Tradeoffs in Designing Networks with End-to-End Statistical QoS Guarantees [*]

Jörg Liebeherr [*] *Stephen Patek* [**] *Erhan Yilmaz* [*]

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

[**] Department of Systems Engineering
University of Virginia
Charlottesville, VA 22903

Abstract

Recent research on statistical multiplexing has provided many new insights into the achievable multiplexing gain in QoS networks, however, generally only in terms of the gain experienced at a single switch. Evaluating the statistical multiplexing gain in a general network remains a difficult challenge. In this paper we describe two distinct network designs for statistical end-to-end delay guarantees, referred to as *class-level aggregation* and *path-level aggregation*, and compare the achievable statistical multiplexing gain. Each of the designs presents a particular trade-off between the attainable statistical multiplexing gain and the ability to support delay guarantees. The key characteristic of both designs is that they do not require, and instead, intentionally avoid, consideration of the correlation between flows at multiplexing points inside the network. Numerical examples are presented for a comparison of the two designs. The presented class-level aggregation design is shown to yield very high achievable link utilizations while simultaneously achieving desired statistical guarantees on delay.

Key Words: Statistical Multiplexing, Statistical Service, Quality-of-Service.

[*] This work is supported in part by the National Science Foundation through grants NCR-9624106 (CAREER), ECS-9875688 (CAREER), ANI-9730103, ANI-9903001, and ANI-0085955.

1 Introduction

In recent years a lot of effort has gone into devising algorithms to support deterministic or statistical QoS guarantees in packet networks. *Deterministic services* [15], which guarantee worst-case end-to-end delay bounds for traffic [7, 9, 28, 29], are known to lead to an inefficient use of network resources [39]. *Statistical services* [15], which make guarantees of the form

$$Pr[Delay > X] < \varepsilon \, , \tag{1}$$

i.e. services which allow a small fraction of traffic to violate QoS specifications, can significantly increase the achievable utilization of network resources. Taking advantage of the statistical properties of traffic, a statistical service can exploit *statistical multiplexing gain*, expressed as

$$\left(\begin{array}{l} \text{Resources needed to} \\ \text{support statistical} \\ \text{QoS of } N \text{ flows} \end{array} \right) \ll N \times \left(\begin{array}{l} \text{Resources needed to} \\ \text{support statistical} \\ \text{QoS of 1 flow} \end{array} \right) .$$

Ideally, the statistical multiplexing gain of a statistical service increases with the volume of traffic so that with a high enough level of aggregation the amount of resource allocated per flow is nearly equal to the average resource requirements for a single flow.

Recent research on statistical QoS has attempted to exploit statistical multiplexing gain by taking advantage of knowledge about deterministic bounds on arrivals from individual flows, with limited knowledge about their statistical properties [3, 10, 12, 17, 18, 20, 25, 26, 31, 32, 33]. Under a very general set of traffic assumptions, which are sometimes referred to as ‘regulated adversarial traffic,’ one merely assumes that (1) traffic arrivals from a flow are constrained by a deterministic regulator, e.g., a leaky bucket, and (2) traffic arrivals from different flows are statistically independent. With these general assumptions it has been shown that even if the probability of QoS violations is small, e.g., $\varepsilon = 10^{-9}$, the statistical multiplexing gain at a network node can be substantial [3].

In this paper we are concerned with end-to-end statistical QoS guarantees in a multi-node network under adversarial regulated traffic assumptions. The difficulty of assessing the multiplexing gain in a network environment is that traffic inside the network becomes correlated, and, therefore, the assumption of independence, as made by the regulated adversarial traffic model, no longer holds.

1.1 Networks with Statistical End-to-End Guarantees

We consider a packet network such as the one shown in Figure 1. The network has two types of nodes, edge nodes and core nodes. Edge nodes are located at the boundary of the network and have links to core nodes or other edge nodes. Core nodes have no links that cross the network boundary. The network distinguishes a fixed number of traffic classes, and flows from the same class have the same characteristics and the same QoS requirements. We assume that the traffic from each indivudual flow is bounded according to a prescribed traffic profile. This could be accomplished by conditioning flows at their ingress points. In this way, traffic which conforms to its profile is allowed into the network, whereas traffic which does not conform to its profile is discarded or perhaps marked and subsequently treated with lower priority. Finally, we assume that nodes execute a scheduling algorithm which can provide rate guarantees to groups of flows [35, 41].

Within this framework, we develop and compare two approaches, referred to as *class-level aggregation* and *path-level aggregation*, for provisioning a network with end-to-end statistical QoS guarantees. Our

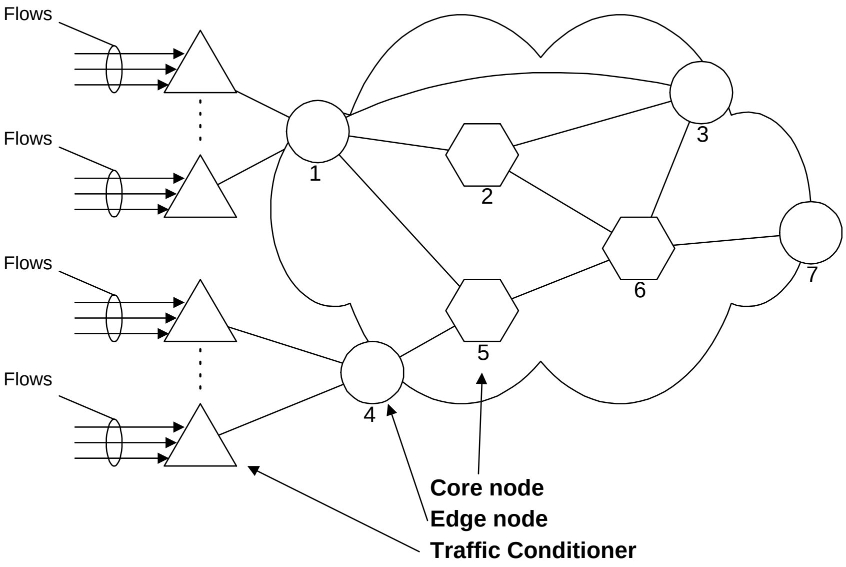


Figure 1: Network Architecture.

discussions will investigate the trade-offs presented by the two schemes. A comparison of the approaches will allow us to make recommendations on the design of QoS networks with statistical QoS guarantees.

Overall, we consider statistical QoS guarantees made to traffic on a per-class basis, and not on a per-flow basis. By making QoS guarantees to the aggregate flows from a class and not for specific flows within a class, the design of the core network can be greatly simplified since no per-flow information is required inside the network. The disadvantage of per-class guarantees is that individual flows may experience a service which is worse than the service guaranteed to the class as a whole.

1.2 Related Work

The available literature on statistical QoS is extensive. We refer to [21, 34] for summaries of the state of the art. Here we highlight only a small subset of related literature that focuses on *end-to-end* statistical QoS.

The main difficulty of provisioning statistical QoS for a multi-node network lies in addressing the complex correlation of traffic at downstream multiplexing points. One group of work on end-to-end statistical QoS, attempts to achieve a characterization of correlated traffic inside a network [5, 22, 36, 40]. An alternative approach, which we adopt in Section 3, is to reconstruct traffic characteristics inside the network so that arrivals to core nodes satisfy the same properties as the arrivals to an edge node. There are two approaches to reconstruct characteristics of traffic: per-node traffic shaping [16, 42], or per-node delay jitter control [37, 38]. In Section 3 we take the latter approach.

Another method to achieve statistical end-to-end guarantees is to allocate network capacity for each path or ‘pipe’ between a source-destination pair in the network, and only exploit the multiplexing gain between the flows on the same path. This method for allocating resource has been considered for use in Virtual Private Networks (VPN) [11] and ATM Virtual Paths [34]. We take such an approach in Section 4.

There are only a few previous studies which apply the traffic model of adversarial regulated arrivals to multiple node networks. The lossless multiplexer presented in [32] bears similarity to our design for class-level aggregation in Section 3, but assumes that routes are such that traffic arrivals at core nodes from different flows are always independent. We relax this assumption in our work, and instead, enforce independence by adding appropriate mechanisms within the network. Probabilistic bounds for end-to-end delays have been derived for networks with coordinated schedulers [2, 23]. This class of scheduling algorithms addresses end-to-end provisioning of QoS, by taking into consideration the time that a packet has already spend in the network at previous nodes.

This paper makes use of results from a recent study [3] which presented a general method to calculate the statistical multiplexing gain at a single node. In particular, we exploit the notion of *effective envelopes*, which are functions that provide bounds on traffic arrivals with high certainty. In addition, previous work on rate-based scheduling algorithms with statistical service guarantees in single-node networks is very relevant to our work [16, 30, 43, 44].

The remainder of this paper is structured as follows. In Section 2 we state our assumptions on traffic arrivals and we introduce the notion of effective envelopes. In Sections 3 and 4, respectively, we present our two designs for end-to-end statistical QoS and analyze their ability to exploit statistical multiplexing gain. In Section 5 we evaluate and compare the two designs through a computational study. In Section 6 we present our conclusions and discuss future research directions.

2 Traffic Arrivals and Effective Envelopes

In this section we present the details of our assumptions for the traffic arrivals. Throughout this paper we will use a fluid-flow interpretation of traffic. We define a function, called an *effective envelope*, which is with high certainty an upper bound on the traffic of multiplexed traffic flows. The concept of effective envelopes will be applied extensively in Sections 3 and 4. The discussion in this section is based on [3].

2.1 Regulated Adversarial Traffic

As in all arrival models for a statistical service, the arrivals of a flow are viewed as a random process. Consider a set $\mathcal{C}$ of flows which are partitioned into Q classes, where $\mathcal{C}_q$ denotes the subset of flows from class q. The traffic arrivals from flow j in the interval $[t_1, t_2)$ are denoted by a random variable $A_j(t_1, t_2)$ with the following properties:

(A1) **Additivity.** For any $t_1 < t_2 < t_3$, we have $A_j(t_1, t_2) + A_j(t_2, t_3) = A_j(t_1, t_3)$.

(A2) **Subadditive Bounds.** A_j is bounded by a deterministic subadditive envelope A_j^* as $A_j(t, t+\tau) \leq A_j^*(\tau)$ for all $t \geq 0$ and for all $\tau \geq 0$.[1]

(A3) **Stationarity.** The A_j are *stationary* so that for all $t, t' \geq 0$ we have $Pr[A_j(t, t+\tau) \leq x] = Pr[A_j(t', t'+\tau) \leq x]$.

(A4) **Independence.** The A_i and A_j are stochastically independent for all $i \neq j$.

(A5) **Homogeneity within a Class.** Flows in the same class have identical deterministic envelopes. Moreover, individual flows within a class have the same nominal delay bound requirements. These individual delay requirements are translated in Sections 3 and 4 into nominal delay bounds for the aggregate of flows within the class at each node.

[1] A function $f : \Re \mapsto \Re$ is subadditive if $f(t_1 + t_2) \leq f(t_1) + f(t_2)$, for all $t_1, t_2 \geq 0$.

These or similar assumptions are used in many recent works on statistical QoS [3, 10, 12, 17, 18, 20, 25, 26, 31, 32, 33]. The assumptions are very general. Specifically, no assumptions are made on the distribution of flow arrivals, other than that each flow satisfies a worst-case constraint.

Within the constraints of assumptions (A1)–(A5), we consider arrival scenarios where each flow exhibits its worst possible ('adversarial') behavior. Traffic which obeys the above assumptions is referred to as *regulated adversarial traffic*. Note that even if flows individually behave in a worst-case fashion, as allowed by assumption (A2), the independence assumption (A4) prevents the flows from 'conspiring' to yield a combined or joint worst case behavior.

2.2 Effective Envelopes of Aggregate Arrivals

For the calculation of statistical multiplexing gain we will take advantage of the notion of *effective envelopes*, which was recently presented in [3]. Effective envelopes are functions that are, with high probability, upper bounds on multiplexed traffic from a set of flows satisfying the assumptions of adversarial regulated traffic. Effective envelopes have been shown to be a useful tool for calculating the statistical multiplexing gain at a network node.[2]

Consider the set of flows $\mathcal{C}_q$ from a given class q. We use $A_{\mathcal{C}_q}$ to denote the aggregate arrivals from class q, that is, $A_{\mathcal{C}_q}(t, t+\tau) = \sum_{j \in \mathcal{C}_q} A_j(t, t+\tau)$. Also, let N_q denote the number of flows in set $\mathcal{C}_q$. Due to assumption $(A5)$, all flows in the same class have the same subadditive bound. Thus, we use A_q^* to denote the bound of class q with $A_j^*(\tau) = A_q^*(\tau)$ for all $j \in \mathcal{C}_q$.

Definition 1 *An effective envelope for $A_{\mathcal{C}_q}(t, t+\tau)$ is a function $\mathcal{G}_{\mathcal{C}_q}$ with:*

$$Pr\Big[A_{\mathcal{C}_q}(t, t+\tau) \le \mathcal{G}_{\mathcal{C}_q}(\tau; \varepsilon)\Big] \ge 1 - \varepsilon, \quad \forall\, t,\ \tau \ge 0\ . \tag{2}$$

Due to assumption (A3), an effective envelope provides a bound for the aggregate arrivals $A_{\mathcal{C}_q}$ for all time intervals of length τ, which is violated with probability ε.

Explicit expressions for effective envelopes can be obtained with large deviation results. In this paper, we will use a bound from [3] which is established via the Chernoff Bound. The Chernoff bound for the arrivals $A_{\mathcal{C}_q}$ from $\mathcal{C}_q$ is given by (see [27])

$$Pr[A_{\mathcal{C}_q}(0, \tau) \ge N_q x] \le e^{-N_q x s} M_{\mathcal{C}_q}(s, \tau)\ , \tag{3}$$

where $M_{\mathcal{C}_q}$ is the moment generating function of $A_{\mathcal{C}_q}$ defined as

$$M_{\mathcal{C}_q}(s, \tau) = E[e^{A_{\mathcal{C}_q}(t, t+\tau)s}]\ .$$

In [3], the following bound on the moment generating functions was proven.

Theorem 1 *(Boorstyn, Burchard, Liebeherr, Oottamakorn [3]) Given a set of flows $\mathcal{C}_q$ from a single class that satisfies assumptions (A1)–(A5). Then,*

$$M_{\mathcal{C}_q}(s, \tau) \quad \le \quad \left[1 + \frac{\rho_q\, \tau}{A_q^*(\tau)}\left(e^{sA_q^*(\tau)} - 1\right)\right]^{N_q}, \tag{4}$$

where $\rho_q := \lim_{\tau \to \infty} A_q^*(\tau)/\tau$.

[2] In [3] two notions of effective envelopes are introduced, called local effective envelope and global effective envelope. In this paper, we only use local effective envelopes and refer to them as effective envelopes.

Using this bound it is possible to show [3] that

$$\mathcal{G}_{\mathcal{C}_q}(\tau;\varepsilon) \ := \ N_q \min(x, A_q^*(\tau)) \ , \tag{5}$$

is an effective envelope for $A_{\mathcal{C}_q}$, when x is the smallest number satisfying the inequality

$$\left(\frac{\rho_q \tau}{x}\right)^{\frac{x}{A_q^*(\tau)}} \left(\frac{A_q^*(\tau) - \rho_q \tau}{A_q^*(\tau) - x}\right)^{1-\frac{x}{A_q^*(\tau)}} \ \leq \ \varepsilon^{1/N_q} \ . \tag{6}$$

We will use the effective envelope given by Eqs. (5) and (6) in all our numerical examples in Section 5.

3 Networks with Class-Level Aggregation ("Jitter Control Method")

In this section we discuss the first of our two approaches to achieve statistical delay guarantees in a multi-node network with regulated adversarial traffic. The key difficulty for analyzing statistical QoS in a network is that, without some kind of intervention, the flows are no longer independent after they have been multiplexed at the edge node. In this section we pursue a solution where each core node has a delay jitter control mechanism that ensures a lower bound on delays [38]. Specifically, if traffic at a node experiences delay which is X seconds shorter than the assigned maximum delay, a delay jitter controller at the next node holds the traffic for X seconds before permitting it to be scheduled. Loosely, the delay jitter controllers ensure that the traffic arriving at each node is bounded by a virtual arrival process whose statistical properties are no worse than the traffic arriving at the network edge.

All network nodes run a rate-based scheduling algorithm which guarantees a minimum bandwidth to each traffic class, and each node has a separate buffer of finite size for each traffic class. Traffic which arrives to a full buffer is dropped. The length of the buffer is provisioned such that traffic is dropped only if it violates a given delay bound. Since each network node performs buffering and scheduling on a per-class basis, we refer to this approach as *class-level aggregation*. Figure 2 illustrates how traffic is processed in the network with class-level aggregation, showing the buffers and jitter controllers for some of the nodes. The conditioners ensure that all traffic flows which arrives to the network satisfy assumption (A2).

We will be able to show that networks with class-level aggregation can guarantee that (1) traffic which is not dropped in the finite-sized buffers meets a given end-to-end delay guarantee, and (2) the drop rate of traffic at each node is bounded.

3.1 Per Class Delay Jitter Control

As shown in Figure 2, core nodes have a delay jitter control mechanism which ensures that traffic experiences its maximum allocated per-node delay. More precisely, if the route of a flow traverses nodes $m_1, m_2, \ldots, m_n$, with per-node delay bounds $d_{m_1}, d_{m_2}, \ldots, d_{m_n}$, then the delay jitter controller at node m_k $(1 < k \leq n)$ holds traffic until the delay of the traffic has a delay equal to $d_{m_1} + d_{m_2} + \ldots + d_{m_{k-1}}$. The implementation of delay jitter control may require time-stamping of packets, and may incur additional buffer requirements [38].

The jitter control at core nodes ensures that all packets from the same flow experience the same fixed delays (with the exception of delays at the last node). As a consequence, traffic from a flow departing from the delay jitter controller is no worse than the traffic which arrive to the network entrance. Specifically,

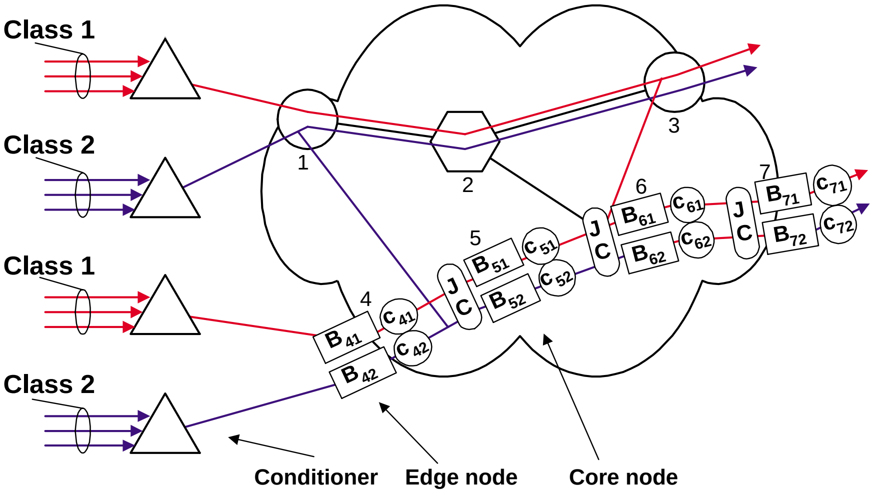


Figure 2: **Network with class-level aggregation.** At each edge node and each core node, there is one finite-length buffer for each class. Each buffer is served at a fixed rate, and arrivals to a full buffer are dropped. Core nodes have a delay jitter controller, labeled *JC* in the graph, which buffers traffic until it satisfies the maximum allocated per-node delay at the previous node. The figure illustrates the buffers and jitter controllers of nodes 4 – 7. B_{mq} indicates the buffer size and c_{mq} indicates the rate at which the buffer for class q at node m is served.

assuming that there are no losses due to buffer overflows, traffic which satisfies assumptions (A1) – (A5) at the network entrance also satisfies these assumptions at downstream nodes after passing through the corresponding delay jitter controllers. Although losses introduce correlations between flow arrivals, even when jitter control is used, we will show that this traffic can be bounded by *virtual* arrival processes in the network which assume that no traffic is lost and all traffic satisfies assumptions (A1)–(A5). Since the provisioning is done with respect to these virtual processes, the corresponding delay bounds also apply for the actual traffic with losses.

Due to the delay jitter control mechanism, the schedulers are non-workconserving. We note that it is feasible to eliminate the delay jitter control, thus, making the schedulers workconserving, and still meet the assumptions (A1) – (A5) for the traffic. Instead of holding the traffic at a delay jitter controller for X seconds before sending it to a scheduler, one can add X seconds to the maximum delay of the traffic at the node and immediately send the traffic to the scheduler [2, 15, 24, 37]. As a result, the traffic arrivals are assigned the same maximum delay at a node as if the delay jitter controllers were employed. If the arrivals are scheduled according to their deadlines, then without delay jitter control the traffic will be served in the same order as with delay jitter control. Since, without delay jitter control, the scheduler can send the traffic whenever the link is idle, we can possibly achieving lower end-to-end delays [2]. However, the absence of delay jitter control may increase the burstiness of traffic, and, hence, result in larger buffer requirements.

3.2 Rate-Based Scheduling with Per-Class Buffering

As already discussed, we assume that the scheduling algorithm at both edge and core nodes provides per-class queueing and per-class rate guarantees. Class-q traffic which arrives to a scheduler, say at node m, is inserted into a finite buffer with length B_{mq}. Arrivals to a full buffer are dropped and considered lost. The buffer for a class is served at a guaranteed minimal rate, denoted by c_{mq}. Let $\mathcal{C}_{mq}$ denote the set of flows from class q with traffic at node m and with delay bound d_{mq}. We use $A_{\mathcal{C}_{mq}}(s,t)$ to denote the total amount of class-q traffic that *would* arrive at node m in the interval $[s,t)$ if there were no losses at upstream nodes. If we ignore upstream losses at upstream nodes in such a fashion, then the delay jitter controllers ensure that the individual flows that comprise $A_{\mathcal{C}_{mq}}$ have the same statistical properties as the exogenous arrival processes A_j. In particular, assumptions (A1) – (A5) hold true for the set of flows in $A_{\mathcal{C}_{mq}}$.[3] Thus, an effective envelope $\mathcal{G}_{\mathcal{C}_{mq}}$ for the traffic from flows in $\mathcal{C}_{mq}$ can be obtained from Eqs. (5) and (6), where we set $N_q = |\mathcal{C}_{mq}|$ and A_q^* is the deterministic envelope of $A_j(t)$ $(j \in \mathcal{C}_{mq})$. We will use $A^*_{\mathcal{C}_{mq}}$ to denote the aggregate worst-case envelope of the traffic in $\mathcal{C}_{mq}$, that is, $A^*_{\mathcal{C}_{mq}}(\tau) = |\mathcal{C}_{mq}| \cdot A_q^*(\tau)$.

Next, we use the effective envelope $\mathcal{G}_{\mathcal{C}_{mq}}$ and the delay bound d_{mq} for provisioning the data rate c_{mq} and the buffer size B_{mq} for class q at node m. We select c_{mq} as the smallest number which satisfies

$$\sup_{\tau>0} \left(\mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon) - c_{mq}\tau\right) \leq c_{mq} d_{mq} \ , \tag{7}$$

and we set B_{mq} to

$$B_{mq} = c_{mq} d_{mq} \ . \tag{8}$$

The rate c_{mq} in Eqn. (7) is set such that all class-q traffic at node m satisfies delay bound d_{mq}, as long as the arrivals comply to $\mathcal{G}_{\mathcal{C}_{mq}}$, that is, $\mathcal{G}_{\mathcal{C}_{mq}}(\tau) \geq A_{\mathcal{C}_{mq}}(t, t+\tau)$ for all t and τ. Likewise, B_{mq} is set such that traffic is dropped if the delay bound d_{mq} is violated.

The next theorem states the statistical service guarantees that are attained with the above selection of c_{mq} and B_{mq}.

Theorem 2 *Given a set of flows $\mathcal{C}_{mq}$ at node m where each A_j with $j \in \mathcal{C}_{mq}$ satisfies assumptions (A1) – (A5), and given a scheduler with per-class buffering and guaranteed service rate for each class, the c_{mq} and B_{mq} are selected as in Eqs. (7) and (8), respectively, then*

1. *Traffic which is not dropped meets its delay bound d_{mq}.*
2. *The rate at which traffic is dropped at node m due to a full buffer is bounded by*

$$\varepsilon \cdot \sup_{\tau>0} \left\{ A^*_{\mathcal{C}_{mq}}(\tau) - \mathcal{G}_{mq}(\tau) \right\} \ , \tag{9}$$

under the assumption that

$$Pr\left(\sup_{\tau>0}\left\{A_{\mathcal{C}_{mq}}(t-\tau,t) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau)\right\} > 0\right) \approx \inf_{\tau>0} Pr\left(\left\{A_{\mathcal{C}_{mq}}(t-\tau,t) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau)\right\} > 0\right) . \tag{10}$$

The assumption in Eqn. (10) is similar to an assumption made in [3], as well as in related work [6, 19, 20, 21, 22]. A theoretical justification for this assumption is made in [21], and the assumption has been supported by numerical examples [3, 6, 21].

Proof: The first part of the theorem follows from the construction of the queue size B_{mq}. Since the queue is serviced at a rate of at least c_{mq}, and since we set $B_{mq} = c_{mq} d_{mq}$, traffic in the buffer cannot have a delay

[3] In the proof of Theorem 2, we use $A_{\mathcal{C}_{mq}}$ as an upper bound for the actual arrivals at node m.

larger than d_{mq}. The second part of the theorem is a result of Eqn. (7) and the definition of the effective envelope, as we now argue. Consider an arbitrary time t with a traffic arrival to the buffer. By Eqs. (7) and (8), an arrival will find a full buffer only if for some $\tau > 0$,

$$\tilde{A}_{\mathcal{C}_{mq}}(t-\tau, t) > \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon) \ , \tag{11}$$

where $\tilde{A}_{\mathcal{C}_{mq}}(s,t)$ is the class-q traffic that arrives at node m in the interval $[s,t)$. Thus, the probability that an arrival from class q at node m is dropped is

$$Pr\left(\exists \tau > 0 \ : \ \tilde{A}_{\mathcal{C}_{mq}}(t-\tau, t) > \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\right) . \tag{12}$$

Using $L_{\mathcal{C}_{mq}}(s,t)$ to denote the class-q traffic that *would have* arrived at node m but was dropped at upstream nodes, we have

$$\tilde{A}_{\mathcal{C}_{mq}}(t-\tau, t) = A_{\mathcal{C}_{mq}}(t-\tau, t) - L_{\mathcal{C}_{mq}}(t-\tau, t) \ . \tag{13}$$

Thus, the probability that an arrival from class q at node m is dropped at time t is

$$\begin{aligned} & Pr\left(\exists \tau > 0 \ : \ A_{\mathcal{C}_{mq}}(t-\tau, t) - L_{\mathcal{C}_{mq}}(t-\tau, t) > \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\right) \\ \leq \ & Pr\left(\exists \tau > 0 \ : \ A_{\mathcal{C}_{mq}}(t-\tau, t) > \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\right) && (14) \\ = \ & Pr\left(\sup_{\tau>0}\left\{A_{\mathcal{C}_{mq}}(t-\tau, t) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\right\} > 0\right) && (15) \\ \approx \ & \sup_{\tau>0} Pr\left(A_{\mathcal{C}_{mq}}(t-\tau, t) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon) > 0\right) && (16) \\ \leq \ & \varepsilon. && (17) \end{aligned}$$

Eqn. (15) is merely a different formulation of the right hand side of Eqn. (14). Eqn. (16) uses the assumption in Eqn. (10). Eqn. (17) follows from the definition of the effective envelope.

Next, we turn to the question of how much traffic is lost when an arrival finds a full queue. By construction of the buffer size, when traffic which complies to $\mathcal{G}_{\mathcal{C}_{mq}}$ arrives to the buffer, the most amount of traffic that may be dropped is bounded by $\sup_{\tau>0}\left\{A_{\mathcal{C}_{mq}}(t-\tau, t) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\right\}$. So, we have

$$\begin{aligned} \text{(Amount of dropped traffic at time } t\text{)} \ & \leq \ \sup_{\tau>0}\left\{A_{\mathcal{C}_{mq}}(t-\tau, t) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\right\} && (18) \\ & \leq \ \sup_{\tau>0}\left\{A^*_{\mathcal{C}_{mq}}(\tau) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\right\} \ . && (19) \end{aligned}$$

The second part of the theorem now follows from Eqs. (17) and (19). ■

3.3 Discussion

There are a number of discussion points to address regarding networks with class-level aggregation as presented in this section.

1. **Loss rate on a path:** Our analysis assumes that the arrivals from a flow j at each node on its path are characterized by $A_{\mathcal{C}_{mq}}$, independent of previous losses. So, our bounds do not quantify the losses that occur at consecutive nodes. As a consequence, we conservatively assume that losses on a path of

nodes occur independent of losses upstream on the path. Consider a sequence of nodes $m_1 \to m_2 \to \ldots \to m_L$, with $\mathcal{C}_{m_l q}$ the set of class-q flows at each node m_l. With the assumptions from Theorem 2, the loss rate for class q on this path is bounded by:

$$\sum_{l=1}^{L} \varepsilon \cdot \sup_{\tau > 0} \left\{ A^*_{\mathcal{C}_{m_l q}}(\tau) - \mathcal{G}_{m_l q}(\tau) \right\} . \tag{20}$$

2. **Calculation of c_{mq} and signaling overhead:** The calculation of c_{mq} and B_{mq} is dependent on the cardinality of the set $\mathcal{C}_{mq}$. Each time a new flow is added to the network, the allocation of c_{mq} and B_{mq} must be modified at all nodes on the route of the new flow. However, compared to traditional QoS approaches which maintain per-flow state information, e.g., IntServ [4] and ATM UNI 4.0 [1], the signaling overhead is small.

3. **Dynamic Routing:** In our discussion we have assumed that all traffic of a given class, traveling from specific network ingress to network egress points, traverses the network on the same fixed route. The assumption of fixed routes can be relaxed if mechanisms such as PATH messages in RSVP [13] are used.

4. **Maximum delay bound is incurred at each node:** The delay jitter control mechanisms at the nodes enforce per-node worst-case delays at all but the last node on a route. Note that the enforced end-to-end delay bounds in a network with class-level aggregation are dependent on the number of nodes traversed.

5. **Discrete packet size:** Our formal framework uses a fluid-flow interpretation of traffic. Since traffic is sent in discrete-sized packets, performance guarantees given to fluid flow traffic must be matched to guarantees for actual traffic. For rate-based scheduling algorithms the issues involved in transforming guarantees on fluid flow traffic for packet-level traffic are well understood [28, 29, 41]. For example, fluid flow guarantees have been used in the IETF to specify a guaranteed service class for packet-level traffic in the Integrated Services architecture [35].

4 Networks with Path-Level Aggregation (“Pipe Model”)

One possible disadvantage of a network with class-level aggregation, as presented in Section 3, is the requirement for delay jitter-control at each node, which seems counterintuitive from the perspective of QoS provisioning. In this section we present an alternative approach, called *path-level aggregation*, which aggregates traffic at a finer level of granularity. Here, flows are multiplexed in the same buffer only if they are in the same traffic class *and* if they traverse the network on an identical end-to-end route. We call an end-to-end route in the network which carries flows from a particular traffic class, a path or ‘pipe’.

Figure 3 illustrates a network with path-level aggregation. The figure depicts six paths (“pipes”) for two classes. At the network entrance, there is one traffic conditioner for each pipe. The traffic conditioner discards that portion of the aggregate traffic which does not comply to a given policing function. At each network node there is a separate buffer for each pipe with traffic at this node. Thus, flows in the same class are only multiplexed in the same buffer if they have the same end-to-end path. That is, networks with path-level aggregation perform traffic control separately for each ‘pipe’, and, hence, exploit statistical

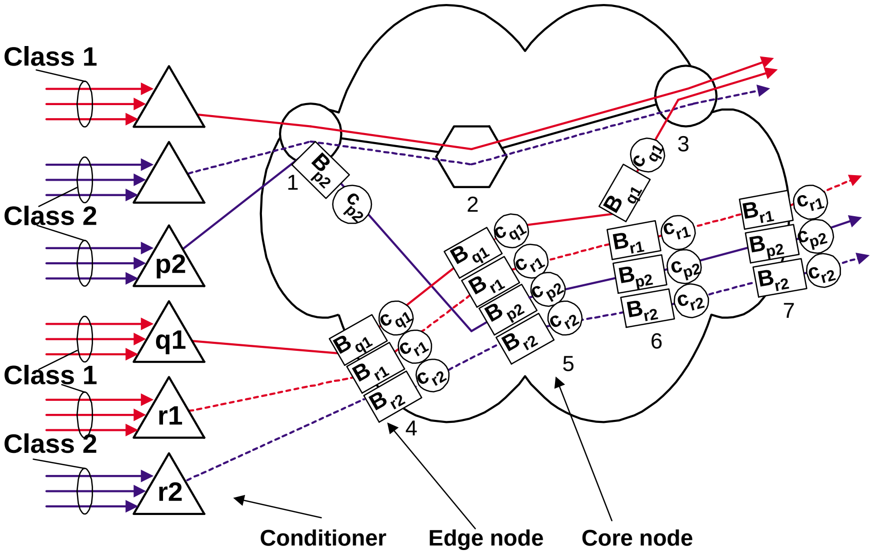


Figure 3: **Network with path-level aggregation.** An end-to-end path for a traffic class defines a path or 'pipe'. The figure depicts a total of six pipes, and depicts the buffers of four of the pipes, labeled, $p2$, $q1$, $r1$, $r2$. For each pipe, the aggregate traffic is policed at the network entrance by a conditioner. At each node there is a separate buffer for each pipe with traffic at this node. Node buffers are dimensioned such that no overflows occur.

multiplexing gain only for flows in the same pipe. In contrast to networks with class-level aggregation, network nodes in the path-level scheme do not perform delay jitter control.

4.1 Traffic Policing at Traffic Conditioners

We use $\mathcal{C}_{pq}$ to denote the set of class-q flows which travel on a path p of nodes, where a path is a unique loop-free sequence of nodes which starts and ends with edge nodes. We may compute an effective envelope $\mathcal{G}_{\mathcal{C}_{pq}}$ for this aggregate from Eqs. (5) and (6), where we set $N_q = |\mathcal{C}_{pq}|$ and A^*_q is the deterministic envelope of $A_j(t)$ ($j \in \mathcal{C}_{pq}$).

An important aspect of path-level aggregation is that the aggregate traffic from $\mathcal{C}_{pq}$ which arrives to the network is conditioned using the effective envelope $\mathcal{G}_{\mathcal{C}_{pq}}(.;\tau)$ as policing function. In other words, if $A_{\mathcal{C}_{pq}}(t, t+\tau)$ denotes the aggregate traffic from class $\mathcal{C}_{pq}$ that is admitted into the network in the time interval $[t, t+\tau)$, the policing function $\mathcal{G}_{\mathcal{C}_{pq}}$ ensures that

$$A_{\mathcal{C}_{pq}}(t, t+\tau) \leq \mathcal{G}_{\mathcal{C}_{pq}}(\tau) \qquad \forall t \geq 0, \forall \tau \geq 0 \, . \tag{21}$$

Traffic in excess of $\mathcal{G}_{\mathcal{C}_{pq}}$ is discarded by the conditioner or at least marked with a lower priority, e.g., as best effort traffic.

For this architecture, Theorem 3 below provides a bound for the traffic from a pipe which is dropped at the network entrance.

Theorem 3 *Given a set of flows $\mathcal{C}_{pq}$, let $A_j(t, t+\tau)$ for $j \in \mathcal{C}_{pq}$ satisfy assumptions (A1) – (A5). If the arrivals $A_{\mathcal{C}_{pq}}$ are policed according to Eqn. (21), then the rate of dropped traffic is bounded by*

$$\varepsilon \cdot \sup_{\tau > 0} \left\{ A^*_{\mathcal{C}_{pq}}(\tau) - \mathcal{G}_{\mathcal{C}_{pq}}(\tau) \right\} \ , \tag{22}$$

with the assumption that

$$Pr\left(\sup_{\tau>0}\left\{A_{\mathcal{C}_{pq}}(t-\tau,t)-\mathcal{G}_{\mathcal{C}_{pq}}(\tau)\right\}>0\right)\approx\inf_{\tau>0}Pr\left(\left\{A_{\mathcal{C}_{pq}}(t-\tau,t)-\mathcal{G}_{\mathcal{C}_{pq}}(\tau)\right\}>0\right)\ . \tag{23}$$

Proof: The assumption in Eqn. (23) is the same as in Eqn. (10). The proof is conducted in a similar fashion as the proof of Theorem 2. The probability that an arrival results in dropped traffic is given by

$$\begin{aligned} Pr\left(\exists\tau>0\ :\ A_{\mathcal{C}_{pq}}(t-\tau,t)>\mathcal{G}_{\mathcal{C}_{pq}}(\tau)\right) &= Pr\left(\sup_{\tau>0}\left\{A_{\mathcal{C}_{pq}}(t-\tau,t)-\mathcal{G}_{\mathcal{C}_{pq}}(\tau)\right\}>0\right) &(24)\\ &\approx \sup_{\tau>0}Pr\left(\left\{A_{\mathcal{C}_{pq}}(t-\tau,t)-\mathcal{G}_{\mathcal{C}_{pq}}(\tau)\right\}>0\right) &(25)\\ &\leq \varepsilon\ . &(26)\end{aligned}$$

The maximum amount of traffic that is lost at any instant in time t is given by

$$\sup_{\tau>0}\left\{A_{\mathcal{C}_{pq}}(t-\tau,t)-\mathcal{G}_{\mathcal{C}_{pq}}(\tau)\right\}\ .$$

Since

$$\sup_{\tau>0}\left\{A_{\mathcal{C}_{pq}}(t-\tau,t)-\mathcal{G}_{\mathcal{C}_{pq}}(\tau)\right\}\leq\sup_{\tau>0}\left\{A^*_{\mathcal{C}_{pq}}(t-\tau,t)-\mathcal{G}_{\mathcal{C}_{pq}}(\tau)\right\}\ , \tag{27}$$

the theorem follows from Eqn. (26). ■

4.2 Scheduling at Edge Nodes and at Core Nodes

With path-level aggregation, allocation of bandwidth and buffer space at a node is done separately for each 'pipe.' At each node on the route of a pipe, the same buffer size and bandwidth is allocated. We use c_{pq} to denote the rate which is allocated at each node on the route of the pipe, and we use B_{pq} to denote the reserved buffer space. We set c_{pq} as the smallest number which satisfies

$$\sup_{\tau>0}\left(\mathcal{G}_{\mathcal{C}_{pq}}(\tau;\varepsilon)-c_{pq}\tau\right)\leq c_{pq}d_{pq}\ , \tag{28}$$

where d_{pq} is the end-to-end delay bound for $\mathcal{C}_{pq}$, and we set the buffer space B_{pq} according to

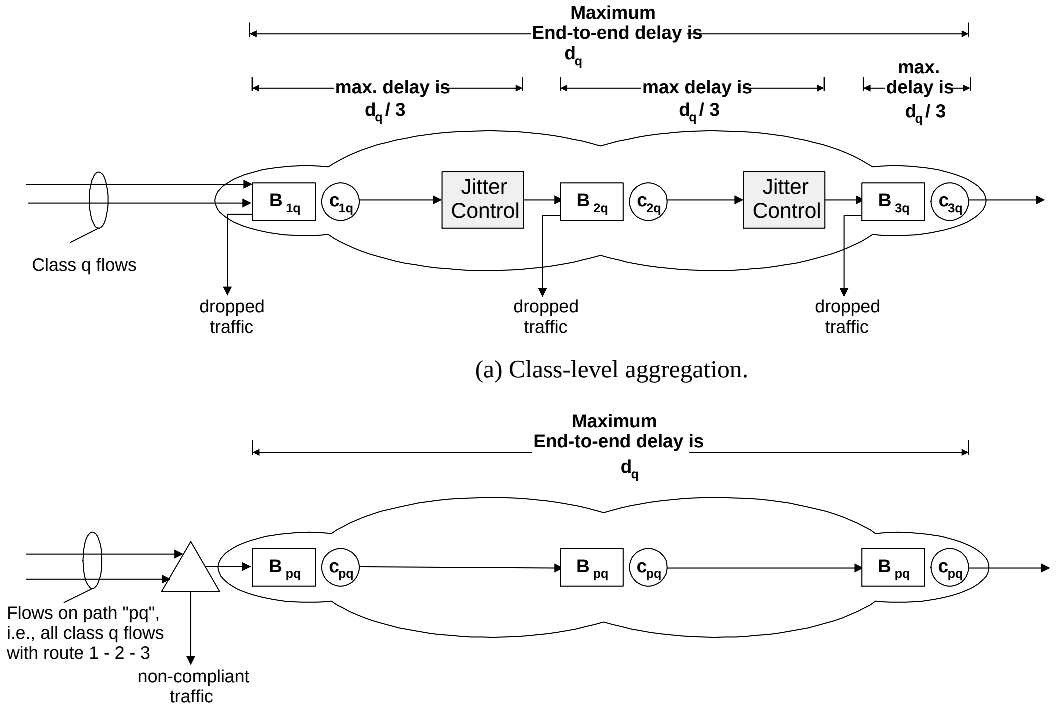
$$B_{pq}=c_{pq}d_{pq}\ . \tag{29}$$

The next theorem states properties for the end-to-end performance of flows in $\mathcal{C}_{pq}$ with end-to-end delay bound d_{pq}.

Theorem 4 *Given a set of flows $\mathcal{C}_{pq}$ where each A_j with $j\in\mathcal{C}_{pq}$ satisfies assumptions (A1) – (A5), and assuming the existence of policing functions at network ingress points that enforce Eqn. (21), if c_{pq} and B_{pq} are allocated as given in Eqs. (28) and (29) at each node on the route taken by flows in $\mathcal{C}_{pq}$, then*

1. *No traffic is dropped inside the network.*

2. *The end-to-end delay of traffic satisfies delay bound d_{pq}.*



(a) Class-level aggregation.

(b) Path-level aggregation.

Figure 4: Comparison of the class-level and path-level design, with a common end-to-end delay bound.

Proof: The proof is done using Cruz's service curves [8]. Assume the flows in $\mathcal{C}_{pq}$ take the route $k = m_1 \rightarrow m_2 \rightarrow \ldots \rightarrow m_K = l$. Allocating a capacity of c_{pq} to $\mathcal{C}_{pq}$ is equivalent to guaranteeing a service curve of

$$S_{m_i}(\tau) = c_{pq}\ \tau \qquad\qquad \forall \tau \geq 0,\ \forall i\ (1 \leq i \leq K)\ . \tag{30}$$

If we use S_p to denote the network service curve we obtain:

$$\begin{aligned} S_p(\tau) &= S_{m_1} \star S_{m_2} \star \ldots \star S_{m_K}(\tau) && (31)\\ &= c_{pq}\ \tau\ , && (32) \end{aligned}$$

where '$\star$' in Eqn. (31) is the convolution operator defined as $G \star H(t) = \inf_{\tau} \{G(\tau) + H(t - \tau)\}$. Eqn. (32) follows from repeated application of the convolution operator. According to Theorem 4 in [8], the maximum end-to-end delay bound, denoted as D_{max}, is bounded by the network service curve S_p and the bound on the arrivals $\mathcal{G}_{\mathcal{C}_{pq}}$ by

$$D_{max} \leq \sup_{\tau > 0} \inf_{\Delta \geq 0} \left\{ \Delta \mid \mathcal{G}_{\mathcal{C}_{pq}}(\tau; \varepsilon) \leq c_{pq}(\tau + \Delta) \right\} . \tag{33}$$

Now, the theorem follows from Eqs. (28) and (29). ■

4.3 Comparing Class-level Aggregation versus Path-level Aggregation

We conclude this section with a comparison of class-level aggregation and path-level aggregation. Class-level aggregation and path-level aggregation instantiate a fundamentally different trade-off between the ability to provision low delay bounds and the ability to yield a high multiplexing gain. Class-level aggregation accounts for multiplexing of all flows from the same class, whereas path-level aggregation merely accounts for the multiplexing of flows in the same ‘pipe’. Hence, we expect the multiplexing gain of class-level aggregation to be better than that of path-level aggregation. On the other hand, class-level aggregation requires a jitter control mechanism at each node, which holds packets which are ‘too early’ with respect to the per-node delay bounds. The delay jitter control enforces that traffic experiences the maximum node delay at each node (except the last node on the path), may contribute to higher end-to-end delays.

In Figure 4 we present a comparison of the two approaches. We illustrate a network with three nodes. We set the end-to-end delay bound to d_q in both networks. With class-level aggregation (Figure 4(a)), the delay jitter controllers at the second and the third node ensure that all packet arrivals to these nodes experience the same delays. Assuming that the per-node delay bounds are set such that the end-to-end delay bound d_q is distributed evenly across all nodes, the jitter controller at the second node enforces that all traffic which is forwarded to the scheduler has a delay of $d_q/3$. Likewise, the jitter controller at the third node enforces a delay of $2/3d_q$. Thus, the end-to-end delay of all traffic sent over the three-node network has a lower bound of $2/3d_q$ and an upper bound of d_q.

In Figure 4(b), we show the same example for a network with path-level aggregation. The path, referred to as ‘path pq’, consists of the class-q flows on the route $1 \rightarrow 2 \rightarrow 3$. The allocation of buffers and bandwidth for the path is identical at all nodes. As in Figure 4(a), the end-to-end delay bound is set to d_q. Note that due to [29] it is not necessary to specify per-node delay bounds with path-level aggregation.

For provisioning QoS guarantees in a general network, the trade-off presented by the class-level and path-level approaches is not immediately obvious. Thus, a numerical evaluation of the approaches is needed to determine which of the two presented approaches is more effective.

5 Numerical Evaluation

In this section, we present numerical examples to compare the ability of class-level and path-level aggregation to support statistical end-to-end delay guarantees. Class-level aggregation considers the multiplexing of larger groups of flows than the path-level approach and is thus expected to yield a better multiplexing gain. On the other hand, delay jitter control in networks with path-level aggregation may result in better delay bounds. In the numerical examples in this section, we will contrast the results for four different approaches for provisioning QoS.

- **Statistical QoS with class-level aggregation:** The bandwidth and buffer allocation is as given in Eqs. (7) and (8). The QoS guarantee for class-q is as stated in Theorem 2. The calculation of the effective envelope is done as discussed in Section 2.2.

- **Statistical QoS with path-level aggregation:** The bandwidth and buffer allocation is as given in Eqs. (28) and (29). The QoS guarantee for class-q is as stated in Theorems 3 and 4.

- **Deterministic QoS:** In our context, a service with deterministic QoS satisfies an end-to-end delay bound of d_q for all class-q traffic. Let $\mathcal{C}_{mq}$ be the set of class-q flows at node m. Select c_{mq} as the smallest value such that $\sup_{\tau>0}\left\{A^*_{\mathcal{C}_{mq}}(\tau) - c_{mq}\tau\right\} \le c_{mq}d_q$, where $A^*_{\mathcal{C}_{mq}}(\tau) = |\mathcal{C}_{mq}| \cdot A^*_q(\tau)$. From [29] we know that, if all nodes m allocate a bandwidth of c_{mq}, an end-to-end delay bound of d_q can be guaranteed for all class-q traffic.

Class	Type		Burst Size	Mean Rate	Peak Rate	End-to-End Delay Bound
	burstiness	delay	σ_q (bits)	ρ_q (Mbps)	P_q (Mbps)	d_q (msec)
1	low	low	10000	0.15	6.0	10
2	low	high	10000	0.15	6.0	40
3	high	low	100000	0.15	6.0	10
4	high	high	100000	0.15	6.0	40

Table 1: Parameters of four traffic classes.

- **Average Rate Allocation:** This scheme allocates bandwidth equal to the average traffic rate for flows. So, if $\mathcal{C}_{mq}$ is the set with class q flows at node m, node m allocates a rate equal to $|\mathcal{C}_{mq}|\rho_q$ for class q. (Recall that $\rho_q = \lim_{\tau\to\infty} A_q^*(\tau)/\tau$). Average rate allocation only guarantees finite delays and average throughput, but no per-flow or per-class QoS.

As an admission control condition, we require for all schemes above that the total allocated bandwidth on a link must not exceed the link capacity.

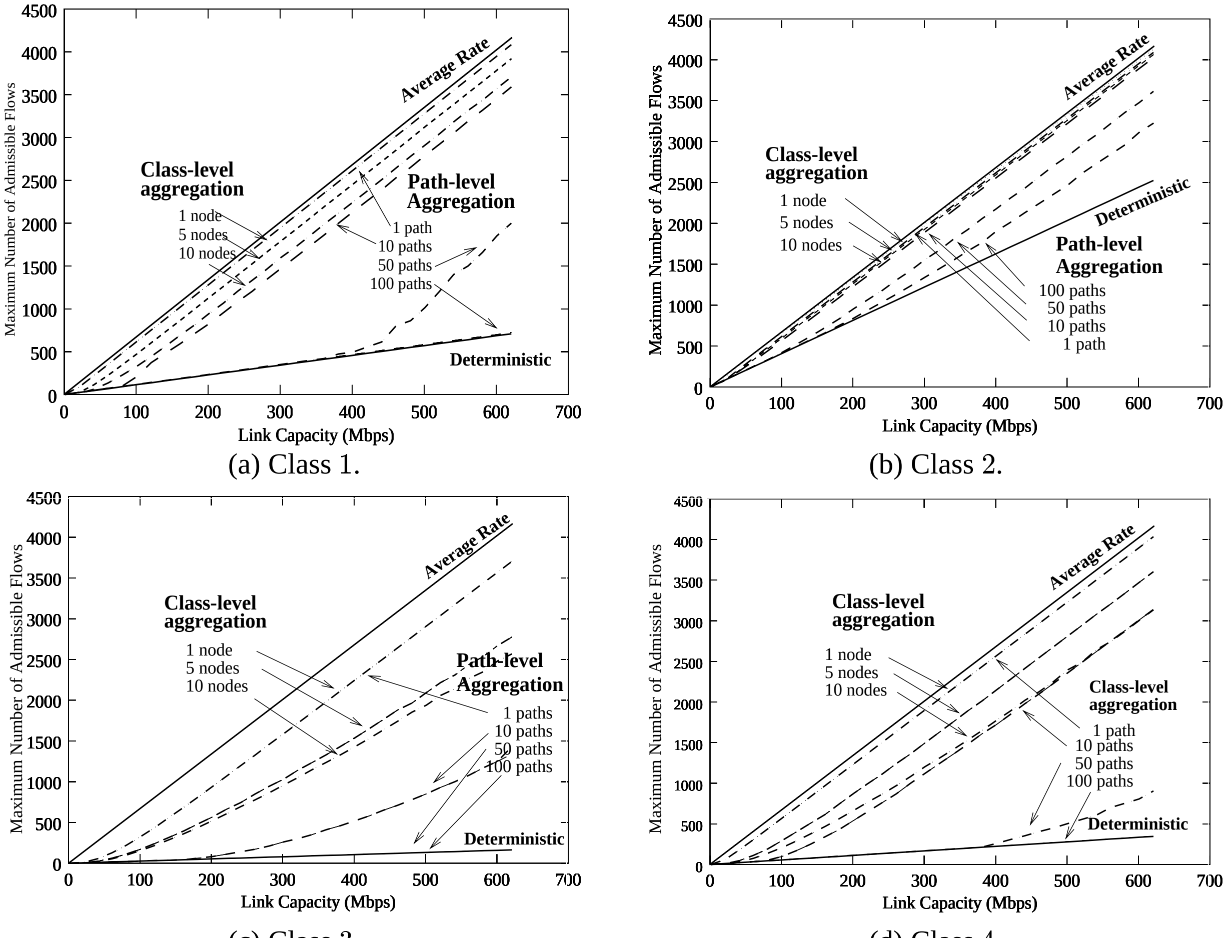
We consider four classes of traffic, and assume that traffic flows in each class are regulated by a peak-rate constrained leaky bucket with parameters (σ_q, ρ_q, P_q), and deterministic envelope $A_q^*(\tau) = \min(P_q\tau, \sigma_q + \rho_q\tau)$ for class q. The parameters for the flow classes are given in Table 1. We set $\varepsilon = 10^{-6}$ in all our examples. The parameters are similar to those used in other studies on regulated adversarial traffic [12, 25, 31].

5.1 Maximum Number of Admissible Flows

In Figures 5(a)-(d) we plot the maximum number of flows which can be provisioned with QoS on a link in the network, as a function of the link capacity. We vary the link capacity in the range 1 Mbps – 622 Mbps.

For class-level aggregation, first recall from our discussions in Section 3.3 and Section 4.2 that, due to delay-jitter control, the end-to-end delay bound is dependent on the number of links. So, if the end-to-end delay bound is given by $d_q = 10$ msec, and the path length is given by L, the per-node delay bound is given by $10/L$ msec (Here, we divide the end-to-end delay bound evenly among all nodes). Thus, the longer the route of a flow, the smaller the per-node delay bound, and, consequently, the smaller the total number of flows that can be accommodated on a link. In Figures 5(a)-(d), we consider path lengths equal to $L = 1, 5,$ and 10 nodes. As the figures show, class-level aggregation yields a significant multiplexing gain, even for longer path lengths. For traffic classes 1 and 2, which exhibit lower burstiness, the number of admissible flows with class-level aggregation is close to that of the admissible number of flows with an average rate allocation, even when the length of the route grows as large as 10 nodes.

For path-level aggregation, the achievable multiplexing gain is dependent on the number of paths (pipes) that extend from a given ingress point. In our examples, we only consider one class at a time, so the number of paths at a node is given by the number of different end-to-end routes. In a network with M edge nodes, the maximum number of paths at any core node is given by $O(M^2)$. Since path-level aggregation only performs multiplexing of flows on the same path, the number of flows which can be multiplexed on a link decreases with M. In Figures 5(a)-(d), we show the results for path-level aggregation with $1, 10, 50,$ and 100 paths. (Note that the maximum number of flows that can be supported with path-level aggregation with only 1 path is identical to class-level aggregation with a 1 hop route.) Figures 5(a)-(d) show that the maximum



(a) Class 1.

(b) Class 2.

(c) Class 3.

(d) Class 4.

Figure 5: Maximum number of admissible flows.

number of flows which can be provisioned with QoS quickly deteriorates as the number of paths increases. For 100 paths in the network, we observe for all traffic classes that path-level aggregation accommodates the same number of flows as deterministic QoS.

In summary, the performance of path-level aggregation quickly decreases as the number of paths increases. Since the number of paths grows (in the worst case) with the square of the number of edge nodes in a network, path-level aggregation appears to be a viable technique only in small networks. Class-level aggregation, on the other hand, even though it is sensitive to the length of the routes, yields a high statistical multiplexing gain throughout.

5.2 Comparison of the Traffic Loss Rate

Next, we compare the expected loss rate of flows. Recall that the loss rate as given in Eqs. (20) and (22) is for traffic classes, and not for single flows. Making QoS guarantees to aggregate flows yields a simple network core, since no per-flow information is required. On the other hand, single flows may experience a service which is worse than the service guaranteed to the class as a whole. To obtain a measure on the loss rate, we normalize the loss rate of a class by the long time average of the expected traffic in the class. (We

Table 2: Normalized loss rate per flow. (For class-level aggregation, the loss rate is dependent on the path length.)

	Class-level Aggregation			Path-level
	1 node	2 nodes	10 nodes	Aggregation
Classes 1, 2	$6.7 \cdot 10^{-8}$	$1.3 \cdot 10^{-7}$	$6.7 \cdot 10^{-7}$	$6.7 \cdot 10^{-8}$
Classes 3, 4	$6.7 \cdot 10^{-7}$	$1.3 \cdot 10^{-6}$	$6.7 \cdot 10^{-6}$	$6.7 \cdot 10^{-8}$

point out that this 'normalization' does not give a precise loss rate.) Under the assumption that $\mathcal{G}_{mq}$ is the set of class-q flows at all nodes, we obtain with Eqn. (20) a (normalized) loss rate of

$$\text{Loss Rate}^{class} < \quad \frac{1}{\rho_q \cdot |\mathcal{C}_{mq}|} \cdot L \cdot \varepsilon \cdot \sup_{\tau > 0} \left\{ A^*_{\mathcal{C}_{mq}}(\tau) - \mathcal{G}_{mq}(\tau) \right\} . \tag{34}$$

In our example, since $A^*_{\mathcal{C}_{mq}}(\tau) = |\mathcal{C}_{mq}| \cdot \min\left(P_q \tau, \sigma_q + \rho_q \tau\right)$ and with $\mathcal{G}_{mq}(\tau) \geq \rho_q \tau \cdot |\mathcal{C}_{mq}|$ we obtain the bound

$$\text{Loss Rate}^{class} < \frac{\sigma_q}{\rho_q} \varepsilon L . \tag{35}$$

The same consideration for the path-level scheme yield

$$\text{Loss Rate}^{path} < \frac{\sigma_q}{\rho_q} \varepsilon . \tag{36}$$

In Table 2 we give the results for bounds on the normalized loss rate for all classes used in our examples. The total loss rate is small in all cases, and is of the same order as ε.

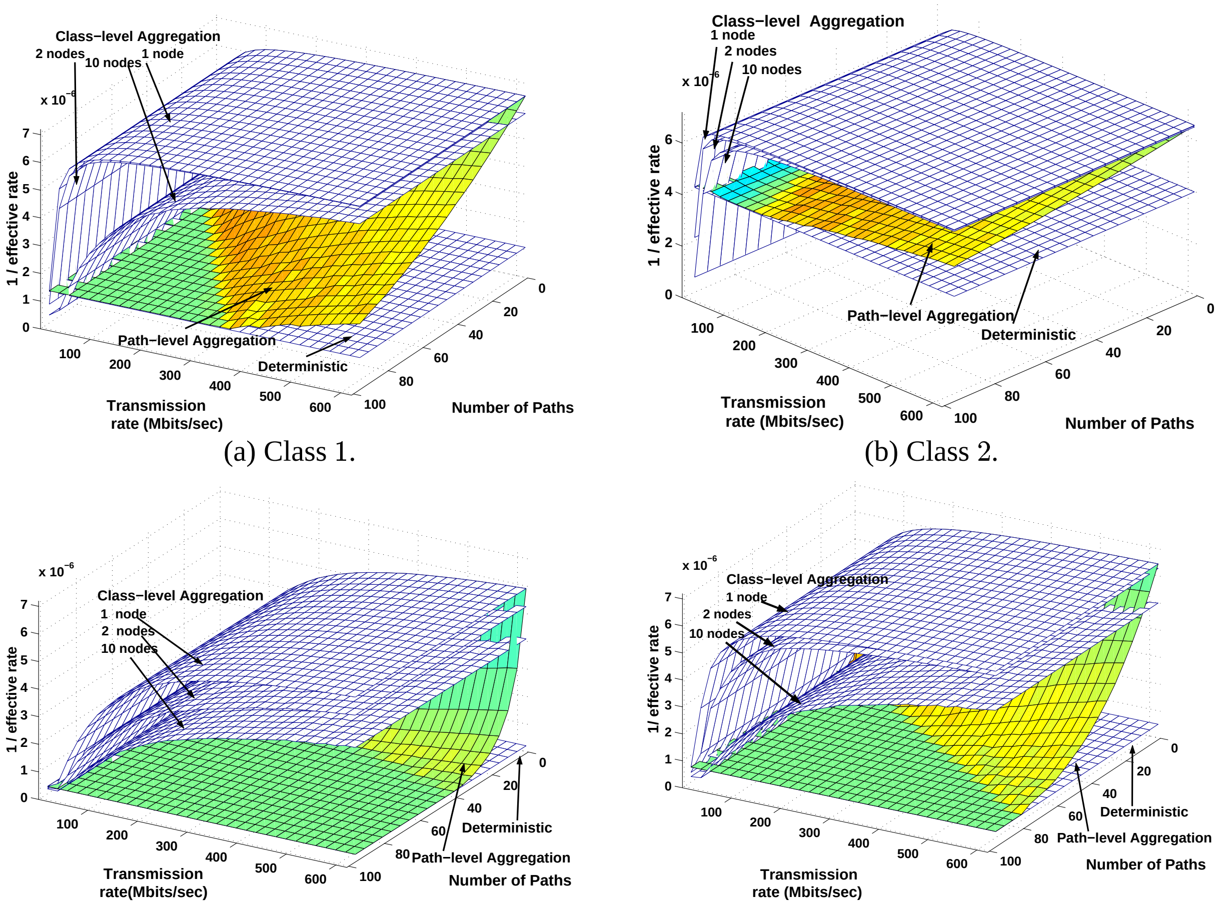
5.3 Sensitivity of Path-Level Aggregation to the Number of Paths

In Figure 5 we saw that path-level aggregation resulted in relatively poor achievable utilization at a link, when the number of paths (routes) in the network was high. Here we provide more insight into the sensitivity of path-level aggregation towards an increase of the number of paths.

We use as performance measure the rate that is allocated per flow to support the desired QoS level on a saturated link. We call this measure the *effective rate* of a flow. The effective rate is determined by first calculating the maximum number of flows which can be provisioned on a link with a desired QoS level (deterministic, statistical with class-level aggregation, statistical with path-level aggregation), and then dividing the link capacity by the number of flows.

Figures 6(a)-(d) show the results for traffic classes 1 through 4. For illustrative purposes, we plot the values of $1/(\text{effective rate})$ as a function of the link capacity and as a function of the number of paths.

Figures 6(a)-(d) show the results for deterministic QoS and for the two statistical QoS schemes considered here. A larger value of $1/(\text{effective rate})$ indicates a better statistical multiplexing gain. The effective rate of a flow with deterministic QoS is not sensitive to increases in link capacity or in the number of paths. For QoS with class-level aggregation, the statistical multiplexing gain increases with the link capacity, but does not increase with the number of paths. However, as discussed earlier, the achievable statistical multiplexing gain is dependent on the length of a route. In Figures 6(a)-(d) we include plots for route lengths of 1, 2, and 10.



(a) Class 1.

(b) Class 2.

(c) Class 3.

(d) Class 4.

Figure 6: Results for $1/c^{eff}$, where c^{eff} is the effective rate of a flow as a function of the link capacity and the number of paths in the network.

The results for path-level aggregation are perhaps the most interesting aspect of Figures 6(a)-(d). We see that a high level of statistical multiplexing gain is achievable only if the link capacity is high, and the number of paths is small. Since the number of paths can grow as fast as the square of the number of edge nodes, the multiplexing gain achievable in network deteriorates quickly as the number of paths grows large.

6 Discussion and Conclusions

In this paper we have studied two designs for networks with end-to-end statistical service guarantees: class-level aggregation (Section 3) and path-level aggregation (Section 4). The class-level approach can achieve a very high level of aggregation, resulting in a good statistical multiplexing gain; however, this comes at the expense of requiring delay jitter control for restoring the statistical independence of the flows at each node. Thus, it is required in this scheme to assign a maximum allowable delay to each node on the path for an end-to-end flow. The need for jitter control is admittedly a counterintuitive notion which we believe is justified by the high level of achievable statistical multiplexing gain. In the alternative, path-level scheme, there is no need for delay jitter control, since flows in this design are multiplexed only if they are of the same class *and* they share the same path through the network. Consequently, statistical multiplexing gain is perceived only at the network ingress points, at a much lower level of aggregation. The tradeoff between the two designs is one of enforced delay (and design complexity) in the form of delay jitter control for high levels of achievable statistical multiplexing gain.

Our numerical results indicate that the increased statistical multiplexing gain achievable with class-level aggregation is worth the price paid in terms of enforced delay. In the path-level aggregation design, as the number of paths in the network increases, the achievable statistical multiplexing gain quickly diminishes to the achievable multiplexing gain in making deterministic (worst-case) QoS guarantees. Thus, we assert that the class-level scheme is the preferred approach for implementing statistical end-to-end delay guarantees, especially considering the possibility of a work-conserving implementation of the model as discussed briefly at the end of Section 3.1.

There are a number of important issues that have to be addressed before the class-level design can be implemented in a real network. First, we must reconcile our assumption of fixed routes with dynamic routing which is prevalent in the Internet today. We need to develop appropriate data structures and algorithms that allow rapid computation of guaranteed rates and buffer allocations within the network. We need to formulate a packet-level version of our fluid-model constructs, in order for a real implementation to be possible. Here, we expect that the well-known approach from [29] will be sufficient. Finally, we plan to address the issue of how to characterize traffic flows in terms of worst-case bounding functions. Our development so far has rested on the assumption that a worst-case, subadditive bounding function is available for each traffic flow. If bounding functions for flows are not available a priori, it becomes essential to estimate these bounds from data.

References

[1] ATM Forum, ATM Forum Traffic Management Specification Version 4.0, April 1996.

[2] M. Andrews. Probabilistic end-to-end delay bounds for earliest deadline first scheduling. In *Proc. IEEE Infocom 2000*, pages 603–612, Tel Aviv, March 2000.

[3] R. Boorstyn, A. Burchard, J. Liebeherr, and C. Oottamakorn. Statistical service assurances for traffic scheduling algorithms. *IEEE Journal on Selected Areas in Communications. Special Issue on Internet QoS*, 2000. (to appear).

[4] R. Braden, D. Clark, and S. Shenker. Integrated services in the internet architecture: an overview. IETF RFC 1633, July 1994.

[5] C. S. Chang. Stability, queue length, and delay of deterministic and stochastic queueing networks. *IEEE Transactions on Automatic Control*, 39(5):913–931, May 1994.

[6] J. Choe and Ness B. Shroff. A central-limit-theorem-based approach for analyizing queue behavior in high-speed network. *IEEE/ACM Transactions on Networking*, 6(5):659–671, October 1998.

[7] R. Cruz. A calculus for network delay, Part I : Network elements in isolation. *IEEE Transaction of Information Theory*, 37(1):114–121, 1991.

[8] R. Cruz. Quality of service guarantees in virtual circuit switched networks. *IEEE Journal on Selected Areas in Communications*, 13(6):1048–1056, August 1995.

[9] R. L. Cruz. A Calculus for Network Delay, Part II: Network Analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, January 1991.

[10] B. T. Doshi. Deterministic rule based traffic descriptors for broadband ISDN: Worst case behavior and connection acceptance control. In *International Teletraffic Congress (ITC)*, pages 591–600, 1994.

[11] N.G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan, and J. E. Van der Merwe. A flexible model for resource management in virtual private networks. In *Proceedings of ACM SIGCOMM'99*, pages 95–108, Boston, MA, September 1999.

[12] A. Elwalid, D. Mitra, and R. Wentworth. A new approach for allocating buffers and bandwidth to heterogeneous, regulated traffic in an ATM node. *IEEE Journal on Selected Areas in Communications*, 13(6):1115–1127, August 1995.

[13] R. Braden et. al. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. IETF RFC 2205, September 1997.

[14] S. Blake et. al. An architecture for differentiated services. IETF RFC 2475, December 1998.

[15] D. Ferrari and D. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, April 1990.

[16] P. Goyal and H. M. Vin. Statistical delay guarantee of virtual clock. In *Proc. of IEEE Real-time Systems Symposium (RTSS)*, December 1998.

[17] G. Kesidis and T. Konstantopoulos. Extremal shape-controlled traffic patterns in high-speed networks. Technical Report 97–14, ECE Technical Report, University of Waterloo, December 1997.

[18] G. Kesidis and T. Konstantopoulos. Extremal traffic and worst-case performance for queues with shaped arrivals. In *Proceedings of Workshop on Analysis and Simulation of Communication Networks*, Toronto, November 1998.

[19] E. Knightly. H-BIND: A new approach to providing statistical performance guarantees to VBR traffic. In *Proceedings of IEEE INFOCOM'96*, pages 1091–1099, San Francisco, CA, March 1996.

[20] E. Knightly. Enforceable quality of service guarantees for bursty traffic streams. In *Proceedings of IEEE INFOCOM'98*, pages 635–642, San Francisco, March 1998.

[21] E. W. Knightly and Ness B. Shroff. Admission control for statistical QoS: Theory and practice. *IEEE Network*, 13(2):20–29, March/April 1999.

[22] J. Kurose. On computing per-session performance bounds in high-speed multi-hop computer networks. In *ACM Sigmetrics'92*, pages 128–139, 1992.

[23] C. Li and E. Knightly. Coordinated network scheduling: A framework for end-to-end services. In *Proc. of IEEE ICNP 2000, Osaka, Japan*, November 2000.

[24] J. Liebeherr and E. Yilmaz. Work-conserving vs. Non-workconserving Packet Scheduling: An Issue Revisited. In *Proc. IEEE/IFIP Seventh International Workshop on Quality of Service (IWQoS '99)*, pages 248–256, London, June 1999.

[25] F. LoPresti, Z. Zhang, D. Towsley, and J. Kurose. Source time scale and optimal buffer/bandwidth tradeoff for regulated traffic in an ATM node. In *Proceedings of IEEE INFOCOM'97*, pages 676–683, Kobe, Japan, April 1997.

[26] P. Oechslin. Worst Case Arrivals of Leaky Bucket Constrained Sources: The Myth of the On-Off source. In *Proc. IEEE/IFIP Fifth International Workshop on Quality of Service (IWQoS '97)*, pages 67–76, New York, May 1997.

[27] A. Papoulis. *Probability, Random Variables, and Stochastic Processes. 3rd edition.* McGraw Hill, 1991.

[28] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control - the single node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[29] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.

[30] J. Qiu and E. Knightly. Inter-class resource sharing using statistical service envelopes. In *Proc. IEEE Infocom '99*, pages 36–42, March 1999.

[31] S. Rajagopal, M. Reisslein, and K. W. Ross. Packet multiplexers with adversarial regulated traffic. In *Proceedings of IEEE INFOCOM'98*, pages 347–355, San Francisco, March 1998.

[32] M. Reisslein, K. W. Ross, and S. Rajagopal. Guaranteeing statistical QoS to regulated traffic: The multiple node case. In *Proceedings of 37th IEEE Conference on Decision and Control (CDC)*, pages 531–531, Tampa, December 1998.

[33] M. Reisslein, K. W. Ross, and S. Rajagopal. Guaranteeing statistical QoS to regulated traffic: The single node case. In *Proceedings of IEEE INFOCOM'99*, pages 1061–1062, New York, March 1999.

[34] J. Roberts, U. Mocci, and J. Virtamo (Eds.). *Broadband Network Traffic: Performance Evaluation and Desgin of Broadband Multiservice Networks. Final Report of Action. COST 242. (Lecture Notes in Computer Science. Vol. 1152)*. Springer Verlag, 1996.

[35] S. Shenker, C. Partridge, and R. Guerin. Specification of guaranteed quality of service. IETF RFC 2212, September 1997.

[36] D. Starobinski and M. Sidi. Stochastically bounded burstiness for communication networks. In *Proc. IEEE Infocom '99*, pages 36–42, March 1999.

[37] I. Stoica and Hui Zhang. Providing guaranteed services without per flow management. In *Proceedings of ACM SIGCOMM'99*, pages 81–94, Boston, MA, September 1999.

[38] D. Verma, H. Zhang, and D. Ferrari. Guaranteeing delay jitter bounds in packet switching networks. In *Proceedings of Tricomm'91*, pages 35–46, Chapel Hill, North Carolina, April 1991.

[39] D. Wrege, E. Knightly, H. Zhang, and J. Liebeherr. Deterministic delay bounds for VBR video in packet-switching networks: Fundamental limits and practical tradeoffs. *IEEE/ACM Transactions on Networking*, 4(3):352–362, June 1996.

[40] O. Yaron and M. Sidi. Performance and stability of communication networks via robust exponential bounds. *IEEE/ACM Transactions on Networking*, 1(3):372–385, June 1993.

[41] H. Zhang. Providing end-to-end performance guarantees using non-work-conserving disciplines. *Computer Communications: Special Issue on System Support for Multimedia Computing*, 18(10):769–781, October 1995.

[42] H. Zhang and E. Knightly. Providing end-to-end statistical performance guarantees with bounding interval dependent stochastic models. In *Proceedings of ACM SIGMETRICS'94*, pages 211–220, Nashville, TN, May 1994.

[43] Z.-L. Zhang, Z. Liu, J. Kurose, and D. Towsley. Call admission control schemes under the generalized processor sharing scheduling discipline. *Telecommunication Systems*, 7(1-3):125–152, July 1997.

[44] Z.-L. Zhang, D. Towsley, and J. Kurose. Statistical analysis of generalized processor sharing scheduling discipline. *IEEE Journal on Selected Area in Communications*, 13(6):1071–1080, August 1995.