Provisioning IP Backbone Networks to Support Latency Sensitive Traffic

Chuck Fraleigh and Fouad Tobagi
School of Electrical Engineering
Stanford University
Stanford, CA 94305
Email: {cjf,tobagi}@stanford.edu

Christophe Diot
Sprint Advanced Technology Labs
1 Adrian Court
Burlingame, CA 94010
Email: cdiot@sprintlabs.com

***Abstract*— To support latency sensitive traffic such as voice, network providers can either use service differentiation to prioritize such traffic or provision their network with enough bandwidth so that all traffic meets the most stringent delay requirements. In the context of wide-area Internet backbones, two factors make overprovisioning an attractive approach. First, the high link speeds and large volumes of traffic make service differentiation complex and potentially costly to deploy. Second, given the degree of aggregation and resulting traffic characteristics, the amount of overprovisioning necessary may not be very large. This study develops a methodology to compute the amount of overprovisioning required to support a given delay requirement. We first develop a model for backbone traffic which is needed to compute the end-to-end delay through the network. The model is validated using 331 one-hour traffic measurements collected from the Sprint IP network. We then develop a procedure which uses this model to find the amount of bandwidth needed on each link in the network so that an end-to-end delay requirement is satisfied. Applying this procedure to the Sprint network, we find that satisfying end-to-end delay requirements as low as 3 ms requires only 15% extra bandwidth above the average data rate of the traffic.**

I. Introduction

IP networks carry many types of traffic. Some traffic, such as web and email, can tolerate long queuing delays which occur during periods of network congestion. Other traffic, such as voice, audio, and video, have unacceptable performance if long delays are incurred. To provide low delay service for such applications, there are two basic approaches which can be used. One option, known as service differentiation, is to give preferential treatment to latency sensitive traffic. The second option, known as bandwidth provisioning, is to provide sufficient bandwidth so that all traffic meets the most stringent delay requirement.

In the context of IP backbone networks, two factors make the bandwidth provisioning approach attractive. First, there are costs associated with traffic differentiation. While some of this cost is related to additional complexity required in network routers, much of the cost is associated with the management and operation of the network. Installers must be trained to configure the traffic differentiation mechanisms when routers are installed in the network and network operators must be trained to manage the different traffic classes. Second, traffic differentiation may not provide much benefit in backbone networks. Traffic in backbone networks is aggregated from many thousands of users. As a result of the high degree of aggregation, as well as the low packet transmission times (a 1500 byte packet takes only 5 μs to transmit on a 2.5 Gb/s OC-48 link), it is expected that queuing delays in backbone networks will be low, and therefore little overprovisioning is required.

This paper investigates the amount of overprovisioining required in backbone networks. Using a set of 331 one-hour traffic measurements from the Sprint IP network, we develop a procedure to evaluate the amount of bandwidth needed on each link in the network in order to meet a given delay requirement.

A. Bandwidth provisioning

Backbone IP networks provide high bandwidth connectivity across a wide geographic area. A backbone network consists of a set of nodes, known as Points-of-Presence (POPs), connected by high speed links. For example, a backbone may have POPs in New York, Chicago, and San Francisco connected by 10 Gb/s links. Customers of the backbone ISP connect to the network at one or more of the POPs.

Bandwidth provisioning is the process by which a backbone ISP determines the amount of bandwidth needed on each of the links in order to support a desired level of performance. For real-time applications such as voice, a reasonable method to specify this performance requirement is in terms of a probabilistic delay requirement of the form $P[d^{(i,j)} > D_{req}] < \epsilon$, that is the probability that the delay between POP i and POP j exceeds D_{req} is less than ϵ.

It is important to emphasize that this delay requirement is the same between all pairs of POPs and for all types of traffic. Since traffic differentiation is not used, it is not possible to offer one level of service to data traffic a higher level of service to real-time traffic. It is also not possible to offer one level of service to one customer and a second level of service to another customer. All traffic receives the same service, and this service must be sufficient to meet the needs of the most stringent application. While this may seem inefficient, we will see that supporting end-to-end queuing delay requirements as low as 3 ms requires bandwidth only marginally greater than the average traffic volume.

To provision the network, the network provider must know the traffic demand between each pair of POPs, and the path each of these traffic demands follows through the network. These demands can be forecast using techniques such as [15]. With this information, the bandwidth required on each link is found by solving a network optimization problem known as the Capacity Assignment (CA) problem.

The CA problem has been solved for networks where the traffic demands are modeled as a Poisson process and where the objective is to minimize the average delay [11]. Using the Kleinrock independence approximation and Jackson's theorem one can derive expressions for the average queuing delay. Given an expression for the average delay, techniques such as Lagrangian relaxation are used to find the bandwidth assignment which minimizes the total network cost, where cost is a function of the bandwidth on each link in the network.

Our problem is different in several respects. First, we consider probabilistic requirements rather than average delay requirements. Second, the Poisson model has been shown to not be an accurate model for actual network traffic [16], [20]. Third, many solutions to the CA problem allow a link to have any possible capacity. In an actual network, a the link capacity must be selected from a discrete set (e.g. 155 Mb/s or 622 Mb/s).

In order to solve the CA problem, we therefore need:

- A realistic model for the traffic demand between POPs in a backbone network.
- A method to assess end-to-end queuing delay using this model.
- An procedure to find the bandwidth needed on each link in order to meet the delay constraints.

We develop a model for backbone traffic by analyzing traffic measurements from the Sprint IP backbone. We find that backbone traffic is significantly easier to model than traffic considered in prior studies such as [9], [8], and [17]. These studies found that at time scales less than 100 ms, the distribution of the traffic arrival process is quite complex. However, the average traffic arrival rate of the measurements used in these studies was between 100 kb/s and 10 Mb/s. We find that once traffic volume reaches 50 Mb/s (and for some traffic between 5 Mb/s and 50 Mb/s), the distribution of the traffic arrival process becomes Gaussian at small time scales and can be modeled using an extension of Fractional Brownian Motion (FBM) [13]. We call this model *two-scale FBM*.

We then develop a method to compute the end-to-end delay through a network where all of traffic demands between POP pairs are modeled using two-scale FBM. Using this method, we develop an algorithm to find the bandwidth needed on each link in the network so that the end-to-end delay delay requirement is satisfied. The remainder of the paper is organized as follows. Section II presents the two-scale FBM model and derives an expression for the delay distribution for a queue fed by a two-scale FBM process. Using the model we evaluate the maximum utilization that may be achieved on a single link while meeting a particular delay requirement. Section III presents the method to compute the end-to-end delay through the network. Section IV describes the algorithm to find the minimum cost network and applies it to the Sprint network. Section V concludes and discusses areas of future research.

II. Traffic Model

In a backbone network, the traffic demand between a pair of POPs is the aggregate of traffic from many individual users. In this section we develop a model for such aggregate traffic and derive the delay distribution for a queue fed by such traffic.

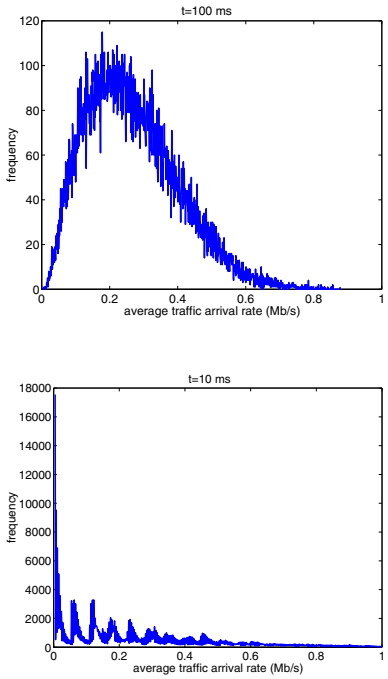

Fig. 1. Distribution of A_t/t for *DEC-WRL-2*

The model must capture the characteristics of the traffic which affect the queueing delay. We therefore begin by reviewing the procedure used to compute queuing delay distributions. Consider an infinite buffer queue with a constant bit rate server of capacity C. Let $A[s,t]$ be the amount of traffic that arrives at the queue over the time interval $(s,t]$, and let $A_t = A[-t,0]$. The queue length at time 0 is

$$Q = \sup_{t\geq 0}(A_t - Ct)$$

and the probability that the queue length exceeds x is

$$P[Q > x] = P[\sup_{t\geq 0}(A_t - Ct) > x]$$

This expression is difficult to evaluate, so we use the lower bound

$$P[\sup_{t\geq 0}(A_t - Ct) > x] \geq \sup_{t\geq 0} P[A_t > x + Ct] \qquad (1)$$

This may seem to be a rather crude approximation, but it has been shown to be logarithmically accurate for large x [7].

The queueing delay experienced by a packet of size b bits is the sum of the waiting time in the queue, $\frac{x}{C}$, and the service time of the packet, $\frac{b}{C}$. The distribution of the waiting time, W, is found directly from the queue length distribution $P(W > d) = \sup_{t\geq 0} P(A_t > C(d+t))$. We do not model the service time distribution, as it is significantly smaller than the delay bounds in which we are interested. On an OC-3 link (one of the lowest speed backbone links), the transmission time of a maximum size packet is only 80 μs.

Name	Start time	Average data rate
T1	Wed9Aug00 9:56am	74.6 Mb/s
T2	Wed9Aug00 9:56am	90.1 Mb/s
T3	Wed9Aug00 9:56am	56.8 Mb/s
T4	Wed5Sep01 10:00am	219 Mb/s
T5	Wed5Sep01 10:00am	103 Mb/s
T6	Wed5Sep01 10:00am	132 Mb/s
T7	Wed5Sep01 10:00am	171 Mb/s
T8	Wed5Sep01 10:00am	208 Mb/s
T9	Wed5Sep01 10:00am	179 Mb/s

TABLE I
TRACE DATA

A. Traffic Characteristics

From (1), the dominant characteristic which affects queuing delay is the marginal distribution of the traffic arrival process A at different time scales, t. Prior measurement studies have found that at time scales greater than several hundred milliseconds the distribution of A_t is approximately Gaussian, while for t less than several hundred milliseconds the distribution of A_t is quite complex [16], [20], [9].

To illustrate this point, we present results from a measurement known as *DEC-WRL-2*, collected on DEC's primary Internet connection and used in [16]. To compute the marginal distribution at time scale t, we divide the *DEC-WRL-2* measurement into non-overlapping blocks of size t and compute the number of bits that arrive over each of these blocks (e.g. we compute the number of bits that arrive over every 100 ms time interval). Fig. 1 plots the empirical distribution of $\frac{A_t}{t}$ at 100 ms and 10 ms time scales for *DEC-WRL-2*[1]. At t = 100 ms, the distribution appears approximately Gaussian, while at t = 10 ms the distribution is clearly non-Gaussian.

DEC-WRL-2 is an accurate representation for WAN backbone traffic as observed in the early 1990s. However, traffic volume has substantially increased from the average rate of 267 kb/s observed in *DEC-WRL-2*. To investigate the characteristics of high volume traffic, we make use of measurements from the Sprint IP backbone network collected on 25 links in August 2000, July 2001, and September 2001. The set of links included 155 Mb/s OC-3 links and 622 Mb/s OC-12 links, and included a variety of link types including links to peering points, links to large web servers, links to large dial-up networks, and links to large corporations. The measurements are one hour traces which contain the arrival time, packet size, and first 40 bytes of every packet transmitted on a link. The packet timestamps are synchronized to a global GPS reference clock and are accurate to within 5 μs. The complete set of measurements includes 331 traces. Table I gives the start time and average traffic volume for nine of the traces considered in detail later in the paper. While all 9 of these traces were collected at 10 am on a Wednesday, the complete set of traces contains data from multiple days of the week and every hour of the day.

[1]We have plotted the distribution of the traffic rate (A_t/t) rather than the distribution of the traffic volume (A_t) to normalize the x-axis.

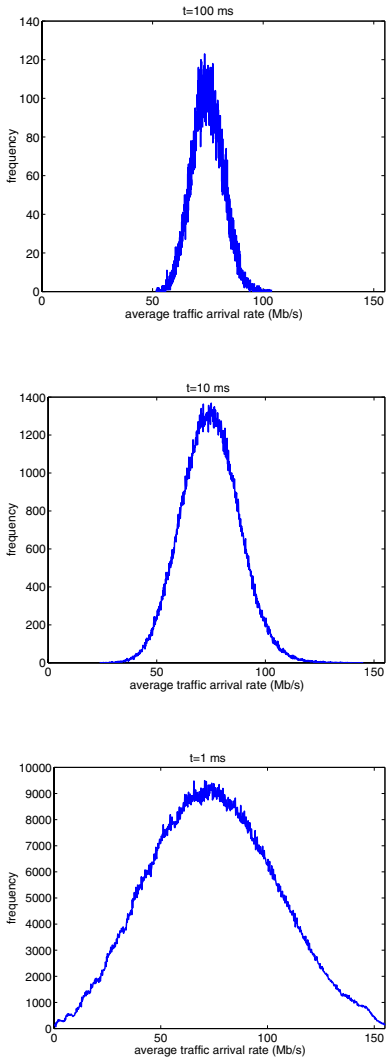

Fig. 2. Marginal distribution of traffic arrivals for trace T1

Our goal is to model the traffic demand between two POPs in a backbone network, which is slightly different than the traffic observed in the measurements. The measurements were collected from links within a single POP and represent a fraction of the entire traffic demand between two POPs. However, both the inter-POP traffic demands and these measurements are both aggregates of a large number of individual user connections. As a result, both are expected to exhibit the same characteristics. Later in this paper we will confirm that increasing the traffic volume to the range of inter-POP traffic demands does not change the fundamental characteristics presented in this section. We therefore present the traffic model by studying the characteristics observed in a single trace.

We begin by studying the characteristics of trace T1. Fig. 2 plots the distribution of A_t for T1 at time scales of 100 ms, 10 ms, and 1 ms. At t = 100 ms it appears Gaussian with mean 74.7 Mb/s and variance 49.5 $(Mb/s)^2$. This distribution is similar

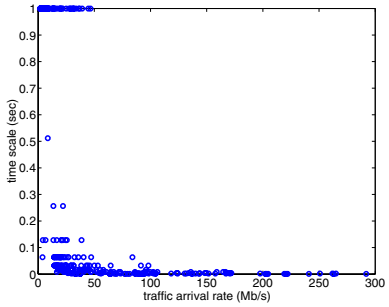

Fig. 3. Minimum time scale at which marginal distributions are Gaussian

to the distribution of *DEC-WRL-2* at the 100 ms time scale. At the 10 ms time scale, the T1 trace is much different from *DEC-WRL-2*. Rather than exhibit a complex distribution similar to that shown in Fig. 1(b), T1 appears Gaussian with mean 74.7 Mb/s and variance 178 $(Mb/s)^2$. Even at a time scale of 1 ms, T1 appears Gaussian with mean 74.7 Mb/s and variance 852 $(Mb/s)^2$.

To determine if these distributions are in fact Gaussian, we use a statistical test known as the Kolmogorov-Smirnov test (K-S test) for normality [4]. Applying the K-S test to each distribution confirms that they are consistent with a Gaussian distribution. Repeating this process for time scales up to 60 seconds and down to 100 μs finds the distributions at these time scales are also Gaussian. Below 100 μs the distributions are quite complex, but we do not need to consider these since we are interested in queuing delays on the order of milliseconds.

The reason for the difference between T1 and *DEC-WRL-2* is that T1 is aggregated from a very large population of users. The *DEC-WRL-2* measurement and nearly all other traffic measurements used in the prior studies, have average traffic volume between several hundred kb/s and 2 Mb/s. This represents traffic from a relatively small number of users (10,000 user connections over a one hour period for the measurement used in [20]). The T1 trace has an average traffic volume of 75 Mb/s, and has nearly 5 million unique user connections.

A natural question to ask is: how much aggregation is needed before the marginal distribution at small time scales becomes Gaussian? We can investigate this by considering the complete set of 331 one hour traffic measurements. For some of the measurements (especially for those collected between 1:00 am and 4:00 am), the average traffic arrival rate can reach as low as 1 Mb/s. For other measurements collected during afternoon hours on highly utilized links, the traffic volume can reach almost 300 Mb/s. Using the K-S test, we can determine which of these traces have Gaussian marginal distributions at small time scales, and which have the more complex distributions.

More precisely, we compute the marginal distribution of the traffic arrival process at time scales from 1 ms to 1 sec for each of the traces. At each time scale we apply the K-S test to determine if the distribution is Gaussian. For each trace we find the smallest time scale at which the marginal distribution is Gaussian.

Fig. 3 plots the minimum Gaussian time scale versus the mean arrival rate of the traffic for each of the traces. We see that for all but four traces with traffic volume greater than 50 Mb/s, the minimum time scale is between 1 ms and 8 ms. These traces have characteristics very similar to those shown for T1. Below 50 Mb/s there is much more variation. Two-thirds of the traces with traffic volume between 5 Mb/s and 50 Mb/s have a minimum Gaussian time scale between 1 ms and 64 ms, while one-third exhibit distributions similar to those shown for the *DEC-WRL-2* measurement. For traces with traffic volume less than 5 Mb/s, the marginal distributions are never Gaussian. All of these low volume traffic measurements resemble the *DEC-WRL-2* traffic. This confirms the results of [9], which found that for low volume traffic, the distribution at small time scales is quite complex. However, as the traffic volume increases, these distributions become much less complex. During the busy hour of the day traffic volume on nearly all backbone links is greater than 50 Mb/s and the distributions are expected to be Gaussian.

It should be noted that there are situations where 50 Mb/s traffic will not have enough aggregation to use Gaussian models. Consider, for example, a link carrying three 20 Mb/s HDTV video streams. The bandwidth guidelines we present are only valid for the traffic with the same mix of user connections that we see in today's backbone. In particular, the traffic must be aggregated from a large population of users, and the rate of an individual user should be much less than the rate of the total traffic aggregate.

Since a Gaussian distribution is fully specified by its mean and variance, for backbone traffic it is sufficient to know the mean and variance of A_t at each time scale t. The mean remains the same for all time scales as seen from Fig. 2. The variance, however, changes from one time scale to the next. The relationship between the variance and time scale can be studied using a technique known as the variance-time (VT) plot. This is simply a plot of the variance versus the time scale t.

Before proceeding, it is important to note that prior studies, such as [9], have demonstrated that the VT plot may not provide much information about the structure of network traffic at time scales less than 100 ms. The reason for this is that the variance does not provide sufficient information to describe a distribution similar to that shown in Fig. 1(b). For such distributions one needs to know information about the higher moments. From Fig. 2, however, we can see that for large traffic aggregates the distribution at small time scales is Gaussian and can therefore be fully described using only the mean and variance. The VT plot, therefore, provides enough information to completely characterize the traffic arrival process for backbone traffic. Later in this section we will address the statistical bias that may be introduced by using the VT plot to estimate the variance at small time scales.

Fig. 4 shows the VT plot for trace T1. The variance exhibits a two-piece linear relationship with the time scale t. It decays quite rapidly at time scales between 1 ms and 75 ms, and starts to decay more slowly after that point. The slow decay of the variance at large time scales is indicative of a statistical property known as *long-range dependence* (LRD) which has been observed in network traffic [16], [20]. For traffic with

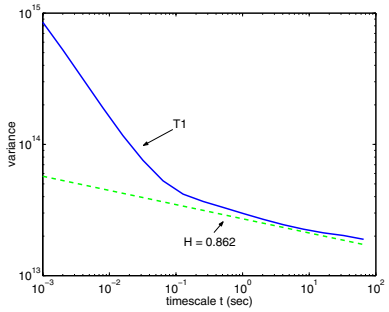

Fig. 4. Variance-time plot for T1

LRD, the variance of the traffic arrival rate decays as a power of the time scale t

$$var(A_t/t) \sim t^{2H-2},\ as\ t \to \infty$$

where H is known as the Hurst parameter and takes a range of $0.5 < H < 1$. For all the other traces, the variance exhibits the same two piece linear behavior as seen in Fig. 4. For 88% of the traces, the transition point between the two linear regions occurs at time scales between 75 ms and 400 ms.

We now investigate the causes of these two linear regions. At large time scales, the LRD of network traffic has been shown to be the result of the heavy-tailed distribution of individual user connection sizes [20]. A heavy-tailed distribution is one in which $P(X > x) \sim x^{-\alpha}, 1 < \alpha < 2$, as $x \to \infty$. The Hurst parameter is directly related to the α parameter of the connection size distribution according to $H = (3 - \alpha)/2$ [20].

To validate that the user connection size distribution is responsible for the large time scale behavior observed in T1, we compute both α and H for trace T1. To avoid the statistical bias of the VT plot, we use use a wavelet estimator developed by Abry and Veitch [19] which yields H = 0.862 for T1. To estimate α we use the Hill estimator as described in [20], and validate it using the procedure described in [5]. We find the connection size distribution is heavy-tailed with α = 1.30 indicating H should be 0.85, which is within the confidence intervals of the Abry-Veitch estimator. For reference, in Fig. 4, we plot a line corresponding to an LRD process with a variance that decays with H = 0.862.

Next we investigate the behavior of the variance at time scales less than 75 ms. We see from Fig. 4 that the variance at small time scales has a linear relationship with t, but the the variance is much higher than can be explained by the connection size distribution. The reason for this is that the theory relating the connection size distribution to H considers user connections to be constant bit rate (CBR). In a real network, however, user connections are far from CBR. In T1, as well as all but five other traces, over 90% of the traffic in the network is generated by TCP. TCP connections transmit a burst of packets corresponding to the TCP window size, wait one round-trip-time (rtt) for the acknowledgment, and then transmit another burst of packets. At time scales greater than the rtt, it has been empirically demonstrated that the connections can be approximated as CBR streams [20]. However, as the time scale falls below the rtt, individual TCP connections become much more variable than CBR streams resulting in the higher variance.

A direct relationship between the rtt and the break point between the two scaling regions of the VT plot has been demonstrated though the use of simulation [9]. This study performed a simulation where all connections had a rtt of 24 ms and a second simulation where all connections had a rtt of 610 ms. They found that the linear relationship between the variance and time scale which was observed at large time scales (i.e. the relationship due to the connection size distribution) broke down at a time scale just above the rtt of the user connections. In general, for each of the 331 one-hour measurements we study, we find that the transition point occurs "near" the median rtt of the connections observed in the traces[2]. For T1 in particular, the median rtt is 96.9 ms which is approximately the point at which the variance begins to rapidly increase.

However, we do not find a statistically significant correlation between the median or mean rtt and the breakpoint location. In general the rtt distribution is quite complex, and the mean or median value is insufficient to fully describe the distribution. As a result, we are unable to fully explain the exact location of the breakpoint and the cause of the linear behavior in the variance at small time scales. However, we are able to develop a model to capture this behavior and compute the resulting queuing delays.

B. Two-Scale Fractional Brownian Motion

To model backbone traffic we would like a process which has Gaussian marginals with a variance that obeys the two-piece linear relationship observed in the traces. Fractional Brownian Motion (FBM) is a process whose marginal distributions are Gaussian with a variance that has a single linear relationship with t and decays as t^{2H-2}. FBM was originally applied to network traffic by Norros [13], and accurately models the large time scale characteristics of network traffic. Modeling the small time scale characteristics, however, has been quite challenging. Cascade based models have been proposed to capture both the large and small time scale characteristics [9], [17], [8]. These models, however, were developed to capture the complex small time scale distributions shown in Fig. 1(b). Since large volume backbone traffic has Gaussian distributions at small time scales the complexities introduced by cascade models are unnecessary, and a simple extension to FBM is sufficient.

Multiscale FBM, $(M_K) - FBM$, is an extension of FBM with a Hurst parameter that varies at different time scales. We can therefore use one Hurst parameter, H_0, for large time scales and another Hurst parameter, H_1, for small time scales. $(M_K) - FBM$ has been used by Benassi and Deguy [2] for image synthesis and Bardet and Bertrand [1] to model biomechanic data.

There are several other processes which could also be used to represent the two-piece linear relationship between the variance and time scale. One such process is a traditional FBM with a periodic component. All such processes, however, will produce

[2]We estimate the rtt using the procedure described in [10].

the same results in terms of queuing delay. For our purposes all of these processes provide an equally accurate model of network traffic, so we have chosen $(M_K) - FBM$ to simplify the analysis.

To construct $(M_K) - FBM$, we start with the harmonizable representation of the traditional FBM process, $B_H(t)$ [18]

$$B_H(t) = \int_{-\infty}^{\infty} \frac{e^{i\omega t} - 1}{C(H)\,|\omega|^{H+1/2}} \widetilde{W}(d\omega)$$

$W(dx)$ is a Brownian measure and $\widetilde{W}(d\omega)$ is its Fourier transform, and $C(H) = \frac{\pi}{H\Gamma(2H)sinH\pi}^{1/2}$. $(M_K) - FBM$ is a generalization of this process where H is a function of the frequency (inverse of the time scale). We define an $(M_K) - FBM$, $X_\eta(t)$, as a process such that

$$X_\eta(t) = \int_{-\infty}^{\infty} \frac{e^{i\omega t} - 1}{\eta(\omega)} \widetilde{W}(d\omega)\ , -\infty < t < \infty$$

where

- $K \in N$, represents the number of Hurst parameters
- for $i = 0, 1, \ldots, K$ there exist $(\omega_i\ ,\ a_i\ ,\ H_i\ ) \in (R_+,\ R_+,\ (0.5, 1))$ such that $\eta(\omega) = \frac{C(H_i)|\omega|^{H_i+1/2}}{\sqrt{a_i}}$ for $\omega_i \le \omega < \omega_{i+1}$ with $0 = \omega_0 < \omega_1 < \cdots < \omega_K < \omega_{K+1} = \infty$
- $\eta(-\omega) = \eta(\omega)$

[1] has shown that $X_\eta(t)$ is a Gaussian process with stationary increments and variance at time scale δ, $var(\delta) = E[X_\eta(t + \delta) - X_\eta(t)]^2$, given by

$$var(\delta) = 4 \sum_{i=0}^{K} \delta^{2H_i} \frac{a_i}{C(H_i)^2} \int_{\delta\omega_i}^{\delta\omega_{i+1}} \frac{1 - cosv}{v^{2H_i+1}} dv \qquad (2)$$

To derive the queue length distribution for this process, we follow the same procedure Norros used to derive the queue length distribution for FBM [13]. Let A_η be the cumulative traffic arrival process

$$A_\eta(t) = mt + \sqrt{m} X_\eta(t)$$

m is the mean arrival rate of the traffic, and the term $\sqrt{m}X_\eta(t)$ describes the fluctuations around the mean.

We use the lower bound (1) to compute the queue length distribution. Since at time scale t, $A_\eta(t)$ has a Gaussian distribution with mean mt and variance $m \cdot var(t)$ the queue length distribution is

$$P[Q > x] = \sup_{t \ge 0} \overline{\Phi}\left(\frac{x + Ct - mt}{\sqrt{m \cdot var(t)}}\right) \qquad (3)$$

where $\overline{\Phi}$ is the residual distribution function of the standard Gaussian distribution.

In the general case for $M_k - FBM$ with many Hurst parameters, finding the t which maximizes the right hand side of (3) is difficult. However, our traffic has only two distinct scaling regions. We therefore consider specific the case when $K = 1$ and call this *two-scale FBM*. In this case we can use the following approximation

$$var(\delta) = \begin{cases} \delta^{2H_1} \frac{a_1}{C(H_1)^2} & , 0 \le \delta < \frac{1}{\omega_1} \\ \delta^{2H_0} \frac{a_0}{C(H_0)^2} & , \frac{1}{\omega_1} < \delta < \infty \end{cases} \qquad (4)$$

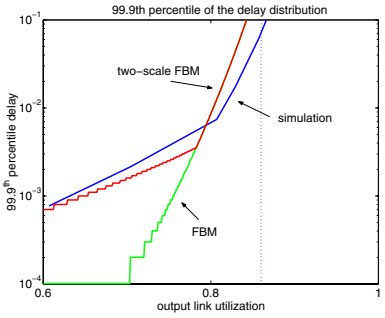

Fig. 5. Simulation delay and two-scale FBM delay for T1

where (a_1, H_1) represent the linear region at small time scales, (a_0, H_0) represent the linear region at large time scales, and $\frac{1}{\omega_1}$ is the transition point between the two regions.

Using (4), we find that (3) is maximized at

$$t = t^* = \begin{cases} \frac{H_1}{1-H_1} \frac{x}{C-m} & , x < x_c \\ \frac{H_0}{1-H_0} \frac{x}{C-m} & , x \ge x_c \end{cases}$$

and the queue length distribution is

$$P[Q > x] \sim \begin{cases} \exp(-\kappa(a_1, H_1)) x^{2-2H_1} & , x < x_c \\ \exp(-\kappa(a_0, H_0)) x^{2-2H_0} & , x \ge x_c \end{cases} \qquad (5)$$

where

$$\kappa(a, H) = \frac{(C-m)^{2H}}{2am(1-H)^{2-2H}(H)^{2H}}$$

$$x_c = \frac{(C-m)(1-H_1)}{H_1} e^{\frac{H_0 \log(\frac{H_1}{H_0}) + (H_0 - 1)\log(\frac{H_1 - 1}{H_0 - 1}) + \frac{1}{2}\log(\frac{a_1}{a_0})}{H_0 - H_1}}$$

A similar result can be derived for a queue fed by N independent two-scale FBM processes. If m_n and $var_n(t)$ are the mean and variance at time scale t for flow n, the queue length distribution is

$$P[Q > x] = \sup_{t \ge 0} \overline{\Phi}\left(\frac{x + Ct - \sum m_n t}{\sqrt{\sum m_n \cdot var_n(t)}}\right) \qquad (6)$$

Unlike the single flow case, we cannot use the variance approximation (4) because the variance of the aggregated flow has more than two distinct regions. As a result, (6) cannot be further simplified. While analytically cumbersome, (6) can be easily computed using Matlab or C programs.

C. Model Validation

To validate the model we compare the actual delay experienced by the measured traffic with the delay computed using the model. To determine the actual delay for the traffic we use a queuing simulator. The simulator reads a packet trace and simulates injecting the traffic into an infinite buffer queue served by a constant bit rate server.

There are three factors that would cause the delay in the simulator to be different than the delay that the traffic would see in the actual network. The first source of error is that a router does not implement an ideal FIFO queue. There may be effects due to routing lookups and other operations that affect the delay. In a prior study, we measured the delay experienced

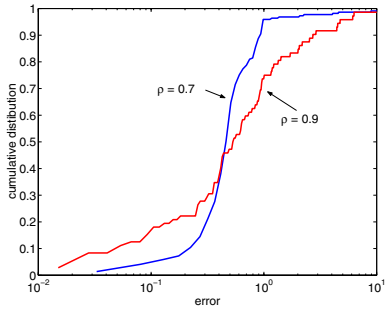

Fig. 6. two-scale FBM performance for 300 sample traces, ϵ = 0.001

through routers in the Sprint network and found that such router behavior has little effect on the total queuing delay [14].

The second source of error is the simulator emulates an infinite buffer queue, while the buffers in the actual network are finite. Typical buffer sizes in the actual network correspond to between 250 ms and 1 second of queuing delay. In the simulation results we present, the maximum delay observed in the simulator is typically less than 100 ms for the range of link utilizations in which we are interested. In such cases, no loss would be experienced in the actual network and there is no difference between the infinite buffer queue used in the simulation and the actual routers in the network.

The final source of error is that we do not consider the feedback mechanism of TCP. If we simulate a network in which there is a large amount of loss or large delays, the TCP congestion control mechanism would cause the sources to reduce their transmission rate. We do not account for this behavior in our simulator. However, as we shall see, over the range of link utilizations in which we are interested, only a small number of packets experience delays greater than several milliseconds and no packets experience loss. In such a case the TCP feedback mechanism would have minimal impact. To further address this point, in several of the figures we will indicate by a dotted line the point at which the maximum delay for any packet exceeds 250 ms. To the right of this line, some of the traffic may be affected by TCP effects. As we will see, the regions in which the network operates will be to the left of this line, so the results will be unaffected by the behavior of TCP.

To compute the delay using our model, we must estimate the model parameters from our traces. We use the Abry-Veitch estimator [19] to determine the H_0 and H_1 parameters and use linear regression on the variance-time plot to estimate a_0 and a_1. However, we do not know a priori over which time scales to estimate (a_0, H_0) and over which time scales to estimate (a_1, H_1). As we have seen in the previous section, the breakpoint between the two regions of the model typically occurs at time scales of 100 ms - 500 ms. We therefore do not consider this region and estimate (a_1, H_1) from the traffic characteristics at time scales of 2 ms - 64 ms, and we estimate (a_0, H_0) from the traffic characteristics at time scales of 512 ms - 2 min. These two regions are consistently above and below the breakpoint, respectively.

We first investigate how well the model estimates the delay for T1. The parameters for T1 are H_1=0.62, H_0=0.89, a_1=69.6 kb·sec, a_0=338 kb·sec, and m=75 Mb/s. We compare the delay distribution obtained using the model and the delay distribution obtained from the simulation for a range of output link utilizations, ρ. The results are shown in Fig. 5, which plots the 99.9th percentile of the delay distribution for different ρ [3]. This percentile corresponds to a delay requirement with ϵ = 0.001, a somewhat strict requirement for real time applications such as voice which can tolerate a small number of packets that exceed the delay requirement. For reference, we also show the delays that are predicted using the standard FBM model.

From the figure we see that the traditional FBM and the two-scale FBM models perform the same when the output utilization is high. In this region, the large time scale characteristics dominate the queuing performance. Both FBM and the two-scale FBM are accurate models for the large time scale characteristics, so they perform the same. At low utilization, the two-scale FBM model performs much better than traditional FBM. In this region t^* is less than several hundred msec. Since two-scale FBM is a much better model for the small time scale characteristics, the delay estimate is much more accurate.

Next we evaluate the model performance for the rest of the traces. It is not possible to repeat Fig. 5 for all traces. Instead we evaluate the model performance at a link utilization ρ = 0.7 and at ρ = 0.9. The performance at ρ = 0.7 determines how well the model fits before the knee of the curve shown in Fig. 5, and the performance at ρ =0.9 is indicative of how well the model fits after the knee. While $\rho = 0.9$ may not be a reasonable operating point for a commercial network as the delays are quite large, we would still like to evaluate the model performance in this region. We only show results for ϵ = 0.001 as this value showed the worst performance for T1 and most of the other traces.

Fig. 6 plots the difference in the delay estimated by the two-scale FBM model and the delay obtained in simulation: $error = \frac{|d_{two-scaleFBM} - d_{simulator}|}{d_{simulator}}$. From the Fig. we see that at $\rho = 0.7$, 80% of the flows have an error less than 0.75, and 96% of the flows have an error of less than 1. An error of 1 may seem to be quite large (100% error). However, in terms of actual delay, it represents a difference between 1 ms and 2 ms or 4 ms and 8 ms. For a reference, at ρ = 0.7, the results for the T1 trace shown in Fig. 5 have $d_{simulator}$ = 1.6 ms and $d_{two-scaleFBM}$ = 2.2 ms. This corresponds to an error of 0.37, close to the median error for all traces.

At ρ = 0.9 the model does not appear to perform as well. Only 75% of the flows have an error of less than 1. However, consider Fig. 5. Due to the rapid increase in delay, shifting one of the curves to the left or right can result in a very large difference between the two delay values. In fact, for T1, the error at ρ = 0.9 and ϵ = 0.001 is almost 10, one of the highest errors of all traces considered. We can consider the results shown in Fig. 5 to be among the worst of all traces we have studied. Furthermore, from a bandwidth provisioning point of view, the location of the knee of the curve shown in Fig. 5 is the most important aspect rather than the actual magnitude of the delay above the knee.

[3] Results for other percentiles are similar and can be found in [10].

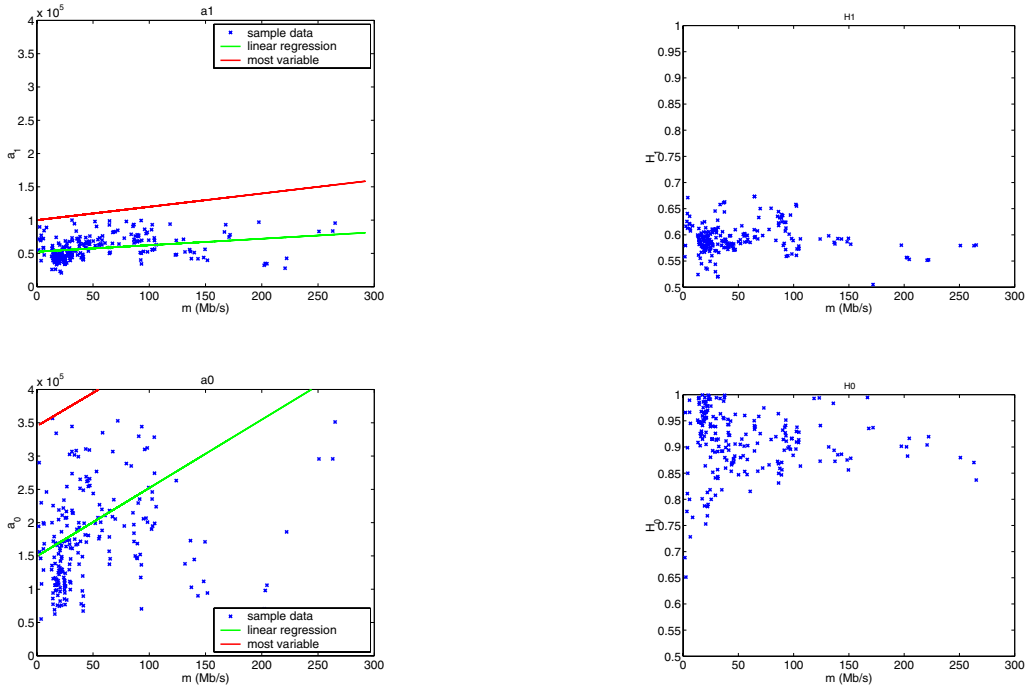

Fig. 7. two-scale FBM parameters for all traffic measurements

The model does accurately predict this knee location.

We have also performed simulations to validate (6), the queue length distribution of a queue fed by multiple two-scale FBM processes. We performed simulations of a queue fed by two flows. We consider the first flow to correspond to the T1 trace and the second flow to correspond to another trace, T2. Both of these traces were collected on input links to the same router over the same time interval. Our measurement systems are synchronized to within 5 μs using GPS, so it is reasonable to consider both of these measurements as representing the input to the same queue. The results show similar performance to that seen in Fig. 5 for the single flow delay. We omit the actual figure due to space constraints. Repeating this experiment with more than two traces produced similar results.

D. Bandwidth provisioning for a single link

Using the two-scale FBM model, we can make some interesting observations about the bandwidth requirements for a single link. A common question is: what is the maximum utilization at which a link can be operated while still meeting a particular delay requirement? The maximum achievable utilization on a link carrying two-scale FBM traffic can be computed directly from (5). (5) gives the delay distribution for a link of capacity C carrying traffic with an average arrival rate m. To find the maximum achievable utilization, we use binary search to find the maximum m which can be supported and still satisfy the delay constraint.

The only remaining question is what are the four model parameters, (H_0, a_0, H_1, a_1) for traffic with an arrival rate of m. Using the measurements, we can make some projections. Fig. 7 plots the model parameters for each trace against the traces's mean arrival rate, m. We see that the a_1 parameter (the variance at small time scales) exhibits a moderate increase with m. Using linear regression we find that $a_1 = 97m + 52562$ where m is in Mb/s. The parameter a_0, on the other hand, does not show as clear a trend. The best fit using linear regression finds the relationship $a_0 = 1027m + 149400$, but there is a wide range for the actual values of a_0. The H_1 and H_0 parameters seem to be relatively stable across all values of m. For very small m (less than 20 Mb/s) we do see a wide range of values for H_0, but they converge to a value of 0.90 as m increases. This is to be expected since H_0 is a function of the connection size distribution. The connection size distribution should not change as m increases, as long as we are multiplexing similar streams. H_1 converges to 0.59 in a similar fashion.

Since there is not a clear trend in the a_0 parameter, we consider a worst-case approximation which we call the "most variable" traffic. This represents the traffic with the highest variability seen in all of our measurements and corresponds to the "most variable" line shown in Fig. 7. For this traffic, we compute the maximum link utilization over a range of link capacities and plot the results in Fig. 8. We show results for two different maximum delay requirements, $D_{req} = 10ms$ and $D_{req} = 1ms$ and three different delay percentiles, $\epsilon = 0.01$, $\epsilon = 0.001$, and $\epsilon = 0.0001$. We consider $D_{req} = 10ms$ as it is comparable to the 20 ms propagation delay for backbone networks. With $D_{req} = 10ms$, the sum of the queuing and propagation delay would be 30 ms. $D_{req} = 1ms$ may seem quite small relative to the propagation delay. However, we consider it to account for low jitter services. With a propagation

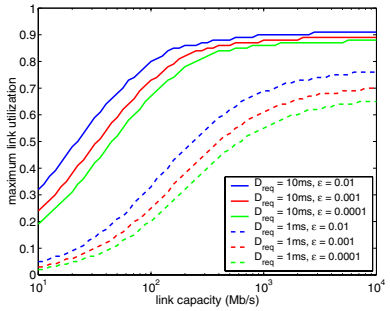

Fig. 8. Maximum achievable link utilization

delay of 20 ms and $D_{req} = 1ms$, a provider could offer a service with an end-to-end delay of 20 ms and 1 ms of jitter.

From Fig. 8 we see that the maximum link utilization reaches a plateau around 1 Gb/s. For links greater than 1 Gb/s, the typical bandwidth found in most backbone networks, link utilization can reach 80% to 90% for all but the most stringent delay guarantees.

III. End-to-end delay

In the previous section we developed a method to compute the delay distribution for a single queue. Now we address how to compute the end-to-end queuing delay through a network. Consider the sample network shown in Fig. 9. The network consists of nine POPs and six customers attached to different POPs[4]. Each customer transmits data to all other customers.

To model the traffic demand between two POPs we use a two-scale FBM process. In the case of our simulation, each traffic demand represents the traffic between a pair of customers while in the actual network each traffic demand would be the aggregate of traffic from many customers. However, as seen in the previous section, increasing the traffic volume will not change the fundamental Gaussian characteristics of the traffic demand. We can therefore approximate the full inter-POP traffic demands by the single customer demands used in this section.

For this network, we would like to compute the end-to-end delay between any of the six edge POPs. To do this, we use the following procedure. For each link in the network we determine which traffic demands arrive at the link and compute the delay experienced on that link. The end-to-end delay over a path in the network is found by convolving the delay distributions for every link along the end-to-end path.

In order to use this procedure we must make two assumptions. First, we assume that the characteristics of a traffic demand remain the same throughout the network. While it is not possible to do this in all networks, in many practical situations it is reasonable to consider the characteristics of a flow are unchanged throughout the network. The most well known example of this behavior is *Kleinrock's independence approximation* [11]. For a queue fed by multiple Poisson input streams, the output of the queue is also Poisson. Similar results have been derived for traffic such as FBM which exhibits a so-called *Large Deviations Principle* [21]. Similar arguments have also been used to justify that the effective bandwidth of a flow remains unchanged throughout the network [6]. The basic idea behind these arguments is that as long as a queue has sufficient output capacity so that very little queueing occurs, the flows passing through the queue will not be affected by the queueing. For the delay requirements we consider, which specify that only a small percentage of the traffic experiences long delay, this is exactly the behavior that will occur.

Second, we assume that the delays at each queue are independent and the end-to-end delay can therefore be obtained by convolution. Since each queue server traffic from many streams and it is unlikely the streams will be correlated, this assumption should be reasonable. Both of these assumptions have been validated in detail in [10]. In this section we present results of a simulation of the entire network to validate the proposed scheme to compute end-to-end delays.

[4]Most commercial backbone networks would have many customers connected to each POP. This sample network, however, provides sufficient complexity to evaluate the proposed delay model.

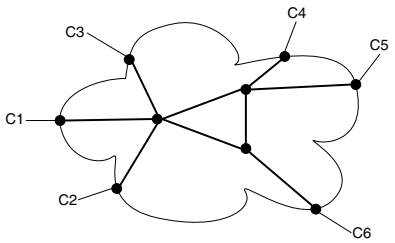

Fig. 9. Sample Network

Customer	Destination Addresses
C1	0.0.0.0-127.255.255.255
C2	128.0.0.0-191.255.255.255
C3	192.0.0.0-199.255.255.255
C4	200.0.0.0-207.255.255.255
C5	208.0.0.0-215.255.255.255
C6	216.0.0.0-223.255.255.255

note: addresses 224.0.0.0 - 255.255.255.255 are the multicast and reserved address range. We do not observe any packets with these destination addresses.

TABLE II

Mapping between customer and destination IP address

A. End-to-end delay validation

To validate this approach of computing the end-to-end delay, we perform a simulation of the network shown in Fig. 9. We use the measurements T4 - T9 to represent the traffic generated by customers C1 - C6 respectively. To generate traffic demand between each customer, we subdivide each trace into six separate sub-traces according to the destination IP address of the packets in the trace. The mapping between destination IP address and customer is shown in Table II.

We find that all but three of the sub-traces have sufficient aggregation to be modeled using two-scale FBM. However, these three traces have an average rate of 0.81, 1.51, and 1.94

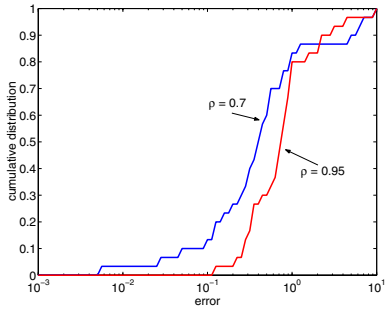

Fig. 10. Difference between end-to-end delay predicted by the model and actual end-to-end delay

Mb/s. Since these flows are so small, we find there is little difference between the end-to-end delay computed when we model these flows as two-scale FBM and when we completely ignore these flows in the computation. We therefore do not consider them in the computation. We do, however, include these flows in the simulation results.

As in the previous section we perform two simulations, one with $\rho = 0.7$ for all links and one with $\rho = 0.95$[5]. For each pair of POPs we compute the percentage difference between the 99.9th percentile of delay distribution computed using the model and the 99.9th percentile of the delay distribution obtained in the simulation. We plot the cumulative distribution of the difference for all POP pairs in Fig. 10. From this figure we see that between 80% of the POP pairs, the difference in the model delay and the simulation delay is less than 1. These results are very similar to the error in the single hop delay shown in Fig. 6. The most noticeable difference is that the error at $\rho = 0.95$ in the end-to-end delay case is quite a bit higher than the error for $\rho = 0.9$ in the single queue case. Recall at high utilization, the difference between the model and the simulation are diverging. As a result, the error at $\rho = 0.95$ will be higher than at $\rho = 0.9$. Since the error in the end-to-end delay is comparable to the error for the single queue seen in the previous sections we conclude that our procedure provides an accurate estimate of the actual end-to-end delay.

IV. Network Optimization

With a method to compute the end-to-end delay over a path in the network, we can now develop a procedure to determine the amount of bandwidth required on each link in a network to meet a particular delay constraint without the use of traffic differentiation. This can be formalized as the following network optimization problem:

Given: a network with fixed topology, fixed routing, and a known traffic demand matrix

Minimize: the total network cost $M = \sum_{l \in L} C_l$

Subject to: $P[d^{(i,j)} > D_{req}^{(i,j)}] < \epsilon^{(i,j)}, \forall i \in N, , j \in N, i \neq j$

[5]We use $\rho = 0.95$ rather than $\rho = 0.9$ as done for the single queue case because for some links in the network $\rho = 0.9$ is still below the knee of the performance curve as a result of the larger traffic volumes used in this simulation.

where:

- L is the set of links in the network
- N is the set of nodes in the network
- C_l is the capacity of link l
- $(m^{(i,j)}, H_1^{(i,j)}, a_1^{(i,j)}, H_0^{(i,j)}, a_0^{(i,j)})$ are the parameters of the traffic demand between POPs i and j

For this problem, we consider the total cost of the network to be the sum of the individual link capacities, but the algorithm may be easily extended to handle more complex cost functions.

The procedure begins by selecting an initial capacity for each link using the following approach. We know that the queuing delay on a single link cannot exceed the total end-to-end delay allowed along the entire path. For a particular link, we can compute the minimum amount of bandwidth needed to satisfy this requirement. This process is repeated for every link, and the end-to-end delay along every path is computed. If the end-to-end delay constraints are satisfied, then we have found the capacity assignment which has the minimum network cost.

In most situations, this procedure finds the minimum cost network. The reason for this is that the end-to-end delays distributions are computed by convolving the delay distributions at each hop (rather than summing the delays as is done to compute the average end-to-end delay). If the delay requirements are satisfied for each queue independently, then it is likely the end-to-end delay is satisfied.

However, this procedure is not guaranteed to work for all possible networks. In the cases where this procedure does not work, more complex heuristics must be used to search for the minimum capacity assignment. We chose to implement the simulated annealing heuristic which has been shown to perform well when applied to the Capacity Assignment problem [12].

A. Capacity Assignment for the Sprint network

In this section we evaluate the feasibility of the bandwidth provisioning approach in the Sprint IP network. To do this, we use the procedure described above to find the amount of bandwidth needed to support a range of end-to-end delay requirements. The inputs to this procedure are the network topology, routing, and traffic demand matrix. The topology and routing of the Sprint network are known, but we do not have measurements of the actual traffic matrix. To generate the traffic demand matrix, we use the approach outlined in [3]. We randomly classify 20% of the POPs as "big," 40% as "medium," and 40% as "small". The mean traffic volume between POPs i and j is selected from a Gaussian distribution with $mean^{(i,j)} = (size_i + size_j)/2$ where $size_{big} = 2.48$ Gb/s, $size_{medium} = 622$ Mb/s, and $size_{small} = 155$ Mb/s. The remaining four model parameters $(a_1^{(ij)}, H_1^{(ij)}, a_0^{(ij)}, H_0^{(ij)})$ are determined based on the mean arrival rate as described in Section II. We consider the "most variable" traffic model.

Since we do not know which specific POPs are "big", "medium", and "small", we generate five random node classifications and the resulting traffic matrices. For each of these scenarios we compute the capacity assignment using the procedure described above. The set of possible link capacities is 155

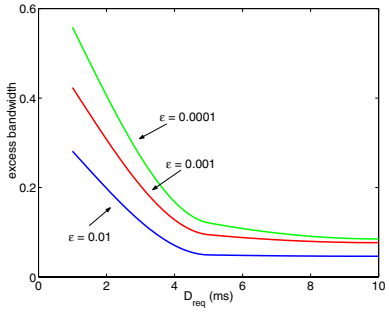

Fig. 11. Excess bandwidth required to meet delay guarantees

Mb/s, 310 Mb/s, 622 Mb/s, 1.24 Gb/s, 2.48 Gb/s, 4.98 Gb/s, 9.95 Gb/s, and 19.9 Gb/s.

Once we have found the bandwidth needed in the network to support the different delay guarantees, we can evaluate the feasibility of the bandwidth provisioning approach. If the network were to be designed without any delay guarantees, it would need enough bandwidth just to support the average data rate of the traffic. It is not possible to reduce the bandwidth in the network beyond this point. To evaluate the bandwidth provisioning approach we compare this minimum bandwidth with the bandwidth found using the capacity assignment algorithm for different delay guarantees.

We define the *excess bandwidth* as the percentage difference between the average rate of the traffic and the link capacity that is found in the solution to the capacity assignment problem: $bw_e = \sum_{l \in L} \frac{c_l - average\ traffic\ volume_l}{average\ traffic\ volume_l}$. For each of the five traffic matrices we compute the excess bandwidth, and the average is plotted in Fig. 11. This figure shows that for the "most variable" traffic queuing delays can be reduced to 4 ms before requiring large amounts of excess bandwidth. Even voice traffic, which is one of the most stringent latency-sensitive applications, does not require such low delays. As a result it appears bandwidth provisioning is an attractive option.

V. Conclusion

This paper developed an approach to compute the amount of bandwidth required on each link in a network so that an end-to-end delay constraint is satisfied. We first analyzed a set of traffic measurements from a commercial IP backbone network and found that straightforward Gaussian processes are an accurate model for aggregate network traffic. Using this model to represent the traffic flow between POPs in the network, we developed a method to compute the end-to-end delay along any path in the network. We then developed a procedure to find the network with the minimum total bandwidth which satisfies the delay constraint.

When applying our approach to a real network we found several interesting results. First, for links with capacity greater than 1 Gb/s, utilization can reach 80%-90% and still meet nearly all delay requirements. For the complete Sprint IP network, we find that between 5% - 15% excess bandwidth is needed to support end-to-end delay requirements as low as 4 ms. Implementing traffic differentiation would only allow the network provider to further decrease the delay by at most 4 ms. Across the entire Internet, this would represent the difference between 124 ms end-to-end delay and 120 ms end-to-end delay. Even stringent applications such as voice would not notice a significant improvement in quality for such a marginal improvement in the total delay.

References

[1] J.-M. Bardet and P. Bertrand, "Detecting abrupt change on the Hurst parameter of a multi-scale fractional Brownian motion with applications," in *New Directions in Time Series Analysis Workshop*, Luminy, France, Apr. 2001.

[2] A. Benassi and S. Deguy, "Multi-scale fractional Brownian motion: definition and identification," Tech. Rep. 83, LLAIC (Laboratoire de Logique, Algorithmique et Informatique de Clermont 1), France, Sep. 1999.

[3] S. Bhattacharyya, C. Diot, J. Jetcheva, and N. Taft, "Pop-level and access-link-level traffic dynamics in a Tier-1 POP," in *Proc. ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, Nov. 2001.

[4] I.M. Chakravarti, R.G. Laha, and J. Roy, *Handbook of Methods of Applied Statistics, Volume I*, John Wiley and Sons, 1967.

[5] M.E. Crovella and M.S. Taqqu, "Estimating the heavy tailed index from scaling properties," *Methodology and Computing in Applied Probability*, vol. 1, no. 1, pp. 55–79, Jan. 1999.

[6] G. de Veciana, G. Kesidis, and J. Walrand, "Resource management in wide-area ATM networks using effective bandwidths," *IEEE J. on Selected Areas in Comm.*, vol. 13, no. 6, pp. 1081–1090, Aug. 1995.

[7] N.G. Duffield and N. O'Connell, "Large deviations and overflow probabilities for the general single-server queue, with applications," in *Proc. Cambridge Phil. Soc.*, 1995, vol. 118, pp. 363–374.

[8] A. Erramilli, O. Narayan, A.L. Neidhardt, and I. Saniee, "Performance impacts of multi-scaling in wide area TCP/IP traffic," in *Proc. IEEE INFOCOM 2000*, Tel-Aviv, Israel, Mar. 2000, pp. 352–359.

[9] A. Feldmann, A.C. Gilbert, P. Huang, and W. Willinger, "Dynamics of IP traffic: A study of the role of variability and the impact of control," in *Proc. ACM SIGCOMM '99*, Cambridge, Massachusetts, Aug 1999, pp. 301–313.

[10] C. Fraleigh, *Provisioning Internet Backbone Networks to Support Latency Sensitive Applications*, Ph.D. thesis, Stanford University, June 2002.

[11] L. Kleinrock, *Queueing Systems Volume 2: Computer Applications*, John Wiley and Sons, 1976.

[12] A. Levi and C. Ersoy, "Discrete link capacity assignment in prioritized computer network: Two approaches," in *Proc. Ninth Int'l Symposium on Computer and Information Networks*, Antalya, Nov. 1994, pp. 408–415.

[13] I. Norros, "On the use of Fractional Brownian Motion in the theory of connectionless networks," *IEEE J. on Selected Areas in Comm.*, vol. 13, no. 6, pp. 953–962, Aug 1995.

[14] D. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, F. Tobagi, and C. Diot, "Analysis of measured single-hop delay from an operational backbone network," in *Proc. IEEE INFOCOM 2002*, New York, New York, June 2002.

[15] D. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, "Long-Term Forecasting of Internet Backbone Traffic: Observations and Initial Models," in *Proc. IEEE INFOCOM 2003*, San Francisco, California, Mar. 2003.

[16] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling," *IEEE/ACM Trans. on Networking*, vol. 3, no. 3, pp. 226–244, June 1995.

[17] V.J. Ribeiro, R.H. Riedi, M.S. Crouse, and R.G. Baraniuk, "Multiscale queuing analysis of long-range dependent network traffic," in *Proc. IEEE INFOCOM 2000*, Tel-Aviv, Israel, Mar. 2000, pp. 1026–1035.

[18] G. Samorodnitsky and M.S. Taqqu, *Stable non-Gaussian Random Processes*, Chapman & Hall, 1994.

[19] D. Veitch and P. Abry, "A wavelet based joint estimator for the parameters of long-range dependence," *IEEE Trans. on Info. Theory*, vol. 45, no. 3, pp. 878–897, Apr. 1999.

[20] W. Willinger, M.S. Taqqu, R. Sherman, and D.V. Wilson, "Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level," *IEEE/ACM Trans. on Networking*, vol. 5, no. 1, pp. 71–86, Feb. 1997.

[21] D. Wischik, "The output of a switch, or, effective bandwidths for networks," *Queueing Systems*, vol. 32, pp. 383–396, 1999.