A Delay Allocation Strategy for End-to-End Statistical Delay Assurances in FIFO Networks

Paper Id: 56

***Abstract*—Many Internet Service Providers deploy FIFO networks with shaping capability at the edge of the network. With this model, we examine the problem of providing statistical assurance on end-to-end delay.**

The end-to-end delay assured for a flow passing through a FIFO network is the sum of the maximal delays assured at each hop along its path. In order to evaluate a given FIFO network's capability in terms of delay assurances, we need to be able to compute delay targets for each hop *independent* of instantaneous traffic description.

Assuming that the flows are leaky-bucket shaped at the entry of the network, we provide a novel algorithm to compute the per-hop delay targets *without* exact traffic description, for a given probability of violation. We exploit this ability to evaluate a given FIFO network topology to compute the end-to-end delays it can provide. For the proposal to work we introduce some restrictions on the leaky-bucket profile of the admitted flow and path utilization. We demonstrate an admission control algorithm that enforces these restrictions without per-hop signaling and with low state requirement. The framework is implemented in NS-2 network simulator. We present results for a synthetic topology generated by GT-ITM and a real Internet topology, viz., the Telstra topology. We evaluate these networks for delay assurances using our proposed theory and verify the predictions via simulations.

I. Introduction

Quality of Service (QoS) in a computer network refers to assurances on performance metrics like bandwidth, delay and loss. A network with predictable QoS parameters is the necessary foundation for future applications. Many applications like voice and video require a bounded delay service from the underlying network. Consequently, the problem of providing end-to-end QoS assurances has seen a lot of interest in recent times [14]. Although there have been end-to-end QoS proposals based on specialized per-hop scheduling (e.g., GPS [23] [24], EDF [26] [27]), many Internet Service Providers (ISPs) continue to operate networks based on FIFO schedulers. Thus there is a need to develop a framework which can leverage existing infrastructure to provide bounded delay services and in this paper we look at one such framework.

The most popular QoS architectures proposed have been Integrated Services (Intserv [4]) and Differentiated Services (Diffserv [2]). By envisaging per-flow scheduling and admission control, Intserv proposes to achieve the aim of guaranteed QoS. The per-flow state requirement hinders the scalability of the architecture. Diffserv trades off the ability to provide guaranteed QoS by employing aggregate scheduling and hence achieves scalability.

Although guaranteed service is desirable, scalable architectures are the ones that are deployable. In this regard, it becomes important to quantify the trade-off dictated by scalability. That is, while adopting aggregate scheduling, it is imperative that we understand the *effective service* seen by constituents of the of the aggregate. In this regard, a framework to analyse flows in a network of multiplexers becomes important. Cruz [9] [10] provided such a framework with a deterministic network calculus. The utility and power of network calculus has been evident in the insight it has provided on such questions as network stability [1] and utilization bounds for finite delays in FIFO networks [7], [15]. These insights have led to new scheduling disciplines and QoS architectures [3] [8] [34] to achieve bounded delay services. However, deterministic guarantees lead to low network utilization. Probabilistic assurances on QoS parameters have been shown to achieve remarkably higher utilization [5].

For probabilistic QoS assurances, the input is characterized by statistical envelopes (e.g., a bounding distribution [19], or stochastic burstiness bounds [29] [30] [35], or a bound on the variance of the rate [18] etc.). Statistical analysis of flows inside a network is highly complex due to the correlation introduced by multiplexing. QoS proposals either include mechanisms to avoid correlation (e.g., per-hop jitter controllers [12] [20] [33], buffer-less multiplexing [25] etc.) or resort to approximate analysis. If network-wide upgrade is not a viable option, approximate methods which treat the network as a black-box, prove to be valuable tools (e.g., analysis of Diffserv EF networks [31] [11]). An underlying principle for such methods is to bound the worst-case effects of each hop of multiplexing. While allowing the network to remain simple, these methods can help provide assurances on QoS metrics with low probability of violation. These techniques have been used to compute delay violation probabilities at a given multiplexer if the exact flow descriptions are known.

Given that a maximal delay, d_i, is assured at each hop i of a path, the end-to-end delay assured is the sum of these per-hop delays. An assurance on per-hop delays implies restriction on the number of flows that can be admitted into the system. The objective then is to allot these delay targets such that a useful end-to-end delay target is met *and* network utilization is maximized.

Although there exist techniques to compute $Pr\{D_i > d_i\}$ (where D_i is the delay at the multiplexer i), given the flow descriptions, they do not tell us what value to choose for d_i. This is important, since without knowing the delay target, we cannot give an a priori assurance on end-to-end delay.

Further, current techniques to compute $Pr\{D > d\}$ rely on *exact* traffic descriptions, i.e., we need to know the number of flows and their leaky-bucket parameters. This again means we cannot make a statement about a network's ability to provide a particular delay assurance, without knowing traffic details.

This leads us to the following questions, which we shall examine in this paper:

- Given a network with FIFO schedulers and a desired probability of violation on end-to-end assurances, can we compute the end-to-end delays that can be assured without needing flow descriptions?
- For a given path, what are the feasible values for an end-to-end delay budget?
- How do we apportion this delay budget among the schedulers along the path?

One strategy that has been used before [26] [31] would be to assume a given d and divide it equally among the hops. However, in a FIFO network [3], the burstiness of a flow increases as it is multiplexed across multiple hops. In the presence of higher burstiness, the probability of delay violation increases. This clearly points to the fact that not all nodes can assure the same delay at a given probability. Besides, the probability of violation also depends on topological characteristics. For example, Figure 1 shows two paths P1 and P2, with P2 passing through a bottleneck link. For P1, we would expect nodes D and E to be able to assure a given delay at a lower probability as compared to A; similarly for P2, one would expect delay violation to be highly likely at C. This indicates that we need to allocate higher delay targets for the nodes that are faced with burstier flows, so that the probability of violation remains the same. Thus, a good delay allocation scheme can

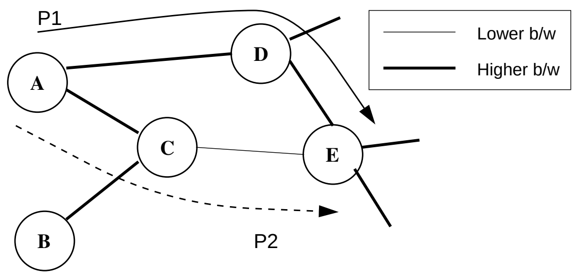

Fig. 1. Per-hop delays vary with nature of the path

lower the probabilities of violation of a delay assurance, for a given delay budget.

The contributions of this paper are as follows.

- We devise a novel technique to derive the per-hop delay targets without exact flow description.
- As a result, we are able to evaluate a FIFO network to compute the delays that it can assure independent of the traffic profile.
- For the proposal to work, we impose some restrictions on the nature of flows admitted, viz., an upper bound on (σ, ρ) of admissible flows, a bound on the busy period at the entry of the network (ingress) and pre-determined per-path utilization targets. In order to realize this framework we introduce an admission control algorithm that does not require per-hop signaling and has low state requirements.

Using simulations with real and synthetic network topologies we demonstrate that the delay allocation scheme in coordination with the admission control algorithm, can provide *a priori* delay assurances in a FIFO network. The rest of the paper is organized as follows. The network model and assumptions used are detailed in §II. In §III we discuss the delay allocation strategy and present the admission control algorithm. The results of applying it to network topologies are treated in §IV. We note the conclusions and future directions in §V.

II. Notation and Assumptions

We consider a network with FIFO schedulers with capabilities to shape aggregates at the entry of the network (the edge or the ingress). A "flow" refers to a traffic aggregate serviced by this network. Thus flows contain a stream of connections. A flow i is described by its leaky bucket parameters (ρ_i, σ_i). All the flows admitted into the network are assumed to be shaped in such a manner. Every node is assumed to have multiple input and output links. Each node is assumed to consist of a set of *output queues*. A packet enters a node and is switched to the appropriate output queue. We shall refer to the output queue as the multiplexer. Thus each multiplexer is associated with a link.

d_i, d_i^*	Delay target for mux i
σ_i	Burst parameter for flow i
ρ_i	Rate parameter for flow i
$\sigma_i^{(j)}$, $\rho_i^{(j)}$	Burst, Rate for flow i at mux j
σ^{max}	Max Burst param. for an admissible flow
β_i	Max Busy period at mux i
β^{max}	Max allowable busy period at the ingress
D	End-to-end delay along a path
D_i	Delay Variable at mux i
D_b	End-to-end delay budget
$D_b^{(i)}$	End-to-end delay budget
P_{vio}	Probability of violation of delay target
Γ_i	Capacity of Path i
u_i	Utilization target for path i

TABLE I

SYMBOLS AND THEIR MEANING

A flow that is admitted into the network first contacts the admission control entity. The admission control module uses information about the path on which the flow is being admitted, its leaky-bucket characteristics and pre-computed delay targets to decide on whether to admit the flow. Table I provides a list of symbols employed in the paper and their meaning.

III. Delay Allocation

In the following paragraphs we present a technique to obtain the delay targets for a path in a given topology of FIFO nodes. In §III-A we motivate the method detailed in §III-B.

A. Overview

Consider a flow i characterized by $(\sigma_i^{(0)}, \rho_i^{(0)})$ served by a multiplexer with capacity C. When it exits the multiplexer its burstiness increases (in the worst-case) to

$$\begin{aligned} \sigma_i^{(1)} &= \sigma_i^{(0)} + \rho_i^{(0)}\beta \\ \beta &= \frac{\sum_j \sigma_j^{(0)}}{C} \end{aligned} \quad (1)$$

Here β is defined as the maximum busy period at the multiplexer. This points toward a method to obtain the characterization of a flow after it has passed through multiple hops. If we compute the flows' characteristics at a multiplexer, by accounting for the worst-case possibilities of each previous hop, they can be considered independent [6] [19]. Given such a characterization for each flow at a multiplexer there are techniques to compute the

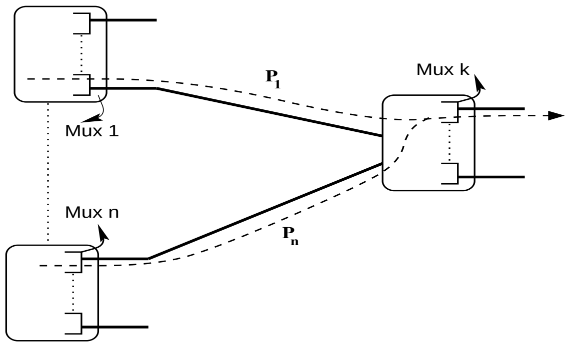

Fig. 2. Illustration of multiple paths feeding into a multiplexer; $\mathbf{P_i}$ indicates the set of paths traversing multiplexer i

d_i which can be assured with a given probability. For example, the bound in [16], adapted for leaky-bucket flows (σ_i, ρ_i) (please refer to Appendix §A) takes the following form for a multiplexer whose busy period is β:

$$Pr\{D_i > d_i\} \approx \max_{0 \le \tau \le \beta} \frac{1}{\sqrt{2\pi}} exp(-\frac{(C(\tau + d_i) - \tau \sum_i \rho_i)^2}{2\tau \sum_i \rho_i \sigma_i}) \quad (2)$$

However, we need to make this computation independent of the individual (σ, ρ) values. In order to do this, we obtain bounds on the quantities of interest, viz., $\sum_i \rho_i$, $\sum_i \rho_i \sigma_i$ and β.

Observe that the maximum busy period increases as the flow passes multiple hops. Thus a multiplexer upstream can assure a lower delay at the same probability of violation as compared to one downstream. Thus, instead of allocating equal *delay targets*, if we allocate the same maximum *probability of violation* (P_{vio}) to each hop we can ensure a low end-to-end probability of violation. If we account for worst-case possibilities at each hop, the per-hop delay probabilities can be approximated to be independent. Then let d_i^* be the delay that can be assured at probability of violation P_{vio} at multiplexer i. Setting $Pr\{D_i > d_i^*\} = P_{vio}$, we have for n hops:

$$Pr\{D > \sum_i d_i^*\} \le 1 - (1 - P_{vio})^n \quad (3)$$

Observe that the d_i^* values obtained are the minimal delays that can be assured at the respective hops to obtain an probability of violation less than P_{vio} at each hop. Thus any end-to-end delay budget has to be greater than $\sum_i d_i^*$. In the succeeding sections, we first present the means to obtain d_i^*, then discuss dividing a given delay budget d and finally present an admission control algorithm that will help achieve these assurances.

B. Computing the delay targets

In order to obtain bounds on the flow characteristics at a multiplexer, we first examine a typical scenario. Figure 2 depicts the multiplexers in a set of interconnected nodes. Consider the multiplexer M_k, fed by $M_1 \dots M_n$ $(k > n)$. Our strategy will be to express the properties of a flow at M_k in terms of the flows entering $M_1 \dots M_n$. The intuition behind the method is that if we bound the flow's properties at the entry of the network, we can obtain its description at any node inside the network by a recursive computation. First, we express the busy period β_k at M_k in terms of those at $M_1 \dots M_n$ and then relate it to the burstiness $\sigma_i^{(k)}$ of a flow at M_k. Second, notice that there are multiple "paths" traversing a multiplexer. By setting a utilization target for each path and computing a path capacity, we show that the term $\sum_i \rho_i$ can be bounded. We first begin with a definition of a path.

Definition III.1: A path is defined as a sequence of multiplexers $\{M_j\}$ such that M_j is connected to M_{j+1} and for any $i \neq j$, $M_j \neq M_i$. That is the paths considered for analysis are loop-free.

Since the analysis relies on recursively evaluating the flow descriptions, the paths should be defined so that *feedback* does not happen. To see this, observe Figure 3. To compute the characteristics of the output of M_2 we need the output of M_3 and vice versa. But if this has to happen then a packet arriving along P_1 (dotted curve) at node A should initially be forwarded to M_3; the same packet should be forwarded to M_4 if it arrives from node B. Disallowing this is not a restriction on the topology, instead it is an assumption about the routing protocol. Under normal operation of a shortest-path routing protocol such paths cannot be created. With the following assumption, we exclude these cases.

Assumption III.1: The paths defined on a topology are such that they do not lead to feedback.

We can now obtain the busy period bound as follows.

Proposition III.1: Consider a multiplexer M_0, fed by $M_1 \dots M_k$. Denote the busy period bound of M_i by β_i and capacity of M_i by C_i. Then β_0 satisfies the following:

$$\beta_0 \leq \frac{\sum_{i=1}^{k} \beta_i C_i}{C_0} \tag{4}$$

Proof: Denote the set of flows coming from M_i, $1 \leq i \leq k$, by F_i. Then,

$$\begin{aligned} \beta_0 &= \frac{\sum_j \sigma_j}{C_0} \\ &= \frac{\sum_{i=1}^{k} \sum_{j \in F_i} \sigma_j}{C_0} \\ &\leq \frac{\sum_{i=1}^{k} C_i \beta_i}{C_0} \end{aligned}$$

■

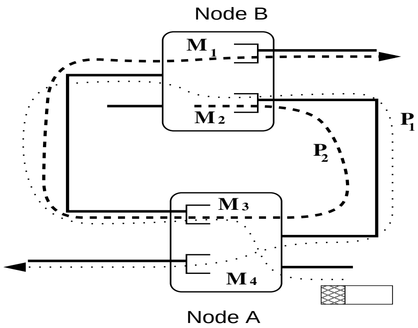

Fig. 3. Paths creating feedback are not allowed

We note that the summation in Proposition III.1 depends on the number of multiplexers feeding into M_0, which is a property of the topology and not on the number of flows. Further, consider the relation in Equation(1). It directly leads us to the following proposition for evaluating the maximum burstiness parameter among the flows in a multiplexer.

Proposition III.2: Consider a multiplexer M_0, fed by $M_1 \dots M_k$. Let ρ^{max} denote the upper bound on the rate parameter of admitted flows. Let $\sigma_{max}^{(i)}$ denote the maximum burstiness parameter among all flows at M_i. Then,

$$\sigma_{max}^{(0)} \leq \max_{1 \leq i \leq k} \{\sigma_{max}^{(i)} + \beta_i \rho^{max}\} \tag{5}$$

Equations(4,5) can be evaluated recursively, if the values for β and $\max_i(\sigma_i, \rho_i)$ are known at the first multiplexer of each path (entry of the network or the ingress). We thus introduce the following assumption to evaluate Equations(4,5) independent of the traffic profile.

Assumption III.2: The maximum busy period at the ingress is fixed at β^{max}. The maximum leaky-bucket burst and rate parameters of a flow allowed into the network are fixed at σ^{max} and ρ^{max} respectively.

In the succeeding section we show that the assumption is easily enforced by an admission control algorithm. In order to obtain bounds on $\sum_i \rho_i$ and $\sum_i \rho_i \sigma_i$, we introduce the notion of *path capacity* and *per-path utilization* target.

Definition III.2: Consider a path P_i defined by the sequence of multiplexers $\{M_j\}$. Let C_j be the capacity of M_j. Let n_j be the number of paths passing through M_j. Then, the capacity of the path P_i, is defined as: $\Gamma_i = \min_j \frac{C_j}{n_j}$. The utilization target for the path, u_i is such that for the flows (σ_i, ρ_i) admitted on the path P_i, $\frac{\sum_i \rho_i}{\Gamma_i} \leq u_i$.

We then have the following proposition.

Proposition III.3: Consider a multiplexer M_0. Let $P_1 \dots P_k$ denote the paths passing through M_0. Let F_i denote the set of flows traversing P_i. Let the capacity and utilization target for P_i be Γ_i and u_i respectively. Then *we have the following result.*

$$\sum_j \rho_j \leq \sum_{i=1}^{k} u_i \Gamma_i = \mu_1 \quad (6)$$

$$\sum_j \rho_j \sigma_j \leq \sigma_{max}^{(0)} (\sum_i u_i \Gamma_i) = \mu_2 \quad (7)$$

$$\sum_j \sigma_j^2 \leq C_0 \beta_0 (C_0 \beta_0 - 2\sigma_{max}^{(0)}) = \mu_3 \quad (8)$$

Proof:

$$\begin{aligned}
\sum_j \rho_j &= \sum_{i=1}^{k} \sum_{j \in F_i} \rho_j \\
&\leq \sum_{i=1}^{k} u_i \Gamma_i \\
\sum_j \sigma_j \rho_j &\leq \sigma_{max}^{(0)} \sum_j \rho_j \\
&\leq \sigma_{max}^{(0)} (\sum_{i=1}^{k} u_i \Gamma_i) \\
\sum_j \sigma_j^2 &= (\sum_j \sigma_j)^2 - 2 \sum_{k \neq l} \sigma_k \sigma_l \\
&\leq C_0^2 \left(\frac{\sum_j \sigma_j}{C_0} \right)^2 - 2\sigma_{max}^{(0)} C_0 \left(\frac{\sum_j \sigma_j}{C_0} \right) \\
&\leq C_0^2 \beta_0^2 - 2\sigma_{max}^{(0)} C_0 \beta_0 \\
&= C_0 \beta_0 (C_0 \beta_0 - 2\sigma_{max}^{(0)})
\end{aligned}$$

■

In Propositions III.1, III.2 and III.3 we have derived bounds that are dependent on a busy period bound at the ingress, an upper limit on the burstiness of flows admitted, a utilization target for the paths being analyzed and the capacity of the paths. Clearly, none of these quantities depend on exact traffic descriptions. Given a network topology and paths specified in accordance with Assumption III.1, we can compute these bounds.

The objective of obtaining these bounds is to evaluate the delay target at each multiplexer along a path. We apply these bounds to results obtained in [16] and [31] to obtain Equation 9 and 10 respectively (please refer to the Appendix) for a multiplexer M_i with paths $P_1 \dots P_k$.

$Pr\{D_i > d_i^*\} \leq$

$$\max_{0 \leq \tau \leq \beta_i} \frac{1}{\sqrt{2\pi}} exp \left(- \frac{(C(\tau + d_i) - \tau \mu_1)^2}{2\tau \mu_2} \right) \quad (9)$$

$Pr\{D_i > d_i^*\} \leq$

$$\sum_{k=0}^{K} exp \left(- \frac{2[(C(\tau_k + d_i) - \tau_{k+1}\mu_1)^+]^2}{(\mu_1 \tau_{k+1} + \sqrt{\mu_3})^2 \bigwedge (4\mu_3)} \right) \quad (10)$$

We can use either of these expressions to compute d_i^*. Having demonstrated the tools to obtain d_i^* for a given topology, we now examine the delay allocation algorithm.

C. An Algorithm for Delay Allocation

In the following paragraphs we consider the question of dividing a delay budget $D_b^{(i)}$ among the multiplexers along a path P_i.

The burstiness of incident flows increases with increase in the number of hops of multiplexing. Consequently, the probability of violating a given delay target is higher for a multiplexer downstream along a path. Thus if equal delay targets are allocated to each hop, the end-to-end probability of violation for the delay increases. Instead, if we obtain the delay targets by assigning a low constant probability of violation, P_{vio}, at each hop, we obtain an end-to-end assurance that is met with low probability of violation. In this case, the delay targets are higher in multiplexers downstream. Also, since each hop has to satisfy a violation probability bound of P_{vio}, we require that $D_b^{(i)} \geq \sum_i d_i^*$.

We then propose an allocation scheme as follows. First, set a desired limit on probability of violation for a multiplexer, say, P_{vio}. The end-to-end probability of violation V_e for a path with n hops would be:

$$\begin{aligned}
V_e &\approx 1 - (1 - P_{vio})^n \\
&\approx nP_{vio} \quad (11)
\end{aligned}$$

Second, compute d_i^* for each multiplexer M_i by using Equation (9) or (10), obtained in §III-B. Allocate d_i^* to M_i. Distribute $(D_b^{(i)} - \sum_i d_i^*)$ equally among those multiplexers that have only path P_i passing through them. To appreciate the reason behind this condition, note that there might be multiple paths passing through a multiplexer. Increasing the delay target for this multiplexer might affect several paths. Algorithm 1 summarizes this procedure.

D. Admission Control Algorithm

Employing Algorithm 1 one can assign delay targets to each multiplexer in a network so that a desired upper limit on end-to-end probability of violation is assured. In order to ensure that the flows admitted into the network actually obtain the assurances described in §III-B and §III-C we need an admission control algorithm that enforces all the requirements. Briefly, the tasks of the admission control module would be to enforce the limits on busy period and burstiness parameter, to prevent violation of the utilization target and to ensure per-hop delay targets are met. Algorithm 2 achieves these goals. The admission control

Algorithm 1 Delay Allocation Algorithm

Input: Probability of violation P_{vio}
Input: Delay Budget for Path i, $D_b^{(i)}$
Input: Paths $P_1 \dots P_n$

Compute d_i^* for each multiplexer
$d_i \leftarrow d_i^*$
for all Paths P_i **do**
  if $D_b^{(i)} < \sum_{i:M_i \in P_i} d_i^*$ **then**
    Return Error
  end if
  $S \leftarrow \{$ M_i: P_i is the only path through M_i $\}$
  $|S| \leftarrow$ Number of elements in S
  for all $M_i \in S$ **do**
    $d_i \leftarrow d_i^* + \frac{(D_b^{(i)} - \sum_j d_j^*)}{|S|}$
  end for
end for
Return vector of d_i values

module is assumed to be a central entity with information about the whole network. In the following section we show that the state requirements are modest and that we do not need the algorithm to contact individual hops for an admission control decision.

E. State Requirements

The admission control algorithm (Algorithm 2) requires that we maintain certain information about the admitted flows. There is both per-multiplexer information and network information that is stored. However, as we shall see, most of the data structures are *constant* in size with increase in number of flows admitted.

The per-multiplexer information includes four quantities, viz., $S_\rho^{(i)}$, $S_\sigma^{(i)}$, $S_{\rho\sigma}^{(i)}$, d_i. Except for d_i, these quantities change as flows are added and deleted. Note that whatever the number of flows, there are only four quantities stored per-multiplexer. Since these values are computed from the flow's properties and topology information, there is *no need* to contact individual hops.

Network information includes the path vector P_i, path capacity Γ_i, the current path utilization, v_i, the maximum admissible leaky-bucket profile $(\sigma^{max}, \rho^{max})$ and maximum permissible ingress busy period β^{max}.

In order to remove a flow from the network, we will need the profile of that flow. This requires that we maintain a vector of admitted flows. This is the only data structure whose size will grow with the number of admitted flows.

Algorithm 2 Admission Control Algorithm

Input: σ^{max}, the maximum allowable burst parameter for a flow
Input: β^{max}, maximum busy period allowable at ingress
Input: (ρ, σ), leaky-bucket parameters of the flow being admitted
Input: Path P_i, Util. target u_i, Path Capacity Γ_i, Current path util v_i
Input: d_i, Delay Target at multiplexer i
Input: P_{vio}, Per-hop probability of violation of delay target
Input: $S_\rho^{(i)}, S_\sigma^{(i)}, S_{\rho\sigma}^{(i)}$, Current $\sum_j \rho_j$, $\sum_j \sigma_j$, $\sum_j \sigma_j \rho_j$ for multiplexer i
Input: C_{ing}, Capacity of ingress on path P_i
if $(\sigma, \rho) > (\sigma^{max}, \rho^{max})$ OR $\rho/\Gamma_i + v_i > u_i$ **then**
  Reject flow
end if
if $\sigma + S_\sigma^{(ing)} > C_{ing}\beta^{max}$ **then**
  Reject flow
end if
for all Multiplexers M_i in P_i **do**
  Compute new $S_\rho^{(i)}, S_\sigma^{(i)}, S_{\rho\sigma}^{(i)}$
  if $P\{D_i > d_i\} > P_{vio}$ **then**
    Reject flow
  else
    continue
  end if
end for
for all Multiplexers M_i in P_i **do**
  Save new $S_\rho^{(i)}, S_\sigma^{(i)}, S_{\rho\sigma}^{(i)}$
end for
$v_i \leftarrow v_i + \rho/\Gamma_i$
Accept Flow

F. Discussion

In this section we examine some of the assumptions needed to ensure that the algorithms work as expected. The delay targets computed for each path are dependent on the fact that traffic is routed along that path and the route does not change. If the path changes, the assurances for the flow are no longer valid and need to be recomputed. The model also does not take into account additional traffic created due to errors in the routing protocols (e.g., stale and looping packets).

One advantage of using a bound on the leaky-bucket profile is that we can express the parameters of interest independent of flow characteristics. However, if there is a lot of variation in the typical flow size (in terms of (ρ, σ)) the utilization target might not be achieved. A possible solution might be to view bigger flows as being constituted by a bunch of smaller sized flows. These issues are to be examined further as part of future work.

IV. RESULTS

The objective of this section is to evaluate a given topology for end-to-end delay assurances. Specifically, we apply the analytical results of §III to compute the delay that can be assured *a priori* for a given probability of violation, P_{vio}. We compare the prediction with simulation results and examine the effect of topological character on the results. The simulation setup is described in §IV-A. In §IV-B we examine numerical results to evaluate the delay allocation algorithm and some strategies for parameter setting. In §IV-C the algorithm is evaluated with an NS-2 simulation setup.

A. Simulation Setup

For all the experiments, two topologies were employed, viz., a synthetic *Transit-Stub* topology generated by GT-ITM [32] and a real (Telstra) ISP topology as discovered by *Rocketfuel* [28]. Table II provides the details for these topologies. A pruned version of the Telstra topology was used; due to space constraints only a schematic with nodes representing major cities is shown in Figure 4. A set of nodes were chosen as ingress-egress pairs. Traffic is destined to the egress from the ingress. The number of such pairs define the number of paths that were evaluated in one run of the simulation and that is indicated under the column *Paths* in Table II. In the experiments that follow, we set $(\rho^{max}, \sigma^{max}) = (0.2Mbps, 40kbits)$. We discuss some strategies to set the parameter values in §IV-B. As indicated in Table II, all links were assigned an identical bandwidth ($10Mbps$ in the case of GT-ITM, $20Mbps$ in the case of Telstra) and a propagation delay of $10ms$. The packet simulations were carried out in the Network Simulator NS-2 using trace-driven traffic sources[1].

We first present results for numerical experiments in §IV-B and then present the validation of the model through simulation in §IV-C.

B. Numerical Experiments

In the following sections, we employ Propositions III.1, III.2, III.3 and Equation 9 to arrive at end-to-end delay values.

[1]http://www-tkn.ee.tu-berlin.de/research/trace/ltvt.html

Topology	Nodes	Links	Paths	Path Len	B/W
GT-ITM	100	193	20	7-13	10M
Rocketfuel (Telstra)	64	116	15	6-12	20M

TABLE II
TOPOLOGIES USED FOR SIMULATIONS

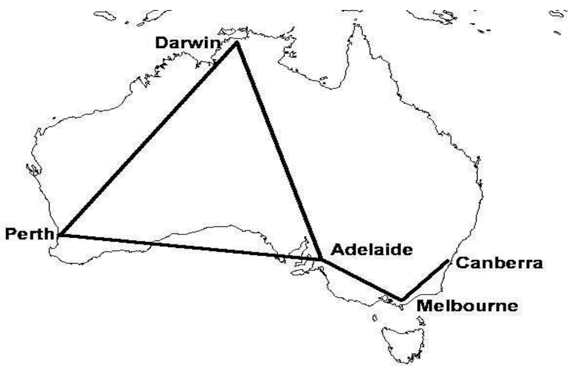

Fig. 4. Telstra Network Topology (only a few nodes are shown here)

1) Delay Allocation: While presenting the analysis in §III, we observed that an equal allocation of delay among the hops of a path is not a suitable choice. In order to verify the assertion, we choose a path in the GT-ITM topology. We set a delay budget of $D_b = \sum_i d_i^*$. Figure 5 shows the result of distributing D_b equally among the hops of a path as against setting d_i^* as the delay target for hop i. Clearly, the probability of violation is higher if we choose equal allocation for the same delay budget. The figure shows the result for a long path (10 hops) and a short path (6 hops).

2) Setting Parameter Values: For an analysis independent of traffic character, we assumed that the maximum admissible leaky bucket profile, maximum allowable busy period at the ingress and utilization targets are known. The

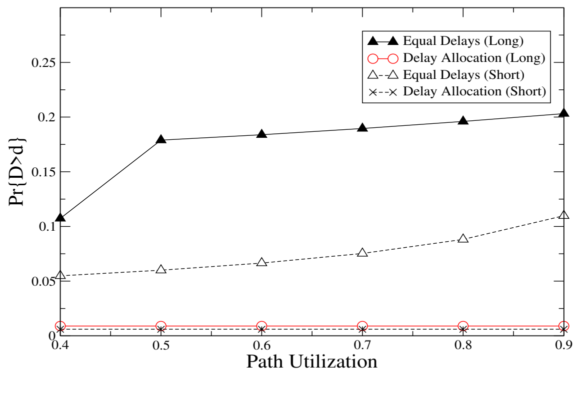

Fig. 5. Equal delay budget allocation leads to high violation probability. Probability of violation increases with path length (hops).

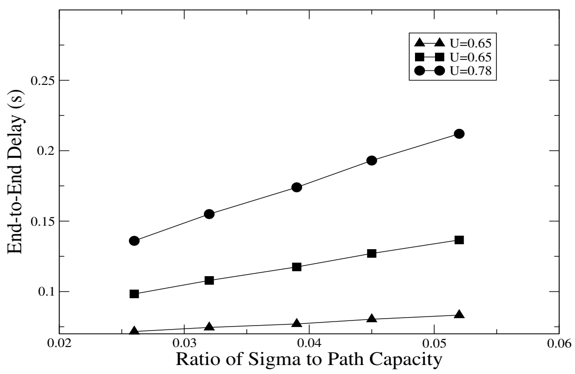

Fig. 6. Predicted Delays for path from Canberra to Adelaide for increasing σ^{max} as a function of path utilization

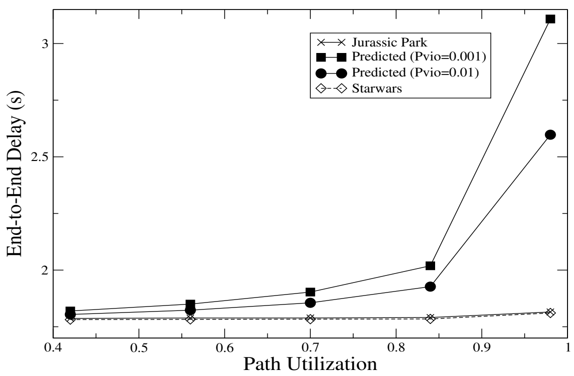

Fig. 7. Predicted delay and simulation results for a GT-ITM generated topology with traffic from movie traces

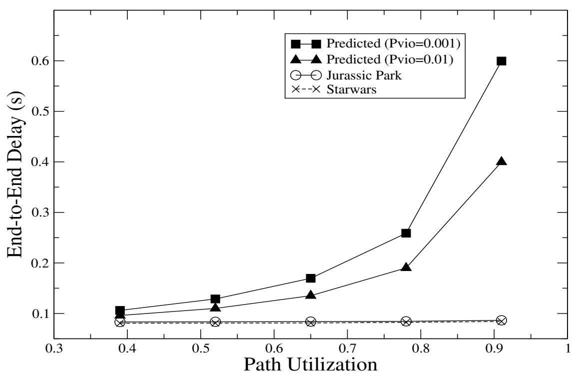

Fig. 8. Predicted and Simulated Delay between Canberra and Darwin

value of ρ^{max} has to be decided depending on the typical size of aggregates that are admitted into the network. Given this value and for $P_{vio} = 0.001$ we consider choosing the utilization target and σ^{max}.

Consider a Path i with capacity Γ_i (as given in Definition III.2). Figure 6 depicts the tradeoff that is inherent in a choice of utilization target with respect to the delay that can be assured. The path under consideration is from the Telstra topology, connecting Canberra to Adelaide. For a specific utilization target, we compute the end-to-end delay assurance as the ratio of σ^{max}/Γ_i increases. Thus if we have a requirement for a delay d', we could choose whether to set a higher utilization target or allow higher burstiness for flows. One way to compute the maximum busy period β^{max} would then be $\sigma^{max} \sum_i u_i \Gamma_i/(C\rho^{max})$, where u_i, Γ_i refer to the paths through the ingress under consideration. This β^{max} would then be specific to that ingress. This was the method used in the simulations.

C. Simulation Results

In order to verify the delay bounds we conducted simulation studies on the GT-ITM generated topology and the Telstra topology. In order to understand the topological significance to the results of the analysis, we consider the effects of path length and node degree (number of links incident at a node). In the succeeding sections, the 95% confidence intervals were too small for the scale of the graph and hence are not shown.

1) Accuracy of the bounds: Figures 7 and 8 show the comparison between predicted delays and the measured delays in simulation. We consider results for two trace driven simulations (Starwars and Jurassic Park) for a path in GT-ITM topology and one in Telstra (Canberra to Darwin). The X-axis denotes the path utilization (note that a multiplexer can have multiple paths passing through it and hence a path utilization of 0.9 does not mean the multiplexer is utilized at 0.9).

For lower utilization values, the simulation results are close to the predicted delays. As we increase the utilization factor, the prediction diverges and becomes quite conservative for utilization greater than 0.75. This is due to the fact that the analysis considers a worst-case scenario for each flow. Since in reality, a flow might not always find a queue due to the burst parameters of other flows, the predicted delays are conservative. However, note that P_{vio} allows us to choose how conservative the provisioning can be. Lower the value of P_{vio}, lower the delay targets.

2) Effect of path length: One of the most important aspects of multiplexing across multiple hops is the potential increase in burstiness of the traffic. For the same path capacity one would expect that the delays assured are much higher for a longer path. We examine this scenario in Figure 9. For shorter path lengths, the delay bound is fairly accurate. As the path length increases, the delay bound becomes more conservative. For very long path lengths, the bound may be too high to be useful as an assurance. However, as we see in this experiment path lengths of up to 8 hops were fairly accurately modeled.

3) Effect of node degree: Another interesting aspect of the topology that affects the delay bounds is the degree of a node. The number of links feeding into a multiplexer is referred to as the degree. To understand why this is im-

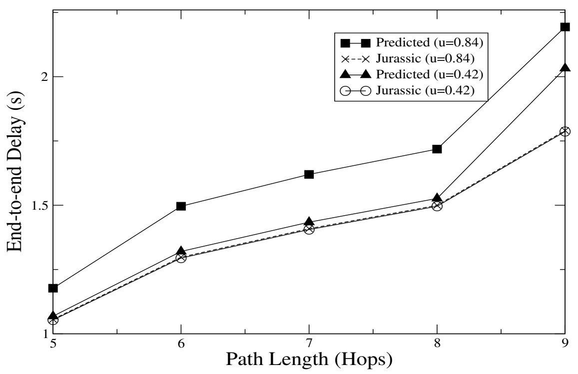

Fig. 9. Increase in End-to-end delay bound with increasing path length (GT-ITM)

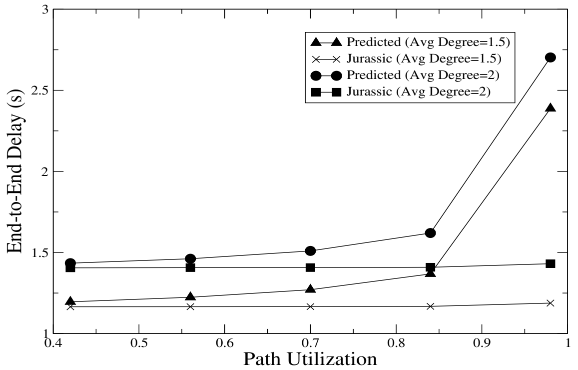

Fig. 10. Increase in End-to-end delay bound with increasing average node degree (GT-ITM)

portant, note that a multiplexer which is fed with multiple paths will probably have a higher number of flows. Intuitively, the probability of delay violation is higher when there are more number of flows with comparable burst parameters.

To illustrate this, we choose two paths in the GT-ITM topology with the *same length* (number of hops) and *capacity*. The paths are distinguished by their *average degree*, i.e., the sum of the degrees of each multiplexer on the path divided by the path length. Figure 10 shows the results of the experiment with paths from the GT-ITM topology. For the path with higher average node degree we observe that the measured delay in simulation and the predicted delays are higher. Thus while evaluating a path for delay assurances, the average node degree is an important measure.

V. Conclusions and Future Work

The problem of delay assurances in FIFO networks is an important one. In order to measure the delays that can be assured by a given network topology, we must be able to set a delay target for each hop, that can be met at a given probability level. We provided analytical tools to arrive at delay targets for a FIFO network independent of exact traffic description, given some restrictions on the nature of flows that can be admitted. We demonstrated an admission control algorithm that is scalable and does not require per-hop signaling, to enforce these assumptions.

In order to evaluate the analytical results we considered a generated topology and a real Internet topology. We found that, for utilization targets in the range of $0.4-0.75$, the model provides good delay estimates. For higher utilization targets, the worst-case nature of the analysis leads to conservative predictions. We evaluated the capabilities of the chosen topologies for a selected set of paths. The dependence of end-to-end delay bounds on topological features such as degree and path length was explored. Given a FIFO network topology, we can provide end-to-end delay assurances for selected paths, under limitations on path length and degree. Future work will involve implementing the framework on a real testbed and measuring the gains for delay sensitive traffic.

References

[1] M. Andrews,"Instability of FIFO in session-oriented networks," *Proc. of 11th ACM-SIAM Symposium on Discrete Algorithms*, 2000.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, *An Architecture for Differentiated Services*, Dec. 1998, IETF RFC 2745.

[3] J.-Y. Le Boudec and P. Thiran, *Network Calculus*, Springer-Verlag, 2001.

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

[5] R.R. Boorstyn, A. Burchard, J. Liebeherr, C. Oottamakorn, "Statistical service assurances for traffic scheduling algorithms," *IEEE J. Select. Areas Commun.*, Vol. 18 No. 12, pp. 2651 -2664, Dec. 2000.

[6] C.S. Chang, "Stability, Queue Length, and Delay of Deterministic Stochastic Queueing Networks," *IEEE Trans. Automat. Control*, Vol. 39 No. 5, pp. 913-931, May 1994.

[7] A. Charny, J.-Y. Le Boudec, "Delay Bounds in a Network with Aggregate Scheduling," *Proc. of Quality of Future Internet Services*, Berlin, Germany, October 2000.

[8] R.L. Cruz, "Sced+: Efficient Management of Quality of Service Guarantees," *IEEE INFOCOM 98*, Vol. 2, pp. 625-634, 1998.

[9] R.L. Cruz, "A calculus for network delay, part I: Network elements in isolation," *IEEE Trans. on Information Theory*, Vol.37., No. 1, pp. 114-131, Jan. 1991.

[10] R.L. Cruz, "A calculus for network delay, part II: Network Analysis," *IEEE Trans. on Information Theory*, Vol.37., No. 1, pp. 132-141, January 1991.

[11] B. Davie, A. Charny, J.C.R. Bennett, K. Benson, J.-Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)," RFC 3246, March 2002.

[12] D. Ferrari, "Design and Application of a Delay Jitter Control Scheme for Packet-switching Internetworks," *Computer Communications*, 15(6):367-373, July 1992.

[13] D. Ferrari, D.C. Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE J. Select. Areas Comm.*, Vol. 8 No. 3, pp. 368-379, April 1990.

[14] V. Firoiu, J.-Y. Le Boudec, D. Towsley and Z. Zhang, "Theories and Models for Internet Quality of Service,"*Proc. of the IEEE*, Vol. 90 No. 9, pp. 1565-1591, Sep. 2002.
[15] Y. Jiang, "Delay Bounds for a Network of Guaranteed Rate Servers with FIFO Aggregation," *IEEE ICC 2002*, Vol. 2, pp. 1149-1153, 2002.
[16] E.W. Knightly, "Second Moment Resource Allocation in Multi-Service Networks," *Proc. of ACM SIGMETRICS 1997*, Vol 25 No. 1, pp. 181-191, Jun. 1997.
[17] E.W. Knightly, "D-BIND, An Accurate Traffic Model for Providing QoS Guarantees to VBR Traffic," *IEEE/ACM Trans. on Networking*, Vol. 5 No. 2, pp. 219-231, April 1997.
[18] E.W. Knightly, "Enforceable Quality of Service Guarantees for Bursty Traffic Streams," *IEEE INFOCOM 98*, Vol. 2, pp. 635-642, 1998.
[19] J. Kurose, "On Computing Per-session Performance Bounds in High-Speed Multi-hop Computer Networks," *Proc. of ACM SIGMETRICS 1992*, Vol. 20 No. 1, pp. 128-139, 1992.
[20] J. Liebeherr, S. Patek and E. Yilmaz, "Tradeoffs in designing networks with end-to-end statistical QoS guarantees,"*Eighth International Workshop on Quality of Service*, pp. 221-230, 2000.
[21] J. Liebeherr, S. Patek and A. Burchard, "A Calculus for End-to-end Statistical Service Guarantees," *Univ. of Virginia Tech. Report CS-2001-19* (revised).
[22] G. Mayor, J. Silvester, "Time Scale Analysis of an ATM Queueing System with Long-Range Dependent Traffic," *IEEE INFOCOM 97*, Vol. 1, pp. 205-212, 1997.
[23] A.K. Parekh, R.G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-node Case," *IEEE/ACM Trans. on Networking*, Vol. 1 No. 3, pp. 344-357, Jun. 1993.
[24] A.K. Parekh, R.G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the multiple node case," *IEEE/ACM Trans. on Networking*, Vol. 2 No. 2, pp. 137-150, Apr. 1994.
[25] M. Reisslein, K.W. Ross and S. Rajagopal, "A Framework for Guaranteeing Statistical QoS," *IEEE/ACM Transactions on Networking*, Vol 10 No. 1, pp. 27-42, Feb. 2002.
[26] V. Sivaraman and F. Chiussi, "Providing end-to-end statistical delay guarantees with earliest deadline first scheduling and per-hop traffic shaping," *IEEE INFOCOM 2000*, Vol. 2, pp. 631-640, 2000.
[27] V. Sivaraman and F. Chiussi, "End-to-end Statistical Delay Service under GPS and EDF Scheduling: A Comparison Study," *IEEE INFOCOM 2001*, Vol. 2, pp. 1113-1122, 2001.
[28] N. Spring, R. Mahajan and D. Wetherall, "Measuring ISP Topologies with Rocketfuel," *ACM SIGCOMM 2002*
[29] D. Starobinski, M. Sidi, "Stochastically bounded burstiness for communication networks," *IEEE Transactions on Information Theory*, Vol.46 No. 1, pp. 206-212, Jan. 2000.
[30] O. Yaron, M. Sidi, "Performance and Stability of Communication Networks via Robust Exponential Bounds," *IEEE/ACM Transactions on Networking*, Vol. 1 No. 3, pp. 372-385, Jun. 1993.
[31] M. Vojnović and J.-Y. Le Boudec, "Stochastic Analysis of Some Expedited Forwarding Networks," *IEEE INFOCOM 2002*, Vol. 2, pp. 1004-1013, 2002.
[32] E.W. Zegura, K.L. Calvert and S. Bhattacharjee, "How to model an internetwork," *IEEE INFOCOM 96*, Vol. 2, pp. 594-602, 1996.
[33] H. Zhang and D. Ferrari,"Rate-Controlled Service Disciplines,"*Journal of High Speed Networks: Special Issue on Quality of Service*, 3(4), 1994.
[34] Z. Zhang, Z. Duan, Y.T. Hou, "Fundamental trade-offs in aggregate packet scheduling,"*Ninth Intl. Conf. Network Protocols*, pp. 129 -137, 2001.
[35] Z. Zhang, D. Towsley, J.F. Kurose, "Statistical Network Performance Guarantees with Generalized Processor Sharing Scheduling," *IEEE J. Select. Areas Commun.*, Vol. 13 No. 6, pp. 1071-1080, Aug. 1995.

APPENDIX

In §III we obtained bounds on σ_i, β, $\sum_i \rho_i$ and $\sum_i \rho_i\sigma_i$. We examine two existing proposals [18] [31] to compute $Pr\{D_i > d_i\}$ using these bounds.

A. Rate-Variance Envelopes

Consider the process $X(t)$, indicating the cumulative arrivals in $[0, t)$. Define the Rate Variance envelope for a flow j as:

$$RV_j(t) = Var\left(\frac{X_j(t, t+\tau)}{\tau}\right)$$

If the flow is stationary and is upper bounded such that $X_j(t, t+\tau) \leq b_j(t)$ for all $\tau, t > 0$ then,

$$RV_j(t) \leq \phi_j \frac{b_j(t)}{t} - \phi_j^2$$

$$\phi_j = \lim_{t\to\infty} \frac{b_j(t)}{t}$$

Now to specialize this for a leaky bucket shaped flow (σ_j, ρ_j), we note that $\phi_j = \rho_j$ and $b_j(t) = \rho_j t + \sigma_j$. Hence, $RV_j(t) = \rho_j\sigma_j/t$. Further, for a multiplexer with N flows,

$$Pr\{D > d\} \approx \max_{0\leq\tau\leq\beta_i} \frac{1}{\sqrt{2\pi}} exp\left(-\frac{(C(\tau+d_i) - m_t)^2}{2v_t^2}\right)$$

$$\begin{aligned} m_t &= \sum_{j=1}^{N} t\phi_j \\ &= t\sum_{j=1}^{N} \rho_j \\ v_t^2 &= \sum_{j=1}^{N} t^2 RV_j(t) \\ &= t\sum_{j=1}^{N} \sigma_j\rho_j \end{aligned}$$

B. *Large Deviations Approximation*

Consider a scheduler with service curve $\beta(t) = Ct$. The following bound then holds [31].

$$Pr\{D_i > d_i^*\} \leq$$

$$\sum_{k=0}^{K} exp\left(-\frac{2[C(\tau_k + d_i) - \tau_{k+1}\sum_{j=1}^{N}\rho_j)^+]^2}{(\tau_{k+1}\sum_{j=1}^{N}\rho_j + \sqrt{\sum_{j=1}^{N}\sigma_j^2})^2 \bigwedge (4\sum_{j=1}^{N}\sigma_j^2)}\right) \tag{12}$$

Substituting for $\sum_j \rho_j$ and $\sum_j \sigma_j^2$ from §III-B we have the required bound.