Telecommunication Systems 23:1,2, 9–34, 2003

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

STEPHEN D. PATEK
*Department of Systems and Information Engineering, University of Virginia, Charlottesville, VA 22904, USA*

JÖRG LIEBEHERR and ERHAN YILMAZ
*Department of Computer Science, University of Virginia, Charlottesville, VA 22904, USA*

**Abstract.** Recent research on statistical multiplexing has provided many new insights into the achievable multiplexing gain in QoS networks. However, usually, these results are stated in terms of the gain experienced at a single switch, and 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*. These designs illustrate an inherent trade-off between 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 statistical multiplexing gain achievable by the two designs. The class-level aggregation design is shown to yield very high achievable link utilizations while simultaneously achieving desired statistical guarantees on delay.

**Keywords:** statistical multiplexing, statistical service, quality-of-service

## 1. Introduction

In recent years a lot of effort has gone into devising algorithms to support either deterministic or statistical QoS guarantees in packet networks. *Deterministic services* [Ferrari and Verma, 14], which guarantee worst-case end-to-end delay bounds for traffic [Cruz, 8,10; Parekh and Gallager, 27,28], are known to lead to an inefficient use of network resources [Wrege et al., 38]. *Statistical services* [Ferrari and Verma, 14], 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*

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

*gain*, expressed as

$$\begin{pmatrix}\text{Resources needed to}\\ \text{support statistical}\\ \text{QoS for } N \text{ flows}\end{pmatrix} \ll N \times \begin{pmatrix}\text{Resources needed to}\\ \text{support statistical}\\ \text{QoS for 1 flow}\end{pmatrix}.$$

Ideally, the statistical multiplexing gain of a statistical service increases with the volume of traffic so that the per-flow allocation of resources approaches 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 [Boorstyn et al., 3; Doshi, 11; Elwalid et al., 13; Kesidis and Konstantopoulos, 16,17; Knightly, 19; LoPresti et al., 24; Oechslin, 25, Rajagopal et al., 30, Reisslein et al., 31,32]. 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 [Boorstyn et al., 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. We gain insight into this issue by analyzing two extreme conceptual designs which avoid the problem of correlated downstream flows. Our main purpose is to explore the fundamental tradeoffs that arise in designing networks for end-to-end statistical QoS.

### *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 traffic characteristics and the same QoS requirements. We assume that the traffic from each individual 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 [Shenker et al., 34; Zhang, 40].

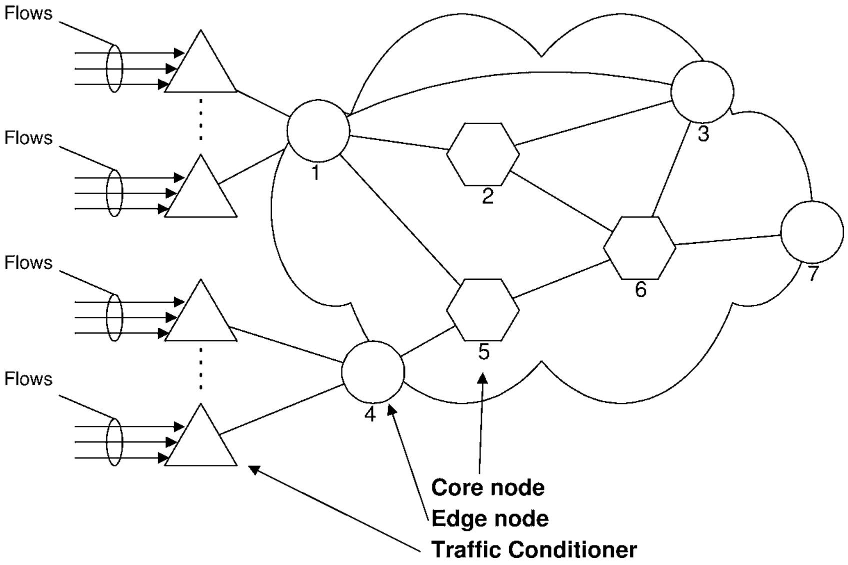


Figure 1. Network architecture.

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 discussions will investigate the trade-offs presented by the two schemes. A comparison of the approaches will allow us to make recommendations about 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. Our focus on aggregate QoS is consistent with recent efforts of the IETF in standardizing an architecture for differentiated services. 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 [Knightly and Shroff, 20; Roberts et al., 33] 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 network lies in addressing the complex correlation of traffic at downstream multiplexing points. One track of research in this area attempts to achieve a characterization of correlated traffic inside a network [Chang, 6; Kurose, 21; Starobinski and Sidi, 35; Yaron and Sidi, 39]. An alternative approach 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 at least two ways to reconstruct characteristics of traffic: per-node traffic shaping [Goyal and Vin, 15; Zhang and Knightly, 41], or per-node delay jitter control [Stoica and Zhang, 36; Verma et al., 37]. We take the latter approach in section 3.

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 resources has been considered for use in Virtual Private Networks (VPN) [Duffield et al., 12] and ATM Virtual Paths [Roberts et al., 33]. 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 in [Reisslein et al., 31] 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 [Andrews, 2; Li and Knightly, 22]. 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 [Boorstyn et al., 3] which presents 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 relevant to our work [Goyal and Vin, 15; Qiu and Knightly, 29; Zhang et al., 42,43].

The remainder of this paper is structured as follows. We state our assumptions about traffic arrivals and introduce the notion of effective envelopes in section 2. 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. We evaluate and compare the two designs through a computational study in section 5. Finally, 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 about traffic arrivals. Throughout this paper we will use a fluid-flow interpretation of traffic, and, moreover, we adopt the so-called regulated adversarial traffic model. A key construction for our analysis is the *effective envelope* associated with multiplexed flows, which we present formally in section 2.2 (following the earlier work of [Boorstyn et al., 3]). Intuitively, an effective envelope is a function that, with high certainty, provides an upper bound for arrivals from multiplexed flows in intervals of length $\tau$. We apply the concept of effective envelopes extensively in sections 3 and 4.

### 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 $A_j(t, t+\tau) \leqslant A_j^*(\tau)$ for all $t \geqslant 0$ and for all $\tau \geqslant 0$.[1] We assume that the limit $\rho_j := \lim_{\tau\to\infty} A_j^*(\tau)/\tau$ exists.

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

(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 requirements on delay. 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.

These or similar assumptions for regulated adversarial traffic are used in many recent works on statistical QoS [Boorstyn et al., 3; Doshi, 11; Elwalid et al., 13; Kesidis and Konstantopoulos, 16,17; Knightly, 19; LoPresti et al., 24; Oechslin, 25; Rajagopal et al., 30; Reisslein et al., 31,32] and are quite general. The main advantage of this model is that no assumptions are made on the distribution of flow arrivals, other than independence and that each flow satisfies a worst-case constraint. Thus, under assumptions (A1)–(A5), we consider arrival scenarios where each flow individually may exhibit its worst possible (‘adversarial’) behavior. Note that even if flows individually behave in a worst-case fashion, 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 [Boorstyn et al., 3]. Effective envelopes have been shown to be a useful tool for calculating the statistical multiplexing gain at a network node.[2]

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

[2] In [Boorstyn et al., 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.

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 $j \in \mathcal{C}_q$. Similarly, we use $\rho_q = \rho_j$ for all $j \in \mathcal{C}$.

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

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

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

Explicit expressions for effective envelopes can be obtained with large deviation results. In this paper, we will use a bound from [Boorstyn et al., 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 [Papoulis, 26])

$$\Pr\bigl[A_{\mathcal{C}_q}(0, \tau) \geqslant N_q x\bigr] \leqslant \mathrm{e}^{-N_q x s} M_{\mathcal{C}_q}(s, \tau), \tag{3}$$

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

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

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

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

$$M_{\mathcal{C}_q}(s, \tau) \leqslant \left[1 + \frac{\rho_q \tau}{A_q^*(\tau)}\bigl(\mathrm{e}^{s A_q^*(\tau)} - 1\bigr)\right]^{N_q}. \tag{4}$$

Using this bound it is possible to show [Boorstyn et al., 3] that

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

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

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

We will use the effective envelope given by equations (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 delay [Verma et al., 37]. Specifically, if traffic at a node experiences delay that 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 the same as the traffic arriving at the network edge.

We assume that 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 that arrives to a full buffer is dropped. The length of the buffer is provisioned so 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 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 $m_k$ ($1 < k \leqslant n$) holds traffic until the delay of the traffic has a delay equal to $d_{m_1} + d_{m_2} + \cdots + d_{m_{k-1}}$. The implementation of delay jitter control may require time-stamping of packets, and may incur additional buffer requirements [Verma et al., 37].

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 we will argue formally below, 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

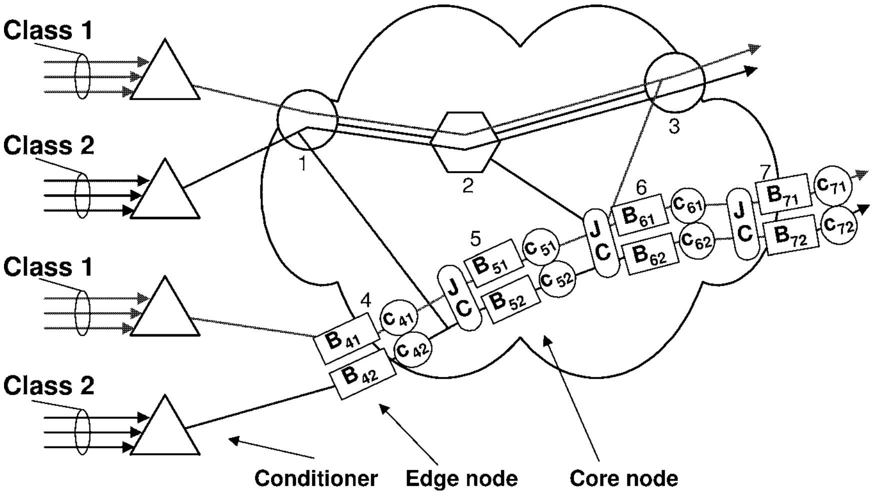


Figure 2. Network with class-level aggregation. At each edge node and each core node, there is one finite-length buffer for each class. Each buffer is served at a fixed rate, and arrivals to a full buffer are dropped. Core nodes have a delay jitter controller, labeled *JC* in the graph, which buffers 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.

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.

### 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 $A_{\mathcal{C}_{mq}} = \sum_{j\in\mathcal{C}_{mq}} A_j$ to denote the aggregate of network arrivals of class-$q$ that find their way to node $m$. Now define

$$\breve{A}_{\mathcal{C}_{mq}}(t-\tau,t) = \sum_{j\in\mathcal{C}_{mq}} A_j(t-d_{jm}) - A_j(t-\tau-d_{jm}), \tag{7}$$

where $d_{jm}$ denotes the total delay imposed by upstream jitter controllers on flow-$j$ packets until their arrival at node $m$. We refer to $\breve{A}_{\mathcal{C}_{mq}}$ as a *virtual arrival process* such that $\breve{A}_{\mathcal{C}_{mq}}(t-\tau,t)$ represents the total traffic that would have arrived at node $m$ in $[t-\tau,t)$ ignoring upstream losses. By the stationarity and independence assumptions of our traffic model, i.e. (A3) and (A4), the aggregated arrival process $\breve{A}_{\mathcal{C}_{mq}}$ is statistically equivalent to $A_{\mathcal{C}_{mq}}$, and assumptions (A1)–(A5) hold true for the set of flows that comprise $\breve{A}_{\mathcal{C}_{mq}}$, just as they do for $A_{\mathcal{C}_{mq}}$. Thus, an effective envelope $\mathcal{G}_{\mathcal{C}_{mq}}$ for the traffic from flows in $\mathcal{C}_{mq}$ can be obtained from equations (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

$$\sup_{\tau>0}\bigl(\mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon) - c_{mq}\tau\bigr) \leqslant c_{mq}d_{mq}, \tag{8}$$

and we set $B_{mq}$ to

$$B_{mq} = c_{mq}d_{mq}. \tag{9}$$

The rate $c_{mq}$ in equation (8) is set such that all class-$q$ traffic at node $m$ satisfies delay bound $d_{mq}$, as long as the arrivals comply to $\mathcal{G}_{\mathcal{C}_{mq}}$. Likewise, $B_{mq}$ is set such that traffic is dropped if the delay bound $d_{mq}$ is violated.

The following theorem states the statistical service guarantees that are attained with the above selection of $c_{mq}$ and $B_{mq}$. Our performance guarantees are stated using a particular characterization of loss. Specifically, using $\mathcal{L}_{mq}(t)$ to denote the total volume of class-$q$ traffic dropped at node $m$ in the busy period containing $t$ (defined to be zero if the corresponding buffer is empty at time $t$), we provide a time-invariant bound on the expected value of $\mathcal{L}_{mq}(t)$.

**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, where $c_{mq}$ and $B_{mq}$ are selected as in equations (8) and (9), respectively, then

1. Traffic which is not dropped meets its delay bound $d_{mq}$.

2. The probability that the class-$q$ buffer at node $m$ overflows at time $t \geqslant 0$ is bounded by $\varepsilon$ for all $t \geqslant 0$, and the expected value of $\mathcal{L}_{mq}(t)$ satisfies:

$$E\bigl[\mathcal{L}_{mq}(t)\bigr] \leqslant \varepsilon \cdot \sup_{\tau>0}\bigl(A^*_{\mathcal{C}_{mq}}(\tau) - \mathcal{G}_{mq}(\tau)\bigr), \quad \forall t \geqslant 0, \tag{10}$$

under the assumption that

$$\begin{aligned}\Pr\Bigl(&\sup_{\tau>0}\bigl(A_{\mathcal{C}_{mq}}(t-\tau,t) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau)\bigr) > 0\Bigr)\\ &\approx \inf_{\tau>0}\Pr\bigl(\bigl(A_{\mathcal{C}_{mq}}(t-\tau,t) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau)\bigr) > 0\bigr).\end{aligned} \tag{11}$$

The assumption in equation (11) is similar to an assumption made in [Boorstyn et al., 3], as well as in related work [Choe and Shroff, 7; Knightly, 18,19; Knightly and Shroff, 20; Kurose, 21]. A theoretical justification for this assumption is made in [Knightly and Shroff, 20], and the assumption has been supported by numerical examples [Boorstyn et al., 3; Choe and Shroff, 7; Knightly and Shroff, 20]. For the remainder of the paper, we refer to $E[\mathcal{L}_{mq}(t)]$ as the *loss metric for class-q traffic at node m*. Since $E[\mathcal{L}_{mq}(t)]$ gives a measure of the amount of traffic lost in a busy period at time $t$, we can normalize it by the average rate $\rho_q \cdot |\mathcal{C}_{mq}|$ to get a measure of loss rate at time $t$. This is the loss rate measure we use in section 5 in comparing the class-level and path-level designs.

*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 larger than $d_{mq}$. The second part of the theorem is a result of equation (8) and the definition of the effective envelope, as we now argue. Consider an arbitrary time $t$ with a traffic arrival to the buffer. By equations (8) and (9), an arrival will find a full buffer only if for some $\tau > 0$,

$$\tilde{A}_{\mathcal{C}_{mq}}(t-\tau,t) > \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon), \tag{12}$$

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

$$\Pr\bigl(\exists\tau > 0\colon\ \tilde{A}_{\mathcal{C}_{mq}}(t-\tau,t) > \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\bigr). \tag{13}$$

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

$$\begin{aligned}\tilde{A}_{\mathcal{C}_{mq}}(t-\tau,t) &= \breve{A}_{\mathcal{C}_{mq}}(t-\tau,t) - L_{\mathcal{C}_{mq}}(t-\tau,t) && (14)\\ &\leqslant \breve{A}_{\mathcal{C}_{mq}}(t-\tau,t). && (15)\end{aligned}$$

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

$$\begin{aligned}&\Pr\bigl(\exists\tau > 0\colon\ \tilde{A}_{\mathcal{C}_{mq}}(t-\tau,t) > \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\bigr)\\ &\quad\leqslant \Pr\bigl(\exists\tau > 0\colon\ \breve{A}_{\mathcal{C}_{mq}}(t-\tau,t) > \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\bigr) && (16)\\ &\quad= \Pr\Bigl(\sup_{\tau>0}\bigl(\breve{A}_{\mathcal{C}_{mq}}(t-\tau,t) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\bigr) > 0\Bigr) && (17)\\ &\quad\approx \inf_{\tau>0}\Pr\bigl(\breve{A}_{\mathcal{C}_{mq}}(t-\tau,t) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon) > 0\bigr) && (18)\\ &\quad\leqslant \sup_{\tau>0}\Pr\bigl(\breve{A}_{\mathcal{C}_{mq}}(t-\tau,t) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon) > 0\bigr) && (19)\\ &\quad\leqslant \varepsilon. && (20)\end{aligned}$$

Note that equation (16) follows from the fact that $\tilde{A}_{\mathcal{C}_{mq}}(t-\tau,t) \leqslant \breve{A}_{\mathcal{C}_{mq}}(t-\tau,t)$, and equation (17) is merely a different formulation of the right-hand side of equation (16).

Equation (18) uses the assumption in equation (11). Equation (19) asserts that the infimum is less than the supremum, and equation (20) 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 arrivals comply with $\mathcal{G}_{\mathcal{C}_{mq}}$, no traffic is dropped. Thus, the amount of traffic dropped up to time $t$ in the busy period containing $t$ is bounded by

$$\sup_{\tau>0}\bigl(\tilde{A}_{\mathcal{C}_{mq}}(t-\tau,t)-\mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\bigr).$$

So, we have

$$\begin{aligned}
&\text{(amount of traffic dropped in the busy period up to time } t\text{)}\\
&\quad\leqslant \sup_{\tau>0}\bigl(\breve{A}_{\mathcal{C}_{mq}}(t-\tau,t)-\mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\bigr) &(21)\\
&\quad\leqslant \sup_{\tau>0}\bigl(A^{*}_{\mathcal{C}_{mq}}(\tau)-\mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon)\bigr). &(22)
\end{aligned}$$

Thus, for all $t$, the total amount of traffic lost in the busy period containing $t$ is bounded as $\mathcal{L}_{mq}(t)\leqslant \sup_{\tau>0}(A^{*}_{\mathcal{C}_{mq}}(\tau)-\mathcal{G}_{\mathcal{C}_{mq}}(\tau;\varepsilon))$. The second part of the theorem now follows from equations (20) and (22). □

### *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.* Consider a class-$q$ flow $j$ that traverses a sequence nodes $m_1\to m_2\to\cdots\to m_L$, with $\mathcal{C}_{m_l q}$ the set of class-$q$ flows at each node $m_l$. Our analysis of the class-level aggregation scheme assumes that the class-$q$ arrivals at each node on the path are characterized by the virtual arrival process $\breve{A}_{\mathcal{C}_{m_l q}}$, which ignores upstream losses. In other words, we conservatively assume that losses at each node on a path occur regardless of losses upstream on the path. Defining $\mathcal{L}_{q,\text{path}}(t)=\sum_{l=1}^{L}\mathcal{L}_{m_l q}(t)$, we refer to $E[\mathcal{L}_{q,\text{path}}(t)]$ as the *loss metric for class-q traffic on the path*, and, with the assumptions from theorem 2, we have

$$E\bigl[\mathcal{L}_{q,\text{path}}(t)\bigr]\leqslant\sum_{l=1}^{L}\varepsilon\cdot\sup_{\tau>0}\bigl(A^{*}_{\mathcal{C}_{m_l q}}(\tau)-\mathcal{G}_{m_l q}(\tau)\bigr),\quad \forall t\geqslant 0. \tag{23}$$

*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 [Braden et al., 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 [Braden et al., 5] 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 analysis 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 [Parekh and Gallager, 27,28; Zhang, 40]. 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 [Shenker et al., 34].

Note that the delay jitter control at each node causes the schedulers for class-level aggregation to be non-work conserving. In principle, one could design the network so that, 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 [Andrews, 2; Ferrari and Verma, 14; Liebeherr and Yilmaz, 23; Stoica and Zhang, 36]. In such a scheme, the traffic arrivals would be assigned the same maximum delay at a node as if the delay jitter controllers were employed. Thus, if the arrivals are scheduled according to their deadlines, the traffic will be served in the same order as with delay jitter control, and the scheduler can send the traffic whenever the link is idle, resulting in lower end-to-end delays [Andrews, 2]. Unfortunately, the absence of delay jitter control (1) may increase the burstiness of traffic, and, hence, result in larger buffer requirements and (2) may introduce correlations into downstream traffic flows, invalidating assumption (A4). More research is needed to make a work conserving implementation of the class-level scheme a reality.

## 4. Networks with path-level aggregation (“Pipe model”)

One disadvantage of a network with class-level aggregation, as discussed above, 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”.

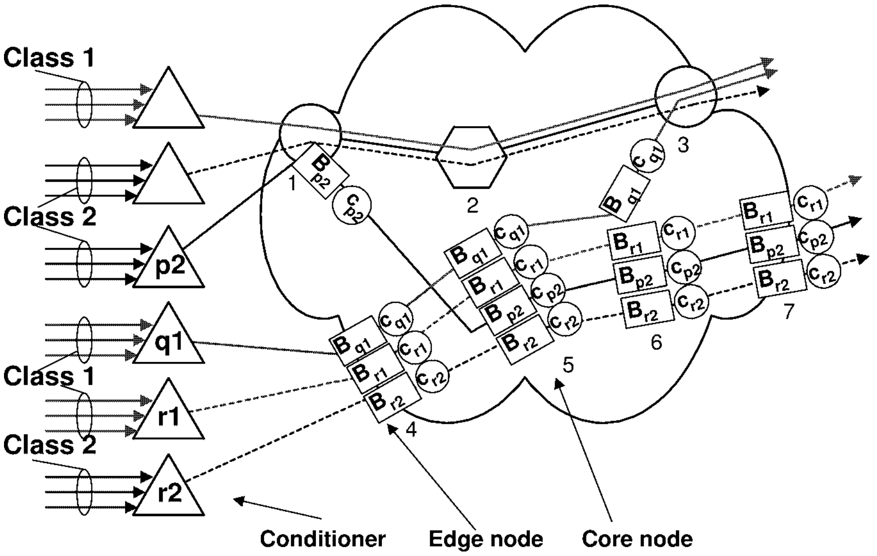


Figure 3. Network with path-level aggregation. An end-to-end path for a traffic class defines a path or "pipe". The figure depicts a total of six pipes, and depicts the buffers of four of the pipes, labeled $p2$, $q1$, $r1$, $r2$. For each pipe, the aggregate traffic is policed at the network entrance 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.

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 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 equations (5) and (6), where we set $N_q = |\mathcal{C}_{pq}|$ and $A^*_q$ is the deterministic envelope of $A_j(t)$, which is the same for each flow $j \in \mathcal{C}_{pq}$.

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 $\mathcal{G}_{\mathcal{C}_{pq}}(\cdot; \tau)$ 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

$$A_{\mathcal{C}_{pq}}(t, t+\tau) \leqslant \mathcal{G}_{\mathcal{C}_{pq}}(\tau) \quad \forall t \geqslant 0,\ \forall \tau \geqslant 0. \tag{24}$$

Traffic in excess of $\mathcal{G}_{\mathcal{C}_{pq}}$ is discarded by the conditioner.

For this architecture, theorem 3 below provides a bound for the traffic from a pipe which is dropped at the network entrance. Here, we use the same loss metric as in section 3. Let $\mathcal{L}_{pq}(t)$ denote the total volume of class-$q$ traffic dropped at the ingress node to path $p$ in the busy period containing $t$ (defined to be zero if the corresponding buffer is empty at time $t$). As before, we provide a time-invariant bound for $E[\mathcal{L}_{pq}(t)]$, which we call the *loss metric for path-$p$ class-$q$ traffic*. In section 4.2 we show how to provision the pipe so that no losses occur downstream from the ingress node, allowing us to interpret $E[\mathcal{L}_{pq}(t)]$ as the *loss metric for the pipe*.

**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 equation (24), then the probability that the class-$q$ buffer at the ingress node for the path $p$ overflows at time $t \geqslant 0$ is bounded by $\varepsilon$ for all $t \geqslant 0$, and the expected value of $\mathcal{L}_{pq}(t)$ satisfies:

$$E\big[\mathcal{L}_{pq}(t)\big] \leqslant \varepsilon \cdot \sup_{\tau>0}\big(A^*_{\mathcal{C}_{pq}}(\tau) - \mathcal{G}_{\mathcal{C}_{pq}}(\tau)\big), \quad \forall t \geqslant 0, \tag{25}$$

under the assumption that

$$\Pr\Big(\sup_{\tau>0}\big(A_{\mathcal{C}_{pq}}(t-\tau, t) - \mathcal{G}_{\mathcal{C}_{pq}}(\tau)\big) > 0\Big) \approx \inf_{\tau>0} \Pr\big(\big(A_{\mathcal{C}_{pq}}(t-\tau, t) - \mathcal{G}_{\mathcal{C}_{pq}}(\tau)\big) > 0\big). \tag{26}$$

*Proof.* The assumption in equation (26) is the same as in equation (11). 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

$$\begin{aligned}
&\Pr\big(\exists\tau > 0\colon\ A_{\mathcal{C}_{pq}}(t-\tau, t) > \mathcal{G}_{\mathcal{C}_{pq}}(\tau)\big) \\
&\quad = \Pr\Big(\sup_{\tau>0}\big(A_{\mathcal{C}_{pq}}(t-\tau, t) - \mathcal{G}_{\mathcal{C}_{pq}}(\tau)\big) > 0\Big) && (27)\\
&\quad \approx \inf_{\tau>0} \Pr\big(\big(A_{\mathcal{C}_{pq}}(t-\tau, t) - \mathcal{G}_{\mathcal{C}_{pq}}(\tau)\big) > 0\big) && (28)\\
&\quad \leqslant \sup_{\tau>0} \Pr\big(\big(A_{\mathcal{C}_{pq}}(t-\tau, t) - \mathcal{G}_{\mathcal{C}_{pq}}(\tau)\big) > 0\big) && (29)\\
&\quad \leqslant \varepsilon. && (30)
\end{aligned}$$

The amount of traffic dropped up to time $t$ in the busy period containing $t$ is bounded by

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

Thus, we have $\mathcal{L}_{pq}(t) \leqslant \sup_{\tau>0}(A^*_{\mathcal{C}_{mq}}(\tau) - \mathcal{G}_{\mathcal{C}_{mq}}(\tau; \varepsilon))$, and the theorem follows from equation (30). □

### *4.2. Scheduling at edge nodes and at core nodes*

With path-level aggregation, we allocate bandwidth and buffer space 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, and we use $B_{pq}$ to denote the reserved buffer space. We set $c_{pq}$ as the smallest number which satisfies

$$\sup_{\tau>0}\bigl(\mathcal{G}_{\mathcal{C}_{pq}}(\tau;\varepsilon)-c_{pq}\tau\bigr)\leqslant c_{pq}d_{pq}, \tag{31}$$

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

$$B_{pq}=c_{pq}d_{pq}. \tag{32}$$

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 equation (24), if $c_{pq}$ and $B_{pq}$ are allocated as given in equations (31) and (32) 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 $d_{pq}$.

*Proof.* The proof is done using Cruz's service curves [Cruz, 9]. Assume the flows in $\mathcal{C}_{pq}$ take the route $k=m_1\to m_2\to\cdots\to m_K=l$. Allocating a capacity of $c_{pq}$ to $\mathcal{C}_{pq}$ is equivalent to guaranteeing a service curve of

$$S_{m_i}(\tau)=c_{pq}\tau\quad\forall\tau\geqslant 0,\ \forall i,\ 1\leqslant i\leqslant K. \tag{33}$$

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

$$S_p(\tau)=S_{m_1}\star S_{m_2}\star\cdots\star S_{m_K}(\tau) \tag{34}$$

$$=c_{pq}\tau, \tag{35}$$

where $\star$ in equation (34) is the convolution operator defined as $G\star H(t)=\inf_\tau(G(\tau)+H(t-\tau))$. Equation (35) follows from repeated application of the convolution operator. According to theorem 4 in [Cruz, 9], 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

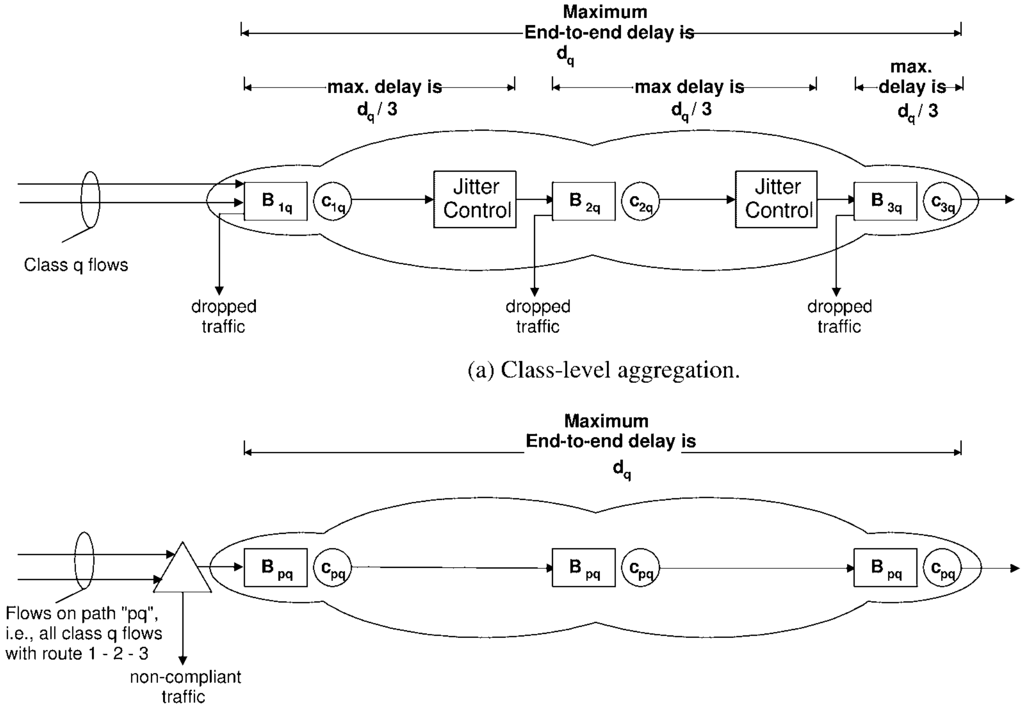
$$D_{\max}\leqslant\sup_{\tau>0}\inf_{\Delta\geqslant 0}\bigl\{\Delta\mid\mathcal{G}_{\mathcal{C}_{pq}}(\tau;\varepsilon)\leqslant c_{pq}(\tau+\Delta)\bigr\}. \tag{36}$$

Now, the theorem follows from equations (31) and (32). □

### *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 at every node, 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. By ensuring that packets experience the maximum node delay at each node (except the last node on the path), the class-level scheme may actually 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



(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.

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 $2d_q/3$. Thus, the end-to-end delay of all traffic sent over the three-node network has a lower bound of $2d_q/3$ 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 [Parekh and Gallager, 28] 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 equations (8) and (9). 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 equations (31) and (32). The QoS guarantee for class-$q$ is as stated in theorems 3 and 4.
- *Deterministic QoS.* For reference, we compare the class-level and path-level schemes above to a service with deterministic QoS that satisfies an end-to-end delay bound of $d_q$ for all aggregates of class-$q$ traffic. Let $\overline{L}$ be the maximum number of links in any path across the network, and let $\mathcal{C}_{mq}$ be the set of class-$q$ flows at node $m$. Select $c_{mq}$ as the smallest value such that $\sup_{\tau>0}(A^*_{\mathcal{C}_{mq}}(\tau) - c_{mq}\tau) \leqslant c_{mq}d_q/\overline{L}$, where $A^*_{\mathcal{C}_{mq}}(\tau) = |\mathcal{C}_{mq}| \cdot A^*_q(\tau)$. If all nodes in the network allocate bandwidth to the class-$q$ traffic in this way, then we can guarantee an end-to-end delay bound of $d_q$. For definiteness in our numerical examples, we assume $\overline{L} = 1$. (In this way we minimize the conservatism of deterministic provisioning. Interestingly, we will still find that the class-level and path-level provisioning schemes above can lead to significantly higher utilization of network resources.)

Table 1
Parameters of four traffic classes.

| Class | Type | | Burst size $\sigma_q$ (bits) | Mean rate $\rho_q$ (Mbps) | Peak rate $P_q$ (Mbps) | End-to-end delay bound $d_q$ (msec) |
|---|---|---|---|---|---|---|
| | burstiness | delay | | | | |
| 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 |

- *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.

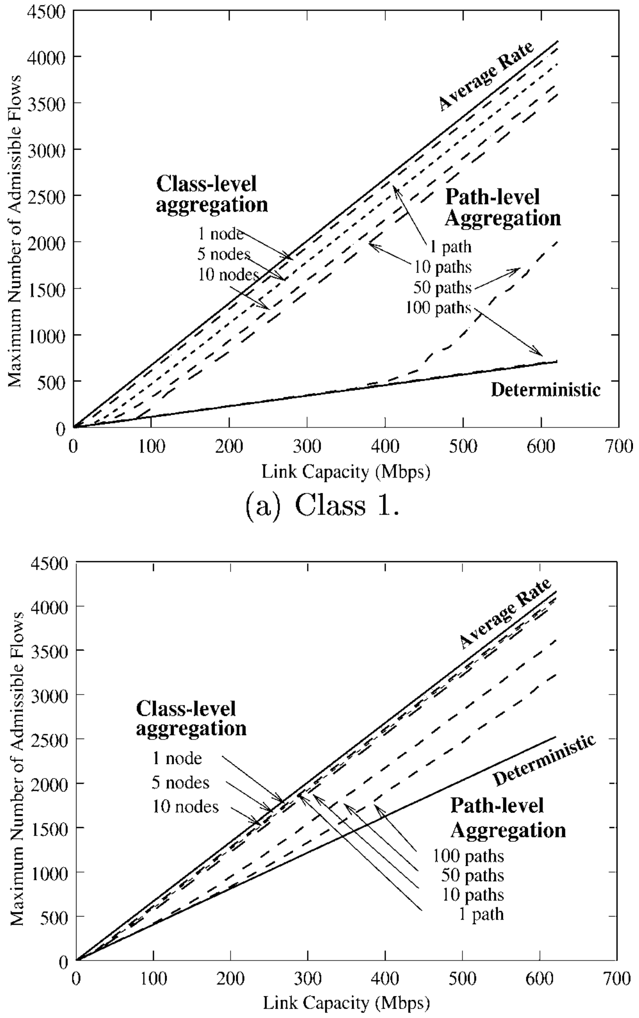
As an admission control condition, we require for all schemes above that the total allocated bandwidth on a link must not exceed the link capacity.

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 $(\sigma_q, \rho_q, P_q)$, 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 [Elwalid et al., 13; LoPresti et al., 24; Rajagopal et al., 30].

### 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 1–622 Mbps.

For class-level aggregation, first recall from our discussions in sections 3.3 and 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.

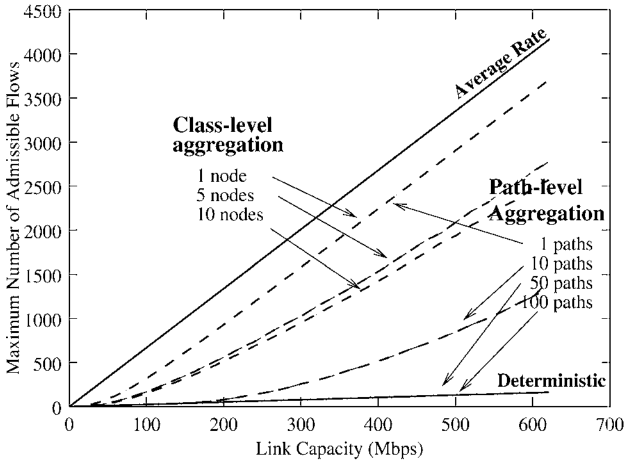


(a) Class 1.

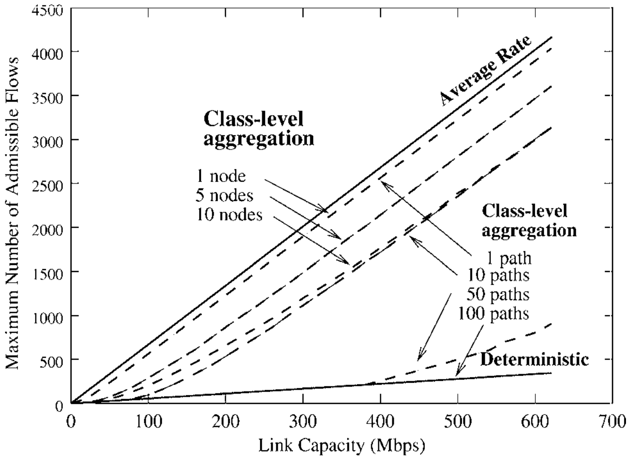
(b) Class 2.

Figure 5. Maximum number of admissible flows.

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



(c) Class 3.



(d) Class 4.

Figure 5. (Continued.)

to class-level aggregation with a 1 hop route.) Figures 5(a)–(d) show that the maximum 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.

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 aggregation |
|---|---|---|---|---|
| | 1 node | 2 nodes | 10 nodes | |
| 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 | $6.7 \cdot 10^{-7}$ | $1.3 \cdot 10^{-6}$ | $6.7 \cdot 10^{-6}$ | $6.7 \cdot 10^{-8}$ |

### 5.2. *Comparison of the traffic loss rate*

Next, we compare the expected loss rate of flows. Recall that the bounds for the alternative loss metric given in equations (23) and (25) apply to traffic aggregates (classes of traffic) and not to individual flows. Making QoS guarantees to aggregate flows yields a simple network core, since no per-flow information is required. On the other hand, individual flows may experience a service which is worse than the service guaranteed to the class as a whole. To obtain a measure of loss rate, we normalize the loss metric of a class by the long time average of the expected traffic in the class. We point out that this 'normalization' does not give a precise loss rate. Under the assumption that $\mathcal{C}_{mq}$ is the set of class-$q$ flows at all nodes, we obtain with equation (23) a (normalized) loss rate of

$$\text{Loss rate}^{\text{class}} < \frac{1}{\rho_q \cdot |\mathcal{C}_{mq}|} \cdot L \cdot \varepsilon \cdot \sup_{\tau>0}\bigl(A^*_{\mathcal{C}_{mq}}(\tau) - \mathcal{G}_{mq}(\tau)\bigr). \tag{37}$$

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

$$\text{Loss rate}^{\text{class}} < \frac{\sigma_q}{\rho_q}\varepsilon L. \tag{38}$$

The same consideration for the path-level scheme yields

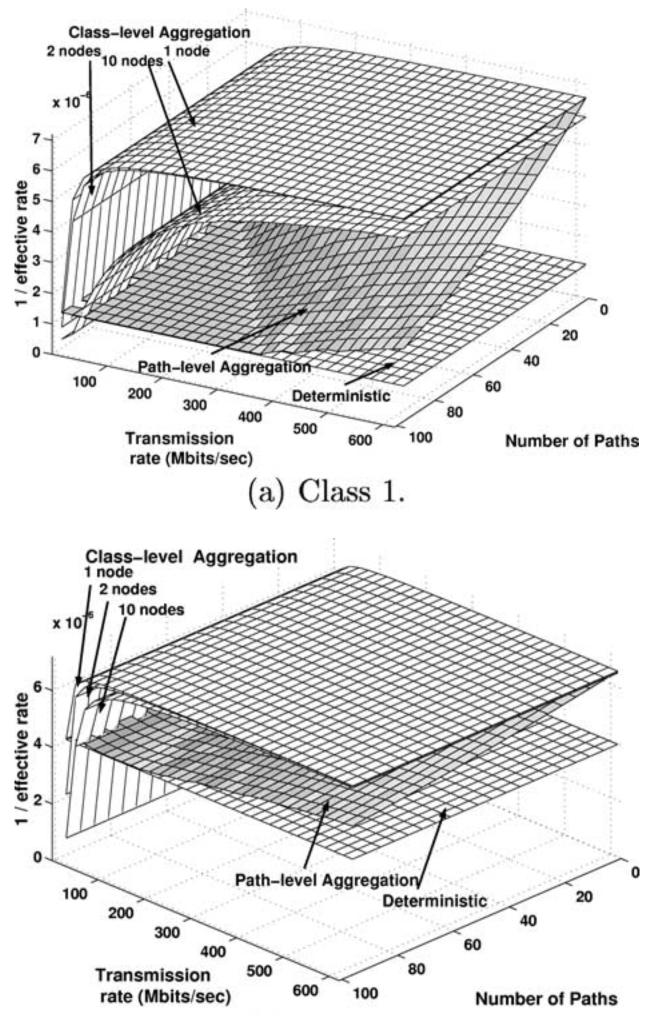
$$\text{Loss rate}^{\text{path}} < \frac{\sigma_q}{\rho_q}\varepsilon. \tag{39}$$

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 $\varepsilon$.

### 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 with respect to an increase of the number of paths.

We use as a 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



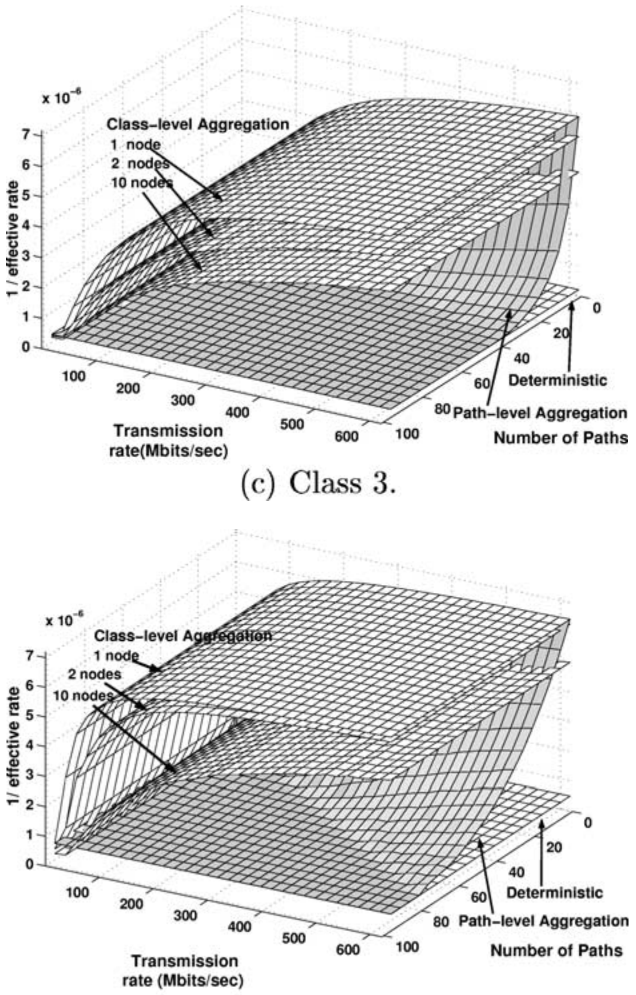
(a) Class 1.

(b) Class 2.

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

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 1–4, under deterministic provisioning, class-level aggregation, and path-level aggregation. For illustrative purposes, we plot values of $1/(\text{effective rate})$ as a function of the link capacity and as a function of the number of paths. 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



(c) Class 3.

(d) Class 4.

Figure 6. (Continued.)

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.

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 the 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 [Parekh and Gallager, 28] 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 traffic management specification version 4.0, ATM Forum (April 1996).

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

[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) 18 (December 2000) 2651–2664.

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

[5] R. Braden et al., Resource ReSerVation Protocol (RSVP) – version 1: functional specification, IETF RFC 2205 (September 1997).

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

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

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

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

[10] R.L. Cruz, A calculus for network delay, Part II: Network analysis, IEEE Transactions on Information Theory 37(1) (1991) 132–141.

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

[12] 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: *Proc. of ACM SIGCOMM'99*, Boston, MA, September 1999, pp. 95–108.

[13] 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) (1995) 1115–1127.

[14] 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) (1990) 368–379.

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

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

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

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

[19] E. Knightly, Enforceable quality of service guarantees for bursty traffic streams, in: *Proc. of IEEE INFOCOM'98*, San Francisco, March 1998, pp. 635–642.

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

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

[22] 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.

[23] J. Liebeherr and E. Yilmaz, Work-conserving vs. non-workconserving packet scheduling: An issue revisited, in: *Proc. of IEEE/IFIP 7th Internat. Workshop on Quality of Service (IWQoS '99)*, London, June 1999, pp. 248–256.

[24] 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: *Proc. of IEEE INFOCOM'97*, Kobe, Japan, April 1997, pp. 676–683.

[25] P. Oechslin, Worst case arrivals of leaky bucket constrained sources: The myth of the on-off source, in: *Proc. of IEEE/IFIP 5th Internat. Workshop on Quality of Service (IWQoS '97)*, New York, May 1997, pp. 67–76.

[26] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. (McGraw-Hill, New York, 1991).

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

[28] 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) (1994) 137–150.

[29] J. Qiu and E. Knightly, Inter-class resource sharing using statistical service envelopes, in: *Proc. of IEEE Infocom '99*, March 1999, pp. 36–42.

[30] S. Rajagopal, M. Reisslein and K.W. Ross, Packet multiplexers with adversarial regulated traffic, in: *Proc. of IEEE INFOCOM'98*, San Francisco, March 1998, pp. 347–355.

[31] M. Reisslein, K.W. Ross and S. Rajagopal, Guaranteeing statistical QoS to regulated traffic: The multiple node case, in: *Proc. of 37th IEEE Conf. on Decision and Control (CDC)*, Tampa, December 1998, pp. 531–531.

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

[33] 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, New York, 1996).

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

[35] D. Starobinski and M. Sidi, Stochastically bounded burstiness for communication networks, in: *Proc. of IEEE INFOCOM'99*, March 1999, pp. 36–42.

[36] I. Stoica and H. Zhang, Providing guaranteed services without per flow management, in: *Proc. of ACM SIGCOMM'99*, Boston, MA, September 1999, pp. 81–94.

[37] D. Verma, H. Zhang and D. Ferrari, Guaranteeing delay jitter bounds in packet switching networks, in: *Proc. of TRICOMM'91*, Chapel Hill, NC, April 1991, pp. 35–46.

[38] 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) (1996) 352–362.

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

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

[41] H. Zhang and E. Knightly, Providing end-to-end statistical performance guarantees with bounding interval dependent stochastic models, in: *Proc. of ACM SIGMETRICS'94*, Nashville, TN, May 1994, pp. 211–220.

[42] 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) (1997) 125–152.

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