utilization of Class 1 and best-effort flows at the links are shown in Figure 13. We can see that the utilization of Class 1 flows is below the minimum guaranteed throughput till new flows are introduced into the link. The reason is all Class 1 flows have achieved their maximum rate of 400 Kbps leading to a utilization of only 4 Mbps at the link. The remaining bandwidth is taken by the best effort flows. When 5 new Class 1 flows are introduced, we can see that the utilization of Class 1 flows increases to accommodate the new flows. The average flow rate of a flow in each class is shown in Figure 14. We can see that each flow regardless of its round-trip delay achieves a throughput close of 400 Kbps. Introduction of new Class 1 flows do not affect the throughput of already established connections since with

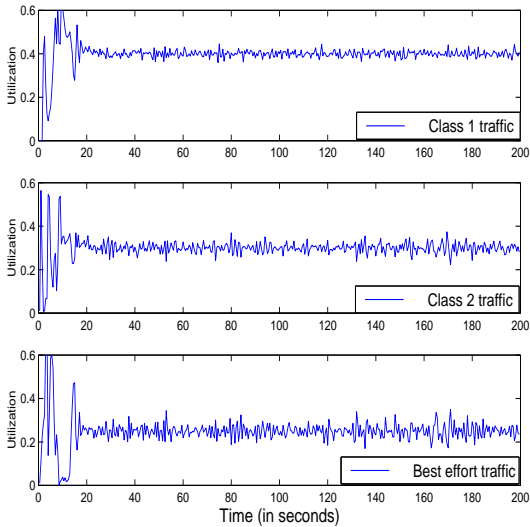


Fig. 8. Simulation 1. Utilization at the link for different classes of traffic

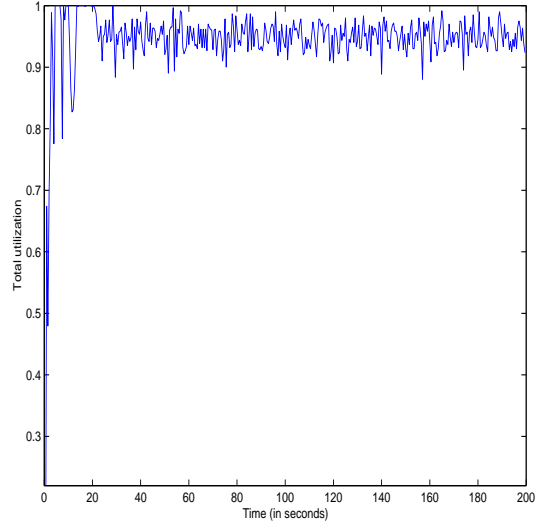


Fig. 9. Simulation 1. Total utilization at the link

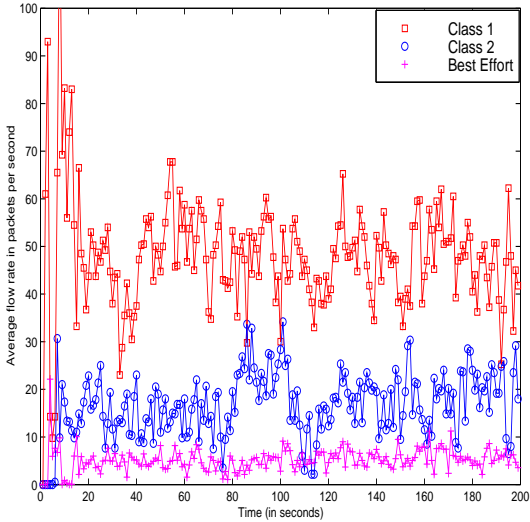


Fig. 10. Simulation 1. Evolution of the average flow rate in each class from Edge 1

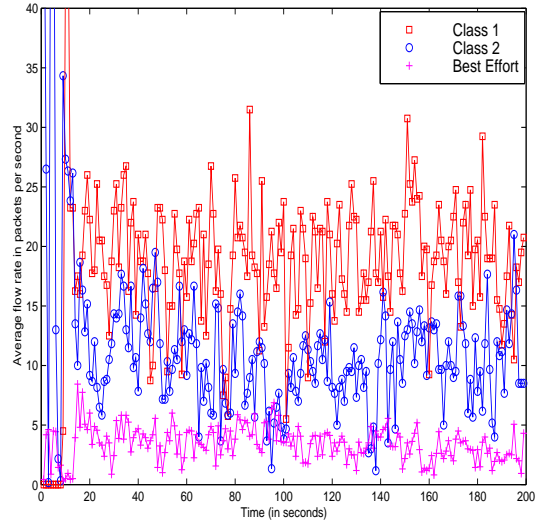


Fig. 11. Simulation 1. Evolution of the average flow rate in each class from Edge 3

the new flows the total throughput of all Class 1 flows at the link is 6 Mbps and equal to the guaranteed minimum throughput of Class 1 flows at the link. As a result, the link does not mark any Class 1 packets. The average flow rate of the Class 1 flows introduced after 100s is shown in Figure 15. We can see that the flows quickly achieve a rate of 400 Kbps.

Simulation 3: In this simulation we show that minimum bandwidth guarantees do not underutilize the links. When Class 1 and Class 2 are not able to achieve their minimum throughput guarantees, the bandwidth is shared by all the flows in inverse proportion to their flow rates, that is, flows with higher rates get a smaller share of the excess bandwidth. We lower the minimum guaranteed throughput at the link for Class 1 and Class 2 traffic to 15% and 10% for better illustration. We start the simulations with 10 flows in Class 1 and Class 2 and 100 flows in the best effort category. At time 150 seconds, all flows of Class 1 depart and at time 250 seconds, all

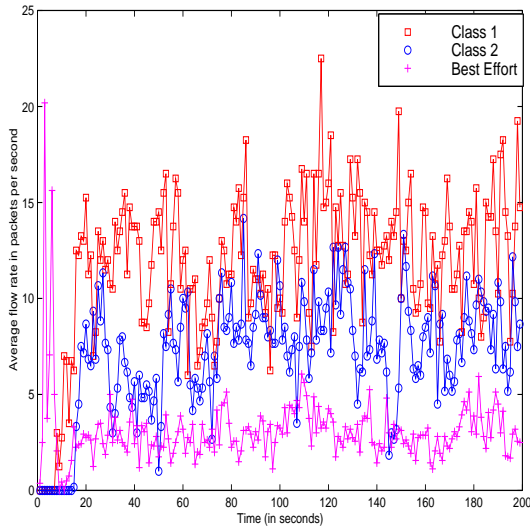


Fig. 12. Simulation 1. Evolution of the average flow rate in each class from Edge 5

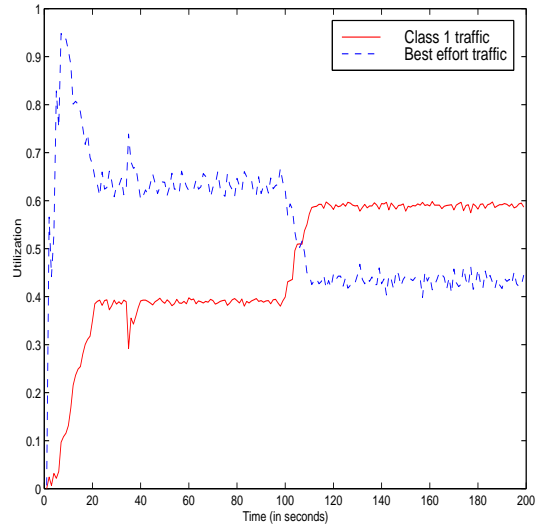


Fig. 13. Simulation 2. Utilization at the link for different classes of traffic

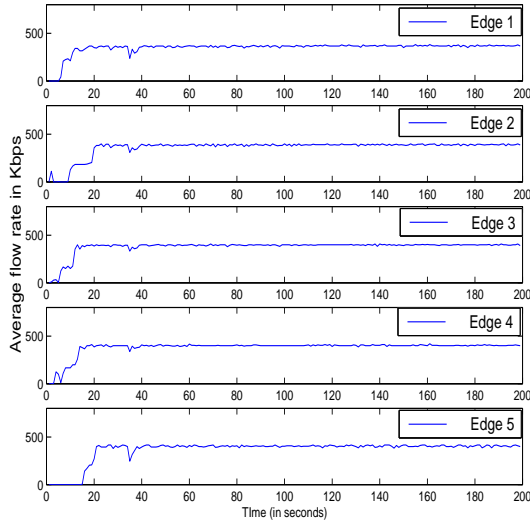


Fig. 14. Simulation 2. Evolution of the average flow rate of a Class 1 flows from each Edge

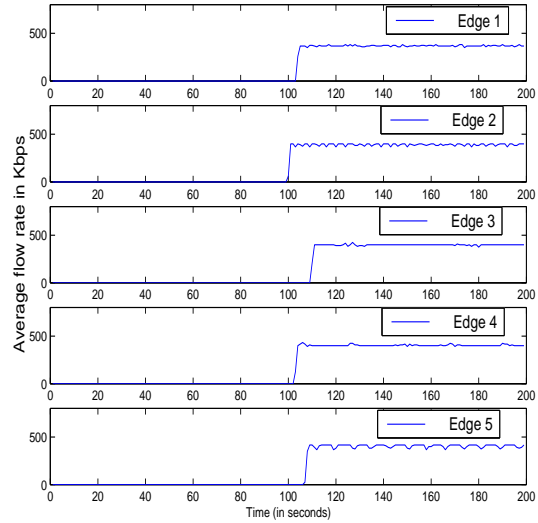


Fig. 15. Simulation 2. Evolution of the average flow rate of a Class 1 from each Edge

flows of Class 2 depart. The utilization at the link for each class is shown in Figure 16. We can see that at time 150 seconds, the utilization of Class 3 increases to use up the excess bandwidth caused by departure of Class 1 flows. There is no significant increase in utilization of Class 2 flows since the excess bandwidth is distributed inversely proportional to the flow rates and hence Class 3 flows get a major share of the excess bandwidth (due to the larger number of class 3 flows and a smaller rate per flow). Similarly, at time 250 seconds, the utilization of Class 3 flows jumps to 0.95 since all Class 1 and Class 2 flows have departed.

Simulation 4: Till now we have ignored the presence of short flows in performance of the proposed scheme. We know that a large number of flows in the Internet can be classified as short flows. As a result, we repeat Experiment 1 with short flows arriving and departing at the link along with long flows. We assume that there are 3 Class 1 flows, 5 Class 2 flows and 14 best effort

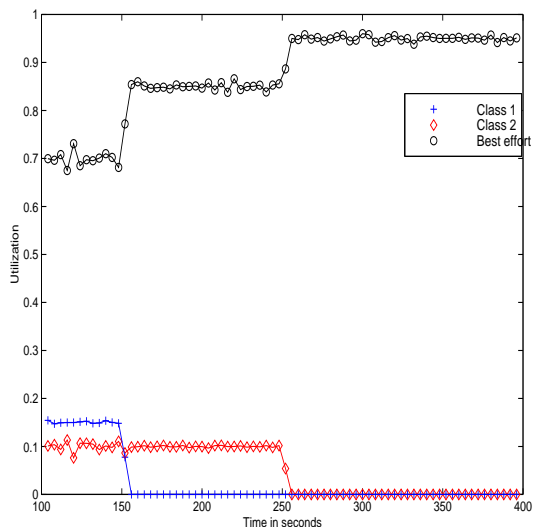


Fig. 16. Simulation 3. Utilization at the link for different classes of traffic when flows depart

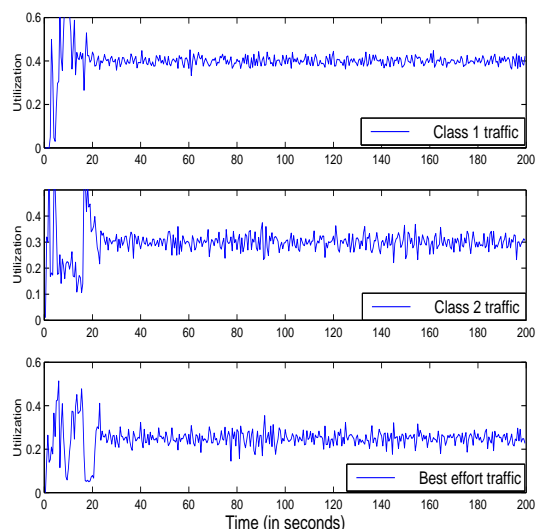


Fig. 17. Simulation 4. Utilization at the link for different classes of traffic with short flows

flows per edge that persist throughout the simulations. We simulate short flows using a Poisson arrival process. Each class has an arrival rate of 5 flows per second with flows choosing the edges randomly. Each short flow transmits 20 packets. The utilization at the link for each class is plotted in Figure 17. We see that the presence of short flows does not alter the performance of the proposed algorithm. Each class is still able to achieve its minimum guaranteed throughput.

V. STABILITY ANALYSIS

In this section we discuss the stability of the congestion controllers and the link adaptation schemes as a function of the adaptation parameters κ_r^j and α_l^j . We will assume that the total desired utilization γ_l , and the minimum guaranteed throughput for each class $\{\eta_l^j\}$ at a link are design parameters that are chosen by the service provider. A high value of the link adaptation parameters might destabilize the network. A low value of the link adaptation parameters might lead to good stability results, but the system will tend to be sluggish and less robust to sudden changes in load. In this section we provide conditions on the congestion-controller parameters ($\{\kappa_r^j\}$) and the link adaptation parameters ($\{\alpha_l^j\}$ and $\{\alpha_r^j\}$) for local stability of the algorithms. For ease of exposition, we will only consider the $\frac{-1}{x}$ utility function in this paper. As mentioned before, the $\frac{-1}{x}$ utility function approximates the TCP congestion controller [16]. The results segue smoothly to more general utility functions and will be omitted in this paper for the sake of clarity. For each flow f_r^j , let $d_1^j(r, k)$ be the propagation delay from the source to the link k , and $d_2^j(r, k)$ be the feedback delay from the link k back to the source. Let T_r^j be the total round-trip delay for flow f_r^j . Note that

$$T_r^j = d_1^j(r, l) + d_2^j(r, l) \quad \forall l \in \mathcal{L}_r^j. \quad (27)$$

We assume T_r^j to be a constant such that (27) is satisfied for all links through which the flow traverses. The system comprising the congestion-controllers at the sources and the adaptation algorithm at the links can now be described by the following set of differential equations:

- For all $f_r^j \in \mathcal{R}$,

$$\frac{dx_r^j}{dt} = \kappa_r^j \left[1 - x_r(t)x_r^j(t - T_r^j) \sum_{l:l \in \mathcal{L}_r^j} (\bar{p}_l(t) - \bar{q}_l^j(t))^+ \right], \quad (28)$$

- for all $l \in \mathcal{L}$,

$$\frac{d\tilde{C}_l}{dt} = \begin{cases} \alpha_l(\gamma_l C_l - \lambda_l(t)) & \tilde{C}_l > 0 \\ \max[0, \alpha_l(\gamma_l C_l - \lambda_l(t))] & \tilde{C}_l = 0 \end{cases}, \quad (29)$$

- and for all $j \in \{1, 2, \dots, K-1\}$ and $l \in \mathcal{L}$,

$$\frac{d\tilde{C}_l^j}{dt} = \begin{cases} \alpha_l^j(\eta_l^j C_l - \lambda_l^j(t)) & \tilde{C}_l^j > 0 \\ \max[0, \alpha_l^j(\eta_l^j C_l - \lambda_l^j(t))] & \tilde{C}_l^j = 0 \end{cases}, \quad (30)$$

where \bar{p}_l and \bar{q}_l^j denote the time-delayed versions of the feedback from link l and are given by,

$$\begin{aligned} \bar{p}_l(t) &= p_l \left(\lambda_l(t - d_2^j(r, l)), \tilde{C}_l(t - d_2^j(r, l)) \right), \\ \bar{q}_l^j(t) &= q_l^j \left(\lambda_l^j(t - d_2^j(r, l)), \tilde{C}_l^j(t - d_2^j(r, l)) \right), \end{aligned}$$

and $\lambda_l(t)$ and $\lambda_l^j(t)$ denote the time delayed versions of the total flow into the link and the total Class j flow link into the link respectively and are given by

$$\begin{aligned} \lambda_l(t) &= \sum_{m,k:f_m^k \in \mathcal{R}_l} x_m^k(t - d_1^k(m, l)), \\ \lambda_l^j(t) &= \sum_{m:f_m^j \in \mathcal{R}_l \cap \mathcal{R}_l^j} x_m^j(t - d_1^j(m, l)). \end{aligned}$$

Note that in the absence of propagation and feedback delays i.e., $T_r^j = 0$, the system reduces to the model in (7)-(9). The starting point of our analysis is the linearization of the system given in (28)-(30) about the equilibrium point. Due to space constraints, we do not present the linearization steps in this paper. In order to do this, we assume that each link has at least one flow of each class that passes only through it. This assumption ensures that the equilibrium point of the system lies in the interior of the domain. With this assumption, we can see that $\hat{p}_l - \hat{q}_l^j > 0$ for all $l \in \mathcal{L}$ and for all j . Let $y_r^j(t) := x_r^j(t) - \hat{x}_r^j$, where \hat{x}_r^j denotes the equilibrium value of $x_r^j(t)$. Let \hat{C}_l denote the equilibrium value of $\tilde{C}_l(t)$ and \hat{C}_l^j denote the equilibrium value of $\tilde{C}_l^j(t)$. Similarly, let $\hat{\lambda}_l$ and $\hat{\lambda}_l^j$ denote the equilibrium values of the total flow into the link and the total Class j flow into the link respectively. Denote:

$$\begin{aligned} p_l \left(\hat{\lambda}_l, \hat{C}_l \right) &\text{ by } \hat{p}_l, & \frac{\partial p_l}{\partial x} \left(\hat{\lambda}_l, \hat{C}_l \right) &\text{ by } \hat{p}_l^x, \\ \left| \frac{\partial p_l}{\partial \tilde{C}_l} \left(\hat{\lambda}_l, \hat{C}_l \right) \right| &\text{ by } \hat{p}_l^{\tilde{C}}, & q_l^j \left(\hat{\lambda}_l^j, \hat{C}_l^j \right) &\text{ by } \hat{q}_l^j, \\ \left| \frac{\partial q_l^j}{\partial x} \left(\hat{\lambda}_l^j, \hat{C}_l^j \right) \right| &\text{ by } \hat{q}_l^{x^j}, & \text{ and } \frac{\partial q_l^j}{\partial \tilde{C}_l} \left(\hat{\lambda}_l^j, \hat{C}_l^j \right) &\text{ by } \hat{q}_l^{\tilde{C}^j}. \end{aligned}$$

Let $y_k(s)$ denote the Laplace transform of $y_k(t)$ and define $Y(s) := [y_1(s), y_2(s), \dots, y_{|\mathcal{R}|}(s)]^T$. Note that \hat{q}_l^j is a decreasing function of x_r^j . Motivated by the approach in [23], we can now write the Laplace transform of the linearized version of (28) - (30) as

$$Y(s) = -L(s)G(s)Y(s), \quad (31)$$

where

$$L(s) = \text{diag}\left\{\frac{e^{-sT_i^j}}{sT_i^j + \phi_i^j T_i^j}\right\}, \quad \phi_i^j = \kappa_i^j \hat{x}_i^j \sum_{l \in \mathcal{L}_i^j} (\hat{p}_l - \hat{q}_l)^+,$$

$$G(s) := \text{diag}\{\kappa_i^j T_i^j\} X^2 \left[X^{-2} + \tilde{M}(s) + \frac{1}{s} \bar{M}(s) \right],$$

$$\tilde{M}_{f_k^j f_r^i}(s) = \begin{cases} \hat{p}_l^x e^{-s(d_1^i(r,l) - d_1^j(k,l))} & j \neq i \\ (\hat{p}_l^x + \hat{q}_l^x) e^{-s(d_1^i(r,l) - d_1^j(k,l))} & j = i \end{cases}, \quad (32)$$

$$\bar{M}_{f_k^j f_r^i}(s) = \begin{cases} \alpha_l \hat{p}_l^{\tilde{c}} e^{-s(d_1^i(r,l) - d_1^j(k,l))} & j \neq i \\ (\alpha_l \hat{p}_l^{\tilde{c}} + \alpha_l^j \hat{q}_l^{\tilde{c}j}) e^{-s(d_1^i(r,l) - d_1^j(k,l))} & j = i \end{cases}, \quad (33)$$

$$\text{and } X = \text{diag}\{\hat{x}_i^j\}.$$

Define,

$$K(s) := L(s) \text{diag}\{\kappa_i^j T_i^j\} X^2 \left[X^{-2} + \tilde{M}(s) \right]. \quad (34)$$

$K(s)$ describes the system of congestion-controllers in the absence of any link adaptation, i.e, $K(s)$ describes the system when $\alpha_l = 0$ and $\alpha_l^j = 0$ for all $l \in \mathcal{L}$ and all j .

We first consider the stability of the system of congestion-controllers in (28) without the link adaptation algorithms. In this case, the transfer function of the system is given by

$$Y(s) = -K(s)Y(s), \quad (35)$$

where $K(s)$ is given in (34).

Theorem V.1 *The system of congestion-controllers defined by the transfer function in (35) is locally stable if for all flows $f_r^j \in \mathcal{R}$,*

$$\kappa_r^j \hat{x}_r^j \sum_{l \in \mathcal{L}_r^j} \left(\hat{p}_l + \hat{q}_l^j + \hat{p}_l^x \gamma_l C_l + \hat{q}_l^{xj} \eta_l^j C_l \right) \leq \frac{\pi}{2T_r^j}. \quad (36)$$

Proof: Please refer to Appendix A. ■

Theorem V.1 shows that the adaptation speed of the congestion-controllers should be inversely proportional to the round-trip delay of the flow for ensuring the local stability of the congestion controllers. The result is similar to the result obtained in [23] for a $\log(x)$ utility function when no minimum throughput guarantees are provided. We now state the theorem that provides conditions on the local stability of the system of congestion-controllers and link adaptation algorithms in (28)-(30). Define, $T_{\max} := \max_{f_r^j \in \mathcal{R}} T_r^j$ and $x_{\max} := \max_{f_r^j \in \mathcal{R}} \hat{x}_r^j$.

Theorem V.2 *Given a $0 < \delta < 1$ if*

$$\kappa_r^j \hat{x}_r^j \sum_{l \in \mathcal{L}_r^j} \left(\hat{p}_l + \hat{q}_l^j + \hat{p}_l^x \gamma_l C_l + \hat{q}_l^{xj} \eta_l^j C_l \right) = \delta \frac{\pi}{2T_r^j} \quad (37)$$

$$\alpha_l \leq \frac{\hat{p}_l^x}{\hat{p}_l^{\tilde{c}} T_{\max}} \min \left\{ \frac{1}{\sqrt{2}}, \frac{(1-\delta)}{\delta} \frac{\pi}{8(\hat{x}_{\max} T_{\max})} \right. \\ \left. \min_{f_m^k} \left[\frac{\sum_{l \in \mathcal{L}_m^k} (\hat{p}_l + \hat{q}_l^j)}{\sum_{l \in \mathcal{L}_m^k} (\hat{p}_l + \hat{q}_l^k + \hat{p}_l^x \hat{\lambda}_l) + \hat{q}_l^{xk} \hat{\lambda}_l^k} \right] \right\}$$

and

$$\alpha_l^j \leq \frac{\hat{q}_l^{x^j} \hat{p}_l^{\tilde{c}}}{\hat{q}_l^{\tilde{c}^j} \hat{p}_l^{x^j}} \alpha_l \quad \forall l \in \mathcal{L},$$

then the system comprising of the congestion-controllers at the sources and the link adaptation algorithms is locally asymptotically stable.

Proof: We outline the proof here. For the complete proof, please refer to Appendix A.

To show the stability of (31), we show that the eigenvalues of $L(j\omega)G(j\omega)$ do not encircle -1 for all values of ω . Towards this, we briefly outline the three steps that we use to show the local stability of the system:

- (i) There exists a ω^* such that no eigenvalue of $L(j\omega)G(j\omega)$ is real for all $\omega < \omega^*$.
- (ii) Given a δ , each user can choose $\{\kappa_r\}$ according to (37) such that a $\frac{\delta}{2}$ neighborhood of the eigenvalues of $K(j\omega)$ do not enclose -1 for all ω .
- (iii) Finally, if α_l and α_l^j are chosen according to (38), the eigenvalues of $L(j\omega)G(j\omega)$ can be bounded by a $\frac{\delta}{2}$ neighborhood around the eigenvalues of $K(j\omega)$ for all $\omega > \omega^*$. Using (i) and (ii), we can now easily show that the eigenvalues of $L(j\omega)G(j\omega)$ do not enclose -1 .

Appealing to the Generalized Nyquist criteria, the system is locally asymptotically stable. ■

The following corollary specializes the above theorem to a special case when the marking function takes a particular form.

Corollary V.1 Suppose the marking functions at each node are given by

$$p_l(\lambda, \tilde{C}) = \left(\frac{\lambda}{\tilde{C}}\right)^B \quad \text{and} \quad q_l(\lambda^j, \tilde{C}^j) = \left(\frac{\tilde{C}^j}{\lambda^j}\right)^B,$$

for some $B > 1$. Given a $0 < \delta < 1$, if

$$\kappa_r^j \hat{x}_r^j (1+B) \sum_{l \in \mathcal{L}_r^j} (\hat{p}_l + \hat{q}_l^j) = \delta \frac{\pi}{2T_r^j}, \quad \forall f_r^j \in \mathcal{R}$$

$$\alpha_l \leq \frac{1}{\hat{p}_l^{1/B} T_{\max}} \min\left\{\frac{1}{\sqrt{2}}, \frac{1-\delta}{\delta} \frac{\pi}{8(1+B)\hat{x}_{\max} T_{\max}}\right\}, \quad \forall l \in \mathcal{L}$$

and for all $j \neq K$,

$$\alpha_l^j \leq \frac{(\hat{q}_l^j)^{1/B}}{T_{\max}} \min\left\{\frac{1}{\sqrt{2}}, \frac{1-\delta}{\delta} \frac{\pi}{8(1+B)\hat{x}_{\max} T_{\max}}\right\}, \quad \forall l \in \mathcal{L}$$

then the system of congestion-controllers and class based link adaptation algorithms are locally asymptotically stable. ■

VI. CONCLUSIONS AND FUTURE WORK

In this paper we presented a framework to provide minimum throughput guarantees in the network using FIFO queues. The framework assures each class a minimum throughput during periods of congestion while maintaining the same delay characteristics for all flows. The maximum delay that each user perceives can be controlled using the desired utilization of the network and the maximum buffer size at the queues. Choice of this desired utilization and buffer size to guarantee a particular delay characteristic is a subject of future research.

We then proposed and analyzed a decentralized scheme to achieve the minimum throughput guarantees. The scheme allows any congestion-controller to be used at the source and does not

require any changes at the source. As a result, one can use current TCP protocols at the sources as shown in the simulations in Section IV. We then presented a way to implement such schemes at the router and presented simulations results that agreed with analytical results derived earlier. We finally discussed the local stability properties of the algorithm and provided conditions on the adaptation parameters to ensure stability. One can use such conditions as design rules in implementing such schemes at routers.

In some instances, a service provider might want to create a strict hierarchy among classes to ensure that certain classes are not affected by congestion either due to increased traffic load or due to failures. The framework presented in this paper can be easily adapted to encompass such a hierarchy.

REFERENCES

- [1] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin. REM: Active queue management. *IEEE Network*, pages 48–53, June 2001.
- [2] Y. Bernat, J. Binder, S. Blake, M. Carlson, B. Carpenter, S. Keshav, E. Davies, B. Ohlman, and D. Berma. A framework for Differentiated services. In *Internet Draft*, 1999. Available at <http://www.ietf.org/proceedings/99nov/I-D/draft-ietf-diffserv-framework-02.txt>.
- [3] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999.
- [4] R. Braden, D. Clark, and S. Shenker. Integrated services in the Internet architecture. In *RFC1633*, 1994.
- [5] Y. Chait, C. Hollot, V. Misra, D. Towsley, H. Zhang, and J. Lui. Providing throughput differentiation for TCP flows using adaptive two-color marking and two-level AQM. In *Proceedings of Infocom*, June 2002.
- [6] D. Clark and W. Fang. Explicit allocation of best-effort packet delivery service. *IEEE/ACM Transactions on Networking*, August 1998.
- [7] D. Clark, S. Shenker, and L. Zhang. Supporting real-time services in the Internet. In *Proceedings, ACM Sigcomm*, pages 14–26, Baltimore, MD, 1992.
- [8] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *ACM SIGCOMM*, pages 1–12, 1989.
- [9] S. Floyd, R. Gummadi, and S. Shenker. Adaptive red: An algorithm for increasing the robustness of RED's active queue management. Available at <http://www.icir.org/floyd/adaptivered/>.
- [10] S. Floyd and V. Jacobson. Linksharing resource management models for packet networks. *IEEE/ACM transactions on networking*, 3(4):344–357, August 1995.
- [11] R. J. Gibbens and F. P. Kelly. On packet marking at priority queues. *IEEE Transactions on Automatic Control*, 47:1016–1020, 2002.
- [12] C.V. Hollot, V. Misra, D. Towsley, and W. Gong. On designing improved controllers for AQM routers supporting TCP flows. In *Proceedings of INFOCOM 2001*, Anchorage, Alaska, April 2001.
- [13] N. K. Jaiswal. *Priority queues*. Academic Press, 1968.
- [14] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis. Weighted round-robin cell multiplexing in a general purpose ATM switch chip. *IEEE journal on special areas in communication*, 9(8):1265–1279, October 1991.
- [15] F. P. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.
- [16] S. Kunniyur and R. Srikant. End-to-end congestion control: utility functions, random losses and ECN marks. In *Proceedings of INFOCOM 2000*, Tel Aviv, Israel, March 2000. Revised version to appear in *IEEE/ACM Transactions on Networking*, 2003.
- [17] S. Kunniyur and R. Srikant. Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management. In *Proceedings of SIGCOMM 2001*, San Diego, CA, August 2001. Revised version to appear in *IEEE/ACM Transactions on Networking*.
- [18] S. Kunniyur and R. Srikant. Designing AVQ parameters for a general network topology. In *Proceedings of the 4th Asian Control Conference*, September 2002. Revised version to appear in *IEEE/ACM Transactions on Automatic Control*.
- [19] C. Lagoa, Hao Che, and B. Movsichoff. Adaptive control algorithms for decentralized optimal traffic engineering. To appear in *Transactions on Networking*. Available from <http://crystal.uta.edu/hche/PUBLICATIONS/publication.htm>.
- [20] L. Massoulié and J. Roberts. Bandwidth sharing: Objectives and algorithms. In *Proc. INFOCOM*, New York, NY, March 1999.
- [21] ns 2 (online). <http://www.isi.edu/nsnam/ns>.
- [22] S. Shakkottai and R. Srikant. Mean FDE models for internet congestion control under a many-flows regime. In *Proceedings of IEEE Infocom*, 2002.
- [23] G. Vinnicombe. On the stability of end-to-end congestion control for the Internet. *Technical Report, CUED/F-INFENG/TR.398*, University of Cambridge, UK, December 2000.

APPENDIX A

In this section we provide the complete proof of the local stability theorem given by Theorem V.2.

Lemma VI.1 *If for all $l \in \mathcal{L}$ and $j \in \{1, 2, \dots, K-1\}$,*

$$\alpha_l < \frac{\hat{p}_l^x}{\sqrt{2\hat{p}_l^c} T_{\max}} \quad \text{and} \quad \alpha_l^j < \frac{\hat{q}_l^{x^j}}{\sqrt{2\hat{q}_l^{c^j}} T_{\max}}$$

then, for all $\omega < \omega^ := \frac{\pi}{4T_{\max}}$, the eigenvalues of $L(j\omega)G(j\omega)$ cannot be real.*

Proof: The proof of this lemma follows the proof in [18] which shows the stability of the congestion-controllers and AVQ algorithms with no classes. Define

$$H(j\omega) := \text{diag}\left\{\frac{e^{-j\omega T_i^j}}{j\omega T_i^j}\right\} \text{diag}\{\kappa_i^j T_i^j\} \sqrt{X} \left[X^{-2} + \tilde{M}(j\omega) + \frac{1}{j\omega} \bar{M}(j\omega) \right] \sqrt{X}.$$

Note that $\sigma(H(j\omega)) = \sigma(L(j\omega)G(j\omega))$. Let $\lambda(\omega)$ be an eigenvalue of $H(j\omega)$ and μ be the corresponding normalized eigenvector. Therefore,

$$\begin{aligned} \lambda \mu &= \left(\text{diag}\left\{\frac{e^{-j\omega T_i^j}}{T_i^j(j\omega + \theta_i^j)} \frac{\kappa_i^j T_i^j}{\hat{U}_i^j(\hat{x}_i^j)}\right\} \right) \sqrt{X} ([X^{-2} + \tilde{M}(j\omega)] + \frac{1}{j\omega} \bar{M}(j\omega)) \sqrt{X} \mu \\ \lambda \text{diag}\left\{e^{j\omega T_i^j} \frac{j\omega}{\kappa_i^j}\right\} \mu &= \sqrt{X} ([X^{-2} + \tilde{M}(j\omega)] + \frac{1}{j\omega} \bar{M}(j\omega)) \sqrt{X} \mu \\ \lambda &= \frac{\mu^* \sqrt{X} ([X^{-2} + \tilde{M}(j\omega)] + \frac{1}{j\omega} \bar{M}(j\omega)) \sqrt{X} \mu}{\mu^* \text{diag}\left\{e^{j\omega T_i^j} \frac{j\omega}{\kappa_i^j}\right\} \mu} \\ \lambda &= - \frac{(\mu^* \sqrt{X} (j\omega [X^{-2} + \tilde{M}(j\omega)] + \bar{M}(j\omega)) \sqrt{X} \mu) \left[\sum_{f_k^m} \frac{|\mu_k^m|^2}{\kappa_k^m} (\cos(\omega T_k^m) - j \sin(\omega T_k^m)) \right]}{\omega^2 \left(\sum_{f_k^m} \frac{|\mu_k^m|^2 \cos(\omega T_k^m)}{\kappa_k^m} \right)^2 + \omega^2 \left(\sum_{f_k^m} \frac{|\mu_k^m|^2 \sin(\omega T_k^m)}{\kappa_k^m} \right)^2}. \end{aligned} \quad (38)$$

Defining

$$\Lambda := \left(\sum_{f_k^m} \frac{|\mu_k^m|^2 \cos(\omega T_k^m)}{\kappa_k^m} \right)^2 + \left(\sum_{f_k^m} \frac{|\mu_k^m|^2 \sin(\omega T_k^m)}{\kappa_k^m} \right)^2 \quad \text{and} \quad \Sigma := X^{-2} + \tilde{M}(j\omega),$$

we can write the imaginary part of (38) as

$$\text{Im}(\lambda) = - \frac{\mu^* \sqrt{X} \Sigma \sqrt{X} \mu}{\omega \Lambda} \sum_{f_k^m} \frac{|\mu_k^m|^2}{\kappa_k^m} \left[\cos(\omega T_k^m) - T_k^m \frac{\mu^* \sqrt{X} \bar{M}(j\omega) \sqrt{X} \mu}{\mu^* \sqrt{X} \Sigma \sqrt{X} \mu} \frac{\sin(\omega T_k^m)}{\omega T_k^m} \right]. \quad (39)$$

Let $\alpha_l = \hat{\alpha} \frac{\hat{p}_l^x}{\hat{p}_l^c}$ and $\alpha_l^j = \hat{\alpha} \frac{\hat{q}_l^{x^j}}{\hat{q}_l^{c^j}}$. Therefore,

$$\frac{\mu^* \sqrt{X} \bar{M}(j\omega) \sqrt{X} \mu}{\mu^* \sqrt{X} \Sigma \sqrt{X} \mu} = \hat{\alpha} \frac{\mu^* \sqrt{X} \bar{M} \sqrt{X} \mu}{\mu^* \sqrt{X} [X^{-2} + \tilde{M}(j\omega)] \sqrt{X} \mu} < \hat{\alpha}.$$

Hence $\hat{\alpha} \leq \frac{1}{\sqrt{2}T_{\max}}$ is sufficient to ensure that (39) can never be zero if $\omega < \frac{\pi}{4T_{\max}}$. ■

Lemma VI.2 Let $\eta(\omega)$ be any eigenvalue of $K(j\omega)$. For a given $0 < \epsilon < 1$, if

$$\kappa_r^j \sum_{l \in \mathcal{L}_r^j} \left(\hat{p}_l + \hat{q}_l^j + \hat{p}_l^x \gamma_l C_l + \hat{q}_l^{x^j} \eta_l^j C_l \right) = \delta \frac{\pi}{2T_r^j}$$

then, for all ω , a $(1 - \epsilon)$ neighborhood of $\eta(\omega)$ will not enclose -1 .

Proof: Follows directly from Theorem V.1. ■

Lemma VI.3 Let $\lambda(\omega)$ be an eigenvalue of $G(j\omega)$. If

$$\alpha_l \leq \frac{\hat{p}_l^x}{\hat{p}_l^c T_{\max}} \min \left\{ \frac{1}{\sqrt{2}}, \frac{(1 - \delta)}{\delta} \frac{\pi}{8(\hat{x}_{\max} T_{\max})} \min_{f_m^k} \left[\frac{\sum_{l \in \mathcal{L}_m^k} (\hat{p}_l + \hat{q}_l^j)}{\sum_{l \in \mathcal{L}_m^k} (\hat{p}_l + \hat{q}_l^k + \hat{p}_l^x \lambda_l) + \hat{q}_l^{x^k} \lambda_l^k} \right] \right\},$$

$$\alpha_l^j \leq \frac{\hat{q}_l^{x^j} \hat{p}_l^c}{\hat{q}_l^c \hat{p}_l^x} \alpha_l$$

for all $l \in \mathcal{L}$ and $j \in \{1, 2, \dots, K - 1\}$ and if for all $f_j^r \in \mathcal{R}$,

$$\kappa_r^j \sum_{l \in \mathcal{L}_r^j} \left(\hat{p}_l + \hat{q}_l^j + \hat{p}_l^x \gamma_l C_l + \hat{q}_l^{x^j} \eta_l^j C_l \right) = \delta \frac{\pi}{2T_r^j}$$

then, for all $\omega > \omega^*$, $\text{Re}(\lambda(\omega)) > -1$.

Proof: For ease of exposition we define the following variables:

$$Q = \text{diag}\{\sqrt{\kappa_i^j T_i^j \hat{x}_i^j}\} (X^{-2} + \tilde{M}(j\omega)) \text{diag}\{\sqrt{\kappa_i^j T_i^j \hat{x}_i^j}\},$$

$$\tilde{Q} = \text{diag}\{\sqrt{\kappa_i^j T_i^j \hat{x}_i^j}\} \tilde{M}(j\omega) \text{diag}\{\sqrt{\kappa_i^j T_i^j \hat{x}_i^j}\}, \quad \text{and } L = \text{diag}\left\{\frac{e^{-j\omega T_i^j}}{j\omega T_i^j}\right\}.$$

Let $\alpha_l = \hat{\alpha} \frac{\hat{p}_l^x}{\hat{p}_l^c}$ and $\alpha_l^j = \hat{\alpha} \frac{\hat{q}_l^{x^j}}{\hat{q}_l^c}$. Note that

$$\sigma(L(j\omega)G(j\omega)) = \sigma\left(\left(Q + \frac{\hat{\alpha}}{j\omega} \tilde{Q}\right)L\right).$$

Let $\lambda(\omega)$ be an eigenvalue of $G(j\omega)$ and let μ be the corresponding eigenvector. Also note Q is a Hermitian positive definite matrix and \tilde{Q} is a Hermitian matrix. Therefore

$$\begin{aligned} (Q + \frac{\hat{\alpha}}{j\omega} \tilde{Q})L\mu &= \lambda\mu \\ (\Leftrightarrow) \quad Q(I + \frac{\hat{\alpha}}{j\omega} Q^{-1} \tilde{Q})L\mu &= \lambda\mu \\ (\Leftrightarrow) \quad (I + \frac{\hat{\alpha}}{j\omega} Q^{-1} \tilde{Q})L\mu &= \lambda Q^{-1}\mu \\ (\Rightarrow) \quad \mu^*(I + \frac{\hat{\alpha}}{j\omega} Q^{-1} \tilde{Q})L\mu &= \lambda \mu^* Q^{-1}\mu \\ (\Leftrightarrow) \quad \frac{\mu^* L\mu}{\mu^* Q^{-1}\mu} + \frac{\hat{\alpha}}{j\omega} \frac{\mu^* Q^{-1} \tilde{Q} L\mu}{\mu^* Q^{-1}\mu} &= \lambda. \end{aligned}$$

Note that from Lemma VI.2, we know that the real part of the first term $\frac{\mu^* L \mu}{\mu^* Q^{-1} \mu}$ does not encircle $-\epsilon$. Therefore, we need to show that

$$\left| \frac{\hat{\alpha} \mu^* Q^{-1} \tilde{Q} L \mu}{j\omega \mu^* Q^{-1} \mu} \right| \leq (1 - \epsilon).$$

Next, note that

$$\begin{aligned} \left| \frac{\hat{\alpha} \mu^* Q^{-1} \tilde{Q} L \mu}{j\omega \mu^* Q^{-1} \mu} \right| &\leq \frac{\hat{\alpha} \|Q^{-1}\|_2 \|\tilde{Q}\|_2 \|L\|_2}{\omega^* \lambda_{\min}(Q^{-1})} \\ &\leq \frac{\hat{\alpha} \rho(Q)^2}{\lambda_{\min}(Q) (\omega^*)^2} \max_k \left(\frac{1}{T_i} \right), \end{aligned}$$

where $\|\cdot\|_2$ is the matrix induced spectral norm given by $\|A\|_2 = \sqrt{\lambda_{\max}(A^*A)}$. The first inequality follows directly from the Cauchy-Schwartz inequality and the last inequality is due to the fact that $\rho(\tilde{Q}) < \rho(Q)$. We know that

$$\begin{aligned} \lambda_{\min}(Q) &= \lambda_{\min}(\text{diag}\{\sqrt{\kappa_k T_k \hat{x}_k}\} (X^{-2} + \tilde{M}(j\omega)) \text{diag}\{\sqrt{\kappa_k T_k \hat{x}_k}\}) \\ &\geq \lambda_{\min}(\text{diag}\{\sqrt{\kappa_k T_k \hat{x}_k}\} X^{-2} \text{diag}\{\sqrt{\kappa_k T_k \hat{x}_k}\}) \\ &= \min_k \left(\frac{\Delta_k \kappa_k T_k}{\hat{x}_k} \right). \end{aligned}$$

Since $\rho(Q) = \epsilon \frac{\pi}{2}$ and $\omega^* = \frac{\pi}{4T_{\max}}$, we have

$$\left| \frac{\hat{\alpha} \mu^* Q^{-1} \tilde{Q} L \mu}{j\omega \mu^* Q^{-1} \mu} \right| \leq \hat{\alpha} \frac{4\epsilon^2}{\min_k \left(\frac{\Delta_k \kappa_k T_k}{\hat{x}_k} \right) T_{\min} T_{\max}^2}.$$

If x_{max} is measured in packets-per-second, $T_{\min} \geq \frac{1}{x_{max}}$ due to processing delays. Therefore,

$$\begin{aligned} \alpha_l &\leq \frac{\hat{p}_l^x}{\hat{p}_l^{\tilde{c}} T_{\max}} \min \left\{ \frac{1}{\sqrt{2}}, \frac{(1-\delta)}{\delta} \frac{\pi}{8(\hat{x}_{\max} T_{\max})} \min \left[\frac{\sum_{l \in \mathcal{L}_m^k} (\hat{p}_l + \hat{q}_l^j)}{\sum_{l \in \mathcal{L}_m^k} (\hat{p}_l + \hat{q}_l^k + \hat{p}_l^x \hat{\lambda}_l) + \hat{q}^{x^k} \hat{\lambda}_l^k} \right] \right\}, \\ \alpha_l^j &\leq \frac{\hat{q}_l^{x^j} \hat{p}_l^{\tilde{c}}}{\hat{q}_l^{\tilde{c}^j} \hat{p}_l^x} \alpha_l \end{aligned}$$

ensures that

$$\left| \frac{\hat{\alpha} \mu^* Q^{-1} \tilde{Q} L \mu}{j\omega \mu^* Q^{-1} \mu} \right| \leq (1 - \epsilon).$$

Therefore, the eigenvalues do not enclose -1. ■

\mathcal{L}	Set of all links
C_l	Capacity of link l
γ_l	Desired utilization at link l
η_l^j	Fraction of link capacity that is guaranteed for Class j at link l
\mathcal{R}	Set of all routes
\mathcal{R}^j	Set of all routes that belong to Class j
\mathcal{R}_l	Set of all routes that traverse link l
f_r^j	A flow r in class j
\mathcal{L}_r^j	Set of links that flow f_r^j traverses
x_r^j	Rate at which traffic is generated by flow f_r^j
$d_1^j(r, k)$	Delay from the source to link k for flow f_r^j
$d_2^j(r, k)$	Delay from link k to the source for flow f_r^j
T_r^j	Total delay for flow f_r^j
λ_l	Total arrival rate at link l
λ_l^j	Total arrival rate for Class j at link l
\tilde{C}_l	Parameter that determines the total utilization of link l
\tilde{C}_l^j	Parameter that determines the utilization of Class j at link l

TABLE II

NOTATIONS USED THROUGHOUT THE PAPER

APPENDIX B

Table II summarizes all the notation used in this paper.

DEPT. OF ELECTRICAL AND SYSTEMS ENGINEERING, UNIVERSITY OF PENNSYLVANIA, PHILADELPHIA, PA 19104.
E-mail address: kunnipur@seas.upenn.edu