End-to-end Estimation of the Available Bandwidth Variation Range

Manish Jain
Georgia Tech
jain@cc.gatech.edu

Constantinos Dovrolis
Georgia Tech
dovrolis@cc.gatech.edu

***Abstract*— The available bandwidth (avail-bw) of a network path is an important performance metric and its end-to-end estimation has recently received significant attention. Previous work focused on the estimation of the average avail-bw, ignoring the significant variability of this metric in different time scales. In this paper, we show how to estimate a given percentile of the avail-bw distribution at a user-specified time scale. If two estimated percentiles cover the bulk of the distribution (say 10% to 90%), the user can obtain a practical estimate for the avail-bw variation range. We present two estimation techniques. The first is iterative and non-parametric, meaning that it is more appropriate for very short time scales (typically less than 100ms), or in bottlenecks with limited flow multiplexing (where the avail-bw distribution may be non-Gaussian). The second technique is parametric, because it assumes that the avail-bw follows the Gaussian distribution, and it can produce an estimate faster because it is not iterative. The two techniques have been implemented in a measurement tool called Pathvar. Pathvar can track the avail-bw variation range within 10-20%, even under non-stationary conditions. We identify four factors that play a crucial role in the variation range of the avail-bw: traffic load, number of competing flows, rate of competing flows, and of course the measurement time scale. Finally, we present a new way to detect whether a probing rate is larger than the avail-bw, without relying on the fluid traffic assumption or on static thresholds.**

I. Introduction

Recently, the area of end-to-end available bandwidth (avail-bw) estimation has attracted considerable interest. The avail-bw is an important metric for several applications, such as socket buffer sizing, overlay routing, p2p file transfers, server selection, and interdomain path monitoring. As a result, several estimation techniques and tools based on active measurements have been developed, including Delphi [1], TOPP [2], Pathload [3], IGI/PTR [4], Pathchirp [5], Spruce [6], and Bfind [7]. All previous work aimed to estimate the *average avail-bw*, largely ignoring that the avail-bw is a time-varying quantity, defined as an average over a certain *measurement time scale*. If we view the avail-bw as a stationary random process, the second-order statistics, namely the variance of the marginal distribution and the autocorrelation function, are needed for a more complete characterization of the avail-bw process. In this work, we focus on the end-to-end estimation of the variability of the avail-bw marginal distribution, leaving the identification of the correlation structure for future work.

This work was supported by the DOE Office of Science (award DE-FC02-01ER25467), by NSF (award 0230841), and by an equipment donation from Intel.

The avail-bw, especially in sub-second scales, can exhibit significant variations around its time average, making the latter a rather poor-quality estimator or predictor. To illustrate this point, Figure 1 shows the avail-bw time series, and the corresponding marginal distribution, for two measurement time scales: 20msec and 1sec. This time series was obtained from a packet trace collected at an OC-3 link, and it thus represents an exact (rather than estimated) sample path of the avail-bw process in that link. Notice that the 10%-90% variation range of the distribution in the 20msec scale is approximately 30Mbps to 75Mbps, while the average avail-bw is 52Mbps. We anticipate that information for the variation range of the avail-bw distribution will actually be more important for some applications than an estimate of the mean. For example, a video streaming application with a nominal transmission rate of 3Mbps may prefer to use a path with average avail-bw 5Mbps and a very narrow variation range, rather than a path with average avail-bw 10Mbps but a variation range of 1Mbps to 20Mbps. Also, the measurement time scale is an application-specific parameter, and it represents the minimum time interval in which the avail-bw variations matter for a particular application.

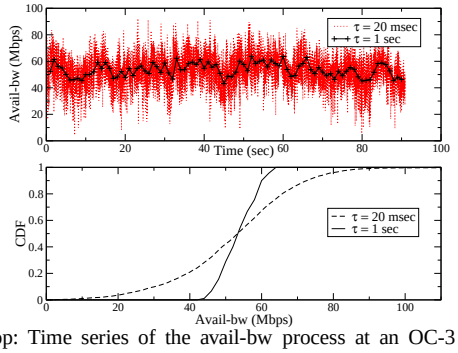

Fig. 1. Top: Time series of the avail-bw process at an OC-3 link in two measurement time scales. Bottom: Empirical CDFs of the two time series.

A. Definitions

We first define the avail-bw at a network link and then at an end-to-end path. Suppose that a network path consists of H store-and-forward first-come first-served links. A link i has an instantaneous utilization $u_i(t)$ at time t; $u_i(t)$=0 if the link is idle and $u_i(t)$=1 if the link transmits a packet at time t. The average utilization $u_i(t, t+\tau)$ of link i during the time interval $(t, t+\tau)$ is

$$u_i(t, t+\tau) = \frac{1}{\tau} \int_t^{t+\tau} u_i(t)\, dt \qquad (1)$$

We refer to τ as the *measurement time scale*.

The available bandwidth $A_i(t, t+\tau)$ of link i during the time interval $(t, t+\tau)$ is defined as the average residual capacity in that interval,

$$A_i(t, t+\tau) = C_i[1 - u_i(t, t+\tau)] \quad (2)$$

Consider now a network path that traverses a sequence of H links. The end-to-end available bandwidth $A(t, t+\tau)$ of the network path during $(t, t+\tau)$ is defined as the minimum avail-bw among the H links in the same interval,

$$A(t, t+\tau) = \min_{i=1...H} A_i(t, t+\tau) \quad (3)$$

We refer to the link with the minimum avail-bw as *tight link* and denote its capacity by C_t. The link with the minimum capacity is referred to as *narrow link* and has a capacity C_n. Note that, in general, the narrow and tight links can be different. Also, all existing avail-bw estimation techniques assume that the tight link has much lower avail-bw than the other links. Otherwise, the path avail-bw may be limited by more than one links. Furthermore, if there are multiple bottlenecks in a path then all existing avail-bw estimation techniques suffer from underestimation errors [8]. In this paper, we adopt the assumption that the path has a clearly distinguishable tight link, meaning that the avail-bw in all other links is significantly larger.

The average end-to-end avail-bw $A(t, t+\tau)$ is a function of time t and therefore it can be viewed as a random process $A_\tau(t)$, where τ is the measurement time scale. If we assume that this process is stationary and identically distributed along the time axis, then at any time instant t the process is described by the same random variable A_τ. Let $F_\tau(a)$ be the cumulative distribution function of A_τ, where $F_\tau(a) = P(A_\tau \leq a)$. The p-th percentile of the avail-bw random variable A_τ, with $p \in (0, 1)$, is the value A_τ^p such that $F_\tau(A_\tau^p) = p$; in the rest of the paper we assume that A_τ^p is unique.

Our main objective in this paper is to estimate the variability of A_τ. One possibility could be to estimate the variance of A_τ. That would be the obvious variability metric if we knew that the avail-bw distribution is symmetric around the mean and close to Gaussian. A more general metric, however, is a percentile-based definition of variability. Specifically, if p^L is a low probability and p^H is a high probability, then we can define the variation range of A_τ as the interval $[A_\tau^L, A_\tau^H]$, where A_τ^L and A_τ^H are the p^L and p^H percentiles of A_τ, respectively. In the rest of this paper, unless noted otherwise, we assume that the user is interested in the 10%-90% variation range, i.e., p^L=0.1 and p^H=0.9. Of course the actual definition of the variation range would be application-specific.

Some further discussion on the relation between the measurement time scale τ and the variability of the avail-bw process is important. The mean of A_τ does not depend on τ. The variance σ_τ^2=Var$[A_\tau]$, however, depends strongly on τ and on the correlation structure of the random process $A_\tau(t)$. In general, as τ increases, the variance σ_τ^2 decreases. The speed with which the variance decreases, however, depends on the correlation structure of the underlying process. For instance, the variance of a self-similar process decreases much more slowly with τ than the variance of an IID process [9]. We return to this point in §VII-C.

B. Related Work

As previously mentioned, the existing avail-bw measurement techniques aim to estimate the average avail-bw. These techniques have been classified in two categories [8]. First, in *direct probing* techniques, each probing packet stream results in a sample of the avail-bw process. Assuming that the probing rate R_i is larger than the avail-bw A during the probing stream, the obtained avail-bw sample is given by

$$A = C_t - R_i(\frac{C_t}{R_o} - 1)$$

where R_o is the output rate of the probing stream and C_t is the capacity of the tight link. The key point about *direct probing* schemes is that it directly samples the avail-bw process, as long as C_t is known and $R_i > A$. Delphi [1], IGI [4] and Spruce [6] are based on this approach. Direct probing assumes a fluid traffic model. Furthermore, direct probing assumes that the tight link is the same with the narrow link, and thus the capacity C_t=C_n can be estimated with standard packet-pair capacity estimation techniques [10]. Because of the limitations of the previous two assumptions, we do *not* use direct probing in this paper.

The second estimation approach is referred to as *iterative probing*. It includes TOPP [2], Pathload [3], Pathchirp [5], PTR [4], and Bfind [7]. In iterative probing, each probing stream is used to examine whether the stream's rate is larger than the avail-bw during the probing interval. The key idea is that if the output rate of a probing stream is smaller than the input rate, or if the one-way delays of consecutive packets in the stream show increasing trend, then the probing rate is larger than the avail-bw during the probing stream. An important difference with direct probing is that iterative probing does not require knowledge of the tight link capacity. The probing rate is varied either linearly or based on what happened in previous streams, until the probing process converges to an estimate of the average avail-bw. An exception to the previous description is Pathload [3]. Pathload was the first tool to consider the variability of avail-bw process and to report a variation range (called "grey region") rather than a point estimate. However, Pathload does not specify the percentiles that correspond to the grey region, and it does not allow the user to control the desired percentiles or the measurement time scale.

A related area in the literature is that of traffic modeling and analysis, and in particular, the measurement of the second-order statistics (variance and autocorrelation) in various time scales using packet traces. That area, which started with the seminal Bellcore work [11], revealed that the traffic count process at a network link is asymptotically self-similar. The reader is referred to [9] for a survey of the related literature. Our approach and objectives in this work are significantly different. First, instead of passive traffic measurements at a single link we are interested in the active estimation of the avail-bw in an end-to-end path. Second, instead of focusing on the scaling properties of the avail-bw process, we focus on the variability of the marginal distribution at a given time scale.

Third, our high-level goal is to develop tools that can be used in practice to measure important path characteristics, rather than to statistically characterize or model network traffic.

C. Main Contributions and Overview

In this paper, we first present a measurement technique, referred to as *percentile sampling*, that can associate a given probing rate with a percentile of the avail-bw distribution. We then use percentile sampling to design two estimation algorithms for the avail-bw variation range.

The first algorithm is iterative in nature. We refer to it as *non-parametric*, because it does not assume a specific avail-bw distribution. The non-parametric algorithm is more appropriate for very short values of the measurement time scale (typically less than 100msec) or in bottlenecks with limited flow multiplexing, where the avail-bw distribution may be non-Gaussian.

The second algorithm is parametric, because it assumes that the avail-bw follows the Gaussian distribution. This assumption is typically valid when $\tau >$100-200msec and when the tight link carries a significant amount of aggregated traffic [12]. The parametric algorithm can produce an estimate faster than the non-parametric algorithm because it is not iterative.

The two estimation algorithms have been implemented in a measurement tool called Pathvar. We have validated Pathvar with simulations and testbed experiments using realistic Internet traffic. Pathvar can track the actual avail-bw variation range within 10-20%, even under non-stationary conditions.

Pathvar also uses a novel mechanism to detect whether a probing rate is larger than the avail-bw. This is a central problem in avail-bw estimation. The proposed mechanism does not rely on the fluid traffic assumption or on static thresholds, which are limitations of previous work.

Finally, we focus on four factors that can significantly affect the variation range of the avail-bw. These factors are the traffic load, number of competing flows, rate of competing flows, as well as the measurement time scale. The results of that study explain why the avail-bw appears as less or more variable depending on the load conditions and the degree of statistical multiplexing at the tight link.

The rest of the paper is structured as follows. The percentile sampling technique is described in §II. §III presents the non-parametric estimation algorithm, while §IV presents the parametric algorithm. §V describes how to determine whether a probing rate is larger than the avail-bw. The implementation of Pathvar, and a few typical validation results, are summarized in §VI. Finally, we examine the four factors that affect the variability of the avail-bw process in §VII. We conclude in §VIII.

II. Percentile sampling

In this section, we first describe the basic technique of percentile sampling, which forms the basis of the proposed estimation algorithms in the next two sections. A number N of probing streams of duration τ and rate R are sent to a path. Each stream provides an indication of whether the avail-bw in the corresponding time interval is higher than R. The resulting N binary samples are used to estimate the percentile of the avail-bw distribution that corresponds to rate R. We also derive the required number of samples N for a given maximum error, assuming independent sampling.

A. Basic idea

Consider a network path. The avail-bw random process measured in time scale τ is $A_\tau(t)$. As mentioned in the Introduction, we assume that this process is stationary and identically distributed. Given the previous assumptions, we can focus on the random variable A_τ and on its time-invariant marginal distribution F_τ.

The sender transmits a probing packet stream of rate R and duration τ during $(t, t+\tau)$ to the receiver. If M is the packet size, then the interarrival between successive packets is M/R and the number of probing packets is $\lceil \frac{\tau R}{M} \rceil$. The avail-bw during $(t, t+\tau)$ is given by a realization of the random variable A_τ. The receiver classifies the stream as *type-G* if it infers that the probing rate R is *greater* (or equal) than A_τ. Otherwise, the stream is classified as *type-L* (for "lower"). The classification of a stream as type-G or type-L is the subject of §V; for now we just note that this classification can be performed based on statistical analysis of the one-way delays of the stream's probing packets.

We use the indicator variable $I(R)$ to represent whether a stream is of type-G ($I(R) = 1$) or type-L ($I(R) = 0$). If $F_\tau(a)$ is the Cumulative Distribution Function (CDF) of A_τ, we have that

$$I(R) = \begin{cases} 1 \text{ with probability } F_\tau(R) \\ 0 \text{ with probability } 1 - F_\tau(R) \end{cases}$$

So, the expected value of $I(R)$ is $E[I(R)] = F_\tau(R)$.

A single probing stream can only tell us if the probing rate R is greater than the realization of the avail-bw random variable in the corresponding time interval. To accumulate N such samples, the sender transmits N identical probing streams[1]. The indicator variable for each stream is denoted by $I_i(R)$. Because different streams will sample different realizations of A_τ, some streams may be classified as type-G and others as type-L. Let $I(R, N)$ be the number of streams of type-G, i.e., $I(R, N) = \sum_{i=1}^{N} I_i(R)$. The expected value of $I(R, N)$ is

$$\begin{aligned} E[I(R,N)] &= \sum_{i=1}^{N} E[I_i(R)] \\ &= F_\tau(R)N \end{aligned} \quad (4)$$

The following proposition summarizes the basic idea of percentile sampling:

Proposition 1: For a stationary avail-bw process, the fraction $I(R, N)/N$ of type-G probing streams of rate R is an unbiased estimator of $p = F_\tau(R)$.

Proposition 1 provides us with a mapping from a given probing rate R to the corresponding cumulative probability in the avail-bw distribution. Since our goal is to estimate a given percentile of the avail-bw distribution, we are interested

[1]The time period between streams should be sufficiently long for the streams to not get queued behind each other while in transit.

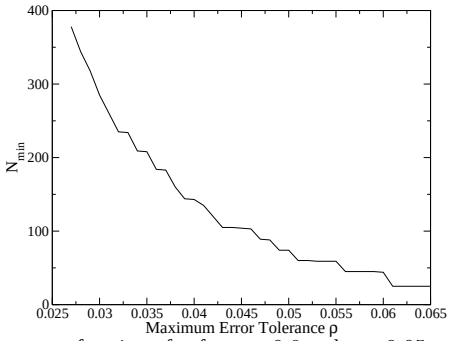

Fig. 2. N_{min} as a function of ρ for $p = 0.9$ and $\epsilon = 0.05$.

in the inverse mapping, from a certain probability to the corresponding probing rate. We present two algorithms that perform this inverse mapping in §III and §IV.

It is important to note that Equation (4) does not require the statistical independence of the N avail-bw samples. Therefore, Proposition 1 can be used even without information about the (generally complex and unknown) correlation structure in the process $A_\tau(t)$.

B. How large should N be?

Proposition 1 refers to the expected value of $I(R, N)$. The obvious question is how large should the sample size N be so that the fraction $I(R, N)/N$ is a good approximation of $F_\tau(R)$? In this section, we derive the minimum value of N that is required for a given error tolerance.

Suppose that we aim to estimate the p-th percentile of A_τ, denoted by A_τ^p. Let ρ be the maximum allowed percentile error in the estimation of A_τ^p. This means that probing rate R would be an acceptable estimate of A_τ^p if the corresponding fraction $I(R, N)/N$ is between $p - \rho$ and $p + \rho$. So, a rate R will be correctly mapped to the p-percentile as long as

$$Prob[N(p - \rho) \leq I(R, N) \leq N(p + \rho)] > 1 - \epsilon \qquad (5)$$

where ϵ is a small mis-classification probability.

To derive the previous probability, we need to make the additional assumption that the binary outcomes $I_i(R)$ of the N probing streams are independent. If that is the case, then $I(R, N)$ follows the binomial distribution with a success probability of $F_\tau(R)$. So, the probability that i out of N streams are of type-G is given by

$$P[I(R, N) = i] = \binom{N}{i} F_\tau(R)^i [1 - F_\tau(R)]^{N-i} \qquad (6)$$

From Equations (5) and (6), we can calculate the minimum value N_{min} of streams required for a given error tolerance ρ and a given mis-classification probability ϵ. The probability $F_\tau(R)$ is determined based on the percentile that we aim to estimate. For instance, if we are interested in the 90% percentile then $F_\tau(R)$=0.9. Figure 2 shows N_{min} as a function of ρ for ϵ=0.05 and for estimating the 90% percentile. As expected, N increases quickly as we decrease the error threshold ρ. Specifically, as ρ becomes less than 4%, we need more than 100 samples (or probing streams).

In practice, generating a large number of probing streams increases the measurement overhead and it slows down the estimation process. Our objective in this work is to design a measurement tool that can track the avail-bw variation range in real time, even if the latter changes with time. For this reason, we prefer to use a relatively small number of probing streams, even if the resulting error tolerance ρ is significant. Specifically, in the rest of the paper we typically use $N = 40 - 50$ streams, limiting the maximum percentile error ρ to about 0.05-0.06.

III. Non-parametric estimation

In this section, we present a simple iterative algorithm for the estimation of the variation range $[A_\tau^L, A_\tau^H]$ in a given time scale τ. We refer to the following algorithm as *non-parametric*, in the sense that it does not assume a specific marginal distribution for the underlying avail-bw, or, equivalently, for the traffic at the tight link.

A. Algorithm

Suppose that we want to first estimate the higher bound A_τ^H of the variation range. If A_τ^H is the p-th percentile, then p=$F_\tau(A_\tau^H)$. The basic idea in the following algorithm is to iteratively adjust the probing rate R so that, based on Proposition 1, the fraction of probing streams that are of type-G is approximately p.

Specifically, in the n-th iteration of the algorithm the sender transmits N streams of rate R_n to the receiver. The receiver classifies each stream as type-G or type-L, and calculates the fraction $f_n = I(R_n, N)/N$ of streams that are of type-G. Based on Proposition 1, the expected value of f_n is equal to $F_\tau(R_n)$. So, if the rate R_n is close to the target percentile A_τ^H, we expect that f_n would be approximately equal to p. Similarly, if R_n is larger than A_τ^H then f_n is expected to be higher than p, while if R_n is less than A_τ^H then f_n is expected to be lower than p. The information about f_n is delivered back to the sender, which then sets the probing rate R_{n+1} accordingly.

In more detail, if f_n is within $p \pm \rho$, where ρ is a maximum percentile error, the rate R_n is reported as an estimate of the p-th percentile and the probing rate remains the same, i.e., $R_{n+1} = R_n$. If $f_n > p + \rho$, the sender needs to reduce the probing rate. Similarly, if $f_n < p - \rho$, the sender needs to increase the probing rate. The probing rate ratio R_{n+1}/R_n in the next iteration is based on the difference $f_n - p$. This is just a heuristic, but it is reasonable given that we do not have additional information about the shape of the underlying avail-bw distribution.

To avoid strong oscillations, we impose an upper bound on the rate variation between two successive iterations through a parameter b. A larger value of b allows faster convergence, especially under non-stationary conditions, but it also increases the estimation error. As will be shown later, a value of b around 0.10-0.20 is a good trade-off between accuracy and responsiveness, at least based on our validation experiments.

Algorithm III.1 shows the pseudo-code of the non-parametric algorithm. The input parameters are the number of streams N, the probability p that corresponds to the desired percentile, and the error tolerance ρ. To measure the variation range $[A_\tau^L, A_\tau^H]$, the algorithm is executed twice in each

iteration: N streams with probing rate R^H to estimate A_τ^H ($p = p^H$) and another set of N streams with rate R^L to estimate A_τ^L ($p = p^L$). The two sets of streams can be interleaved so that the reported estimates of the variation range cover the same time interval.

The non-parametric algorithm is iterative, and so it will be unable to track the avail-bw variation range if the latter does not remain roughly constant during at least a few iterations. The total probing duration for each iteration of the previous algorithm is $2N(\tau+T_{idle})$, where T_{idle} is the idle time which may be introduced between successive streams to reduce intrusiveness. For N=50, τ=20msec, and T_{idle}=80msec, two successive iterations of the previous algorithm will sample the same avail-bw distribution as a long as the underlying avail-bw process remains stationary for at least 10 seconds.

Algorithm III.1: NON-PARAMETRIC(N, p, ρ)

repeat
- Send N streams of duration τ at rate R_n
- $I(R_n, N) \leftarrow 0$
- **for** $i \leftarrow 1$ **to** N
  - **do** **if** $stream[i] =$ *type-G*
    - **then** $I(R_n, N) \leftarrow I(R_n, N) + 1$
- $f_n \leftarrow I(R_n, N)/N$
- **if** $f_n > p + \rho$
  - **then** $diff \leftarrow \text{MIN}(b, f_n - p)$; $R_{n+1} \leftarrow R_n * (1 - diff)$
  - **else if** $f_n < p - \rho$
  - **then** $diff \leftarrow \text{MIN}(b, p - f_n)$; $R_{n+1} \leftarrow R_n * (1 + diff)$
  - **else** $R_{n+1} \leftarrow R_n$; *output* R_n

B. Estimation with non-stationary load in single-hop path

In this section, we show examples of how the previous algorithm performs in a single-hop path with non-stationary traffic load that includes level shifts and short spikes. To make sure that the traffic load is realistic, we use packet traces captured by NLANR-MOAT at various OC-3 links (BWY-1063326722-1, COS-1049166362 and BWY-1063304167-1) [13]. Since we know the actual traffic load, we can calculate the exact 10%-90% percentiles of the avail-bw distribution, and so we can validate the previous estimation algorithm. In the following, the measurement time scale τ is 100msec. In the experiments of this section we make sure that the the classification of streams in type-G or type-L is always correct, by comparing the actual avail-bw in each probing interval with the probing rate.

To create non-stationary traffic loads, we merge different NLANR traces. Each trace is 90sec long, while the avail-bw process in each trace is stationary. The non-stationary traffic time series shown in Figure 3 and 4 was composed as follows: trace-1 was "played-back" twice, followed by trace-2 twice, followed from 20sec of trace-3 (to create the spike that occurs at t=360sec), and finally 50sec of trace-2 again.

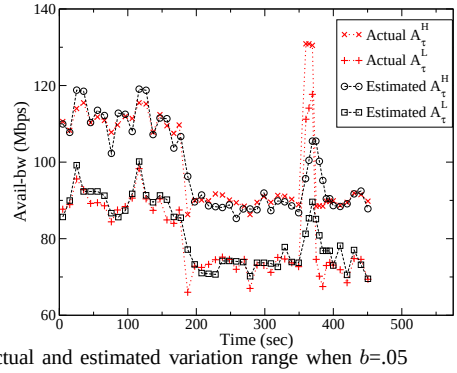

Fig. 3. Actual and estimated variation range when b=.05

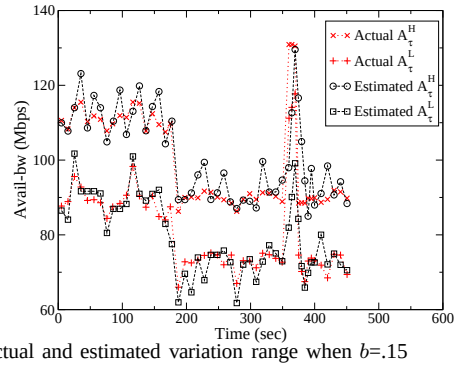

Fig. 4. Actual and estimated variation range when b=.15

The time series of the actual 10%-90% variation range was measured by segmenting the traffic trace in successive intervals of length $2N\tau$. At each segment, we calculate the empirical CDF of the avail-bw measured in time intervals of length τ. So, each successive interval of length $2N\tau$ results in a single measurement of the actual variation range in the time scale τ. The corresponding estimated variation range is inferred from the previous non-parametric algorithm using the same measurement time scale (τ) and measurement period ($2N\tau$).

Figures 3 and 4 show the actual and the estimated 10%-90% variation range for two values of b. Notice that, overall, the estimation algorithm is able to successfully track the avail-bw variation range. During stationary time periods, the estimation error is less than 5%. The estimation errors are larger, however, during level shifts and short spikes.

The accuracy and responsiveness of the algorithm depend on b. The parameter b determines the maximum allowed rate variation in two successive iterations. When the avail-bw process is stationary, and the estimated variation range is already close to the actual variation range, a lower b performs better because it causes lower oscillations around the actual percentiles. For instance, the Root Mean Square Error (RMSE) of the estimated A_τ^H during the first 180 sec of the trace in Figure 3 is 2.4 for b=0.05 and 4.4 for b=0.15. The RMSE values for A_τ^L are 2.1 and 3.6, respectively.

On the other hand, a higher value of b is better during initialization, or when the traffic load exhibits frequent level shifts or spikes. For instance, notice the spike that occurs in

the time interval [360,390] in Figures 3 and 4. Such an intense traffic spike can be due to a route flap or some form of network anomaly. With b=0.05 the estimation algorithm does not track successfully the magnitude of the traffic spike, while with b=0.15 the algorithm is much more responsive. We note that the selection of b should be made based on the nature of the network path that is to be monitored. As it happens with most estimation tools, their accuracy depends on the calibration of certain parameters in the specific environment where these tools are to be used.

IV. Parametric estimation

In this section, we present a parametric estimation algorithm that is based on the assumption that the avail-bw marginal distribution is Gaussian. This is a reasonable assumption for links with a large degree of flow multiplexing (high "vertical aggregation") and for sufficiently long measurement time scales τ (high "horizontal aggregation"). The Gaussian assumption in the context of network traffic and the required degrees of vertical and horizontal aggregation have been examined in [12] and the references therein. Specifically, the measurements presented in [12] show that the vertical aggregation of at least 25 users, with an aggregate average traffic rate of 25Mbps, is a good fit with the Gaussian model in time scales that are longer than 128msec. Also, the Gaussian model is a good approximation when the measurement time scale is longer than 1sec and the aggregate average rate is as small as a few Mbps. When it is not likely that the previous conditions hold, the non-parametric algorithm of the previous section should be used instead.

A. Algorithm

A Gaussian distribution is completely described by its mean and variance. Furthermore, the knowledge of any two percentiles of the Gaussian distribution is sufficient to compute the mean and variance. The basic idea in the following algorithm is to estimate two arbitrary percentiles of the avail-bw distribution based on Proposition 1. Then, we use these two percentiles to estimate the mean and variance of F_τ, and finally we estimate the user-specified variation range $[A_\tau^L, A_\tau^H]$.

In more detail, suppose that the avail-bw distribution has mean μ and variance σ_τ^2 in the time scale τ. Exactly as in the non-parametric algorithm, the sender generates N probing streams of rate R_1 and then it calculates the fraction f_1 of streams that are of type-G. Based on Proposition 1, the expected value of this fraction is equal to the cumulative probability $F_\tau(R_1)$ that corresponds to rate R_1. So, if N is sufficiently large, we expect that $f_1 \approx F_\tau(R_1)$. The previous process is repeated for a different probing rate R_2, resulting in an additional constraint $f_2 \approx F_\tau(R_2)$. With the previous two constraints, we can then calculate the standard deviation and the mean of F_τ as follows:

$$\sigma_\tau = \frac{R_1 - R_2}{\phi^{-1}(f_1) - \phi^{-1}(f_2)} \tag{7}$$

$$\mu = R_1 - \sigma_\tau \phi^{-1}(f_1) \tag{8}$$

where ϕ^{-1} is the inverse of the standard normal distribution CDF. Finally, the percentiles that correspond to the variation range are:

$$A_\tau^H = \mu + \sigma_\tau \phi^{-1}(p^H) \tag{9}$$

$$A_\tau^L = \mu + \sigma_\tau \phi^{-1}(p^L) \tag{10}$$

It is important to note that the probing rates R_1 and R_2 need not be equal to A_τ^H or A_τ^L, respectively. Instead, it is sufficient to choose R_1 and R_2 so that the corresponding percentiles p_1 and p_2 are significantly different, i.e., $|p_1 - p_2| > \rho$. Furthermore, we can choose R_1 and R_2 so that they are at the left half of the Gaussian distribution. Doing so reduces the intrusiveness of the measurements, because the probing streams are of lower rate than the average avail-bw.

In practice, the probing rates R_1 and R_2 can be chosen to track two low percentiles, say the 20% and the 40%. This can be achieved by adjusting the two rates at the end of each repetition of the algorithm, based on the estimated Gaussian distribution. Notice that even with this optimization, the parametric algorithm remains non-iterative because the estimate of the variation range in each repetition of the algorithm does not depend on the estimate in the last repetition.

The pseudo-code for the parametric algorithm is given in Algorithm IV.1. As in the non-parametric algorithm, the transmission of the N streams of rate R_1 can be interleaved with the streams of rate R_2.

Algorithm IV.1: PARAMETRIC(N, p^H, p^L, p_1, p_2)

repeat

Send N streams of duration τ at rate R_1
Send N streams of duration τ at rate R_2

$I(R_1, N) \leftarrow 0$
for $i \leftarrow 1$ **to** N
  do **if** $stream[i, R_1] =$ *type-G*
    then $I(R_1, N) \leftarrow I(R_1, N) + 1$
$f_1 \leftarrow I(R_1, N)/N$

$I(R_2, N) \leftarrow 0$
for $i \leftarrow 1$ **to** N
  do **if** $stream[i, R_2] =$ *type-G*
    then $I(R_2, N) \leftarrow I(R_2, N) + 1$
$f_2 \leftarrow I(R_2, N)/N$

$\sigma_\tau \leftarrow \frac{R_1 - R_2}{\phi^{-1}(f_1) - \phi^{-1}(f_2)}$
$\mu \leftarrow R_1 - \sigma_\tau \phi^{-1}(f_1)$

$A_\tau^H \leftarrow \mu + \sigma_\tau \phi^{-1}(p^H)$
$A_\tau^L \leftarrow \mu + \sigma_\tau \phi^{-1}(p^L)$

$R_1 \leftarrow \mu + \sigma_\tau \phi^{-1}(p_1)$
$R_2 \leftarrow \mu + \sigma_\tau \phi^{-1}(p_2)$

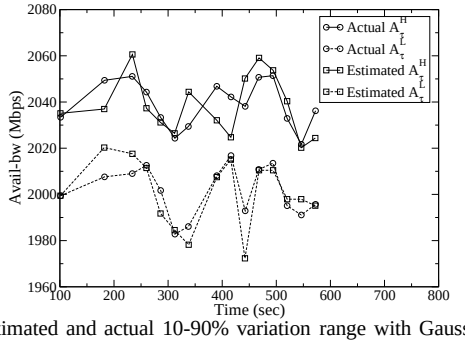

Fig. 5. Estimated and actual 10-90% variation range with Gaussian traffic.

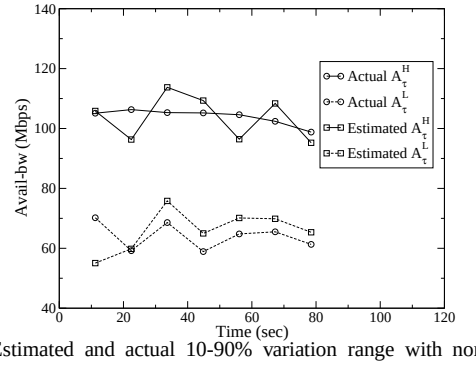

Fig. 6. Estimated and actual 10-90% variation range with non-Gaussian traffic.

B. Validation examples

In this section, we illustrate the accuracy of the parametric algorithm for both Gaussian and non-Gaussian avail-bw distributions. We are again interested in the 10%-90% variation range. The probing rates are chosen based on the 10% and 40% percentiles, i.e., p_1=0.10 and p_2=0.40. The actual variation range is measured by calculating the empirical CDF in successive time windows of legth $2N\tau$, as described in §III-B.

Figure 5 shows the actual and the estimated variation range for an OC-48 NLANR packet trace (IPLS-CLEV-20020814-091000-1). The measurement time scale is τ=250msec. To examine whether the avail-bw distribution is Gaussian in that time scale, we calculate the kurtosis and skewness of the corresponding distribution. A Gaussian random variable has a skewness of zero and a kurtosis of 3. In this trace, the skewnewss and kurtosis are 0.21 and 3.15, respectively, meaning that the avail-bw distribution is reasonably close to Normal even though it is not a perfect match. The main observation in Figure 5 is that the parametric algorithm can closely track the variation range within 5% or better. The RMSE for this trace is 1.1.

On the other hand, Figure 6 shows the actual and estimated variation range for an OC-3 packet trace (ANL-1070464136) that deviates significantly from the Gaussian model. The measurement time scale in this case is τ=50msec, while the skewness and kurtosis are 0.46 and 5.23, respectively. Although the parametric algorithm is still able to track the variation range, there is a non-negligible bias in the estimation of the lower percentile, and the estimation error is significantly larger (RMSE=4.9) compared to the case of Gaussian traffic.

V. Probing rate classification

So far, we have assumed that the receiver can correctly infer whether a probing stream is of type-G or type-L, i.e., whether the stream's input rate R_i is larger than the avail-bw during the probing interval. We remind the reader that R_i is the constant rate with which the sender transmits probing packets to the receiver, while R_o is the average rate with which this packet stream arrives at the receiver. The techniques that have been used in the literature for comparing R_i with $A_\tau(t)$ have some important limitations. Specifically, previous techniques are either based on the oversimplifying fluid traffic assumption, or they use static thresholds that should instead be path and load dependent. Here, we propose a new inference technique that does not have the previous two shortcomings.

The first existing approach to compare a probing rate with the avail-bw was reported in [2]. Assuming that the cross traffic follows the fluid model, i.e., ignoring the burstiness due to discrete packet sizes and random interarrivals, it is easy to show that $R_i > A_\tau(t)$ if and only if $R_o < R_i$. This is true because, when the traffic follows the fluid model, the probing packets are queued in the tight link only when their input rate is sufficiently high to overload that link. This is not the case, however, without the fluid model assumption. In that case, queues build up even before the tight link becomes saturated, causing underestimation of the avail-bw. This issue has been recently studied in [14].

The second approach to compare R_i with $A_\tau(t)$ is based on the time series of OWDs in a probing stream. This approach was first followed in [15]. The basic idea is that if $R_i > A_\tau(t)$ then the OWDs of the probing packets should exhibit an increasing trend. This increasing trend in the delays is due to the queueing build-up at the tight link when its avail-bw is exhausted. This approach does not rely on the fluid model assumption, but its effectiveness strongly depends on the statistical technique and the related parameters that are used to infer the presence of an "increasing trend". In [15], the authors first filter out some OWDs that appear to be outliers. Then, they apply two statistical tests (Pairwise-Comparison-Test and Pairwise-Difference-Test) on the remaining time series to detect if the OWDs present an overall increasing trend or not. Both tests, however, require a key threshold. In [15], that threshold remains the same for all paths and load conditions.

In this paper, we also use the OWD approach to classify a stream as type-G or type-L. Instead of a static threshold, however, we propose an adaptive algorithm to detect the presence of increasing OWD trend. In more detail, suppose that a probing stream consists of K packets. Let D_i and A_i be the OWD and receive time of the i'th packet, respectively. Then, the pairwise OWD slope $S_{i,j}$ of two packets i and $j < i$ of the stream is given by

$$S_{i,j} = \frac{D_i - D_j}{A_i - A_j}$$

We expect that if $R_i > A_\tau(t)$, then the OWDs of the probing stream will exhibit increasing trend due to queueing at the tight link, and so most of the $S_{i,j}$ values will be positive. Otherwise, the $S_{i,j}$ values will be randomly distributed around zero. To

filter out any outliers, and also to summarize the distribution of $S_{i,j}$ into a point estimate, we work with the median $\tilde{S}$ of the $S_{i,j}$ values. Then, similarly with [15], we compare $\tilde{S}$ with a threshold $\beta \geq 0$. If $\tilde{S} \geq \beta$, the corresponding stream is classified as type-G; otherwise, it is of type-L. The appropriate value of β, however, depends on the burstiness and the load intensity of the traffic in that path [8], [14]. So, instead of attempting to estimate an "optimal" but fixed β, we instead propose an adaptive algorithm for the selection of β. The basic idea behind this algorithm is that β should be chosen such that, if rate R is the p'th percentile of the avail-bw distribution, then the classification of N streams of rate R should report roughly pN streams of type-G.

Specifically, suppose that we transmit N streams of rate R_p. We compute the median slope of each probing stream, and then order the streams so that $\tilde{S}_1$ is the lowest slope and $\tilde{S}_N$ the highest. Now, suppose that we somehow know that rate R_p corresponds to the p'th percentile of the avail-bw distribution F_τ. Then, based on Proposition 1, we expect that on the average, pN out of the N probing streams will be of type-G. So, the threshold β should be chosen so that it is

$$\tilde{S}_{\lfloor pN \rfloor} \leq \beta < \tilde{S}_{\lfloor pN \rfloor + 1} \tag{11}$$

Otherwise, if β is chosen outside this range, the classification of streams in type-G or type-L will be biased and the p'th percentile will not be estimated correctly. In the following, we set $\beta = \tilde{S}_{\lfloor pN \rfloor}$.

Of course the issue with the previous approach is that in general we do not know any percentile of the avail-bw distribution; this is actually what we aim to infer. Consider however the following iterative approach, based on the principles of stochastic optimization. If we start with a "sufficiently good" value of β, then we can use the algorithms of §III or §IV to roughly estimate any given percentile of F_τ. Then, we can use that information in adjusting β based on (11). The new value of β is probably better than the previous, given that it is based on the estimation of the underlying avail-bw distribution rather than on a fixed threshold.

The previous approach can be executed iteratively, adjusting β after each round of the algorithms of §III or §IV. After iteration n, the new threshold β_n can be determined based on a EWMA operator as follows

$$\beta_n = \omega \tilde{S}_{\lfloor pN \rfloor} + (1 - \omega)\beta_{n-1} \tag{12}$$

The use of EWMA, instead of just replacing β with $\tilde{S}_{\lfloor pN \rfloor}$ after each iteration, aims to de-noise the estimation of β.

As it can happen with such adaptive algorithms, their convergence depends on the selection of the initial point and on the convergence parameters [16]. If the initial point is not in the vicinity of the global optimum, it is possible that the algorithm will converge to a local optimum. Also, if the convergence parameter (in our case, the parameter ω) is too large, the algorithm may fail to converge. Here, the convergence depends on the initial selection of β and on the parameter ω. We determine the initial threshold β based on the statistical tests presented in [15]. The parameter ω was tuned through simulations and is set to ω=0.05. We do not claim however that these are optimal values or that the convergence of β to its optimal value is guaranteed.

Since the adaptive selection of β is coupled with the estimation of the avail-bw variation range, the two algorithms are jointly evaluated in §VI. Here we simply present two examples of how the adaptive selection of β can succeed or fail to accurately estimate a certain percentile of the avail-bw distribution (see Figures 7 and 8, respectively). These results were obtained from testbed experiments with trace-driven cross traffic (the experimental setup is described in more detail in §VI). We emphasize that the cases of failed convergence are rare, at least in all our validation experiments and simulations. We show one such example however (in Figure 8) to demonstrate that a failure to converge to the optimal value of β can lead to a certain bias in the estimation of a given percentile.

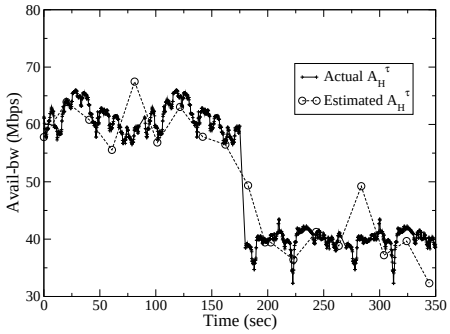

Fig. 7. Example of successful convergence of β.

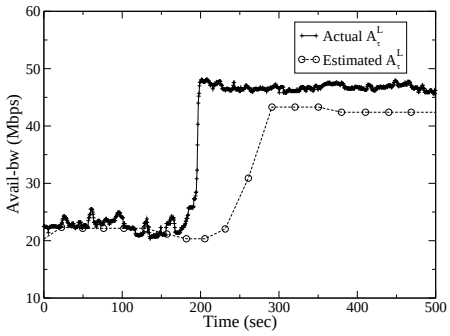

Fig. 8. Example of unsuccessful convergence of β.

VI. PATHVAR

We have implemented both the non-parametric and parametric estimation algorithms in a tool called *Pathvar*. Pathvar consists of two components: the sender is responsible for transmitting the probing streams, while the receiver analyzes the One-Way Delays (OWDs) in each stream and determines whether a stream is of type-G or type-L. The two peers use a TCP connection to reliably transfer control messages and UDP datagrams for the probing streams. The number of streams N, stream duration τ, and the avail-bw variation range probabilities (p^L, p^H) are the key Pathvar inputs.

N determines the number of probing streams of a certain rate R that the sender will transmit to the receiver in each iteration. As described in §II-B, N determines the accuracy with which we can infer the probability $F_\tau(R)$ that corresponds to the probing rate R. Based on the results of §II-B, Pathvar uses $N = 40$ targeting for an error tolerance $\rho = 0.05 - 0.06$ in the estimation of $F_\tau(R)$. The stream duration τ determines the measurement time scale for the average avail-bw, and it has to be chosen by the user based on the application requirements. We note however that τ should not be more than a few hundreds of milliseconds. The reason is that high-rate probing streams that last for too long can be network intrusive, causing congestion and packet losses.

Pathvar sends a total of $2N$ streams in each iteration: N streams at each of two different rates (as described in §III and §IV). A stream is sent only when the previous stream has been acknowledged by the receiver, meaning that the duration of each iteration is $2N(\tau + RTT)$, where RTT is the Round-Trip Time between the two peers. Additionally, the probing streams of the two rates are interleaved, i.e. a stream of rate R_1 is followed by a stream of rate R_2, so that Pathvar probes the avail-bw distribution with the two rates almost simultaneously. After all $2N$ streams are received, the receiving peer examines whether each stream is of type-G or type-L, as described in §V.

Pathvar invokes either the non-parametric or the parametric algorithm depending on the specified time scale τ. If the latter is larger than 100msec, we prefer to use the parametric algorithm for three reasons. First, based on the measurement results of [12], we expect that in those time scales the avail-bw process will be sufficiently close to Normal. Second, with large values of τ, and consequently with long probing streams, the parametric algorithm gives us the advantage that we can select lower probing rates, reducing the intrusiveness of the measurements. Third, the parametric algorithm is not iterative and so it is less dependent on the stationarity assumption; that assumption can be questioned when τ is large.

In Pathvar, the sender timestamps each probing packet just before transmission. Upon arrival, the receiver records the arrival time and measures the OWD. The measured OWD differs from the actual OWD due to the clock offset between the two measurement peers. However, since we are only interested in the OWD differences, the clock offset does not affect the measurements as long as it is constant. The presence of clock skew does not affect Pathvar because the stream duration is less than a second, while the typical magnitude of clock skew in modern quartz clocks is in the order of only a few microseconds per second. Context switching is another source of errors in the OWD measurements because buffering of packets in the kernel adds a variable delay component in the measured OWDs. Pathvar implements simple techniques to detect context switching and remove its effects, similar with the techniques developed for Pathload [15]. Finally, in the current version of Pathvar, the initial probing rates have to be provided by the user based on past experience with the measured path.

A. Testbed examples

We have evaluated the accuracy of Pathvar with both simulations and testbed experiments. The tight link at the testbed is a Fast Ethernet segment between two switches. The traffic at the tight link is generated by replaying the aggregate packet stream observed in NLANR traces. So, the packet sizes and interarrivals are based on realistic Internet traffic. To create non-stationary traffic conditions, and in particular level shifts, we concatenate traces with significantly different load.

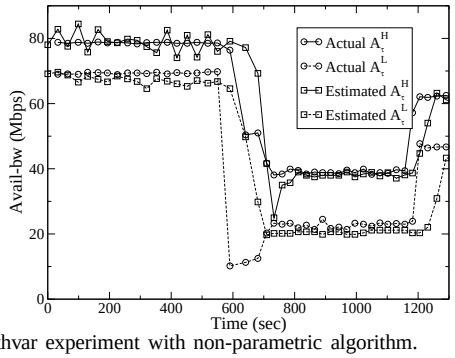

Fig. 9. Pathvar experiment with non-parametric algorithm.

Figure 9 shows the actual and estimated 10%-90% percentile range, measured in a time scale τ=40msec, for a non-stationary traffic load. The estimation in Figure 9 is performed with the non-parametric algorithm (b=0.2). A first observation is that, during the stationary epochs, Pathvar tracks the actual variation range within 10% or better. A second observation is that after level-shift events, the non-parametric estimation algorithm needs some considerable amount of time (100-200sec) to reconverge to the correct variation range. This delay can be reduced by using a larger value of b, but with an associated cost in the accuracy of the estimation during stationary periods. A future improvement that we consider is to dynamically increase b, upon the detection of frequent level-shifts or other forms of non-stationarity, and to gradually decrease b when the avail-bw remains at the same level.

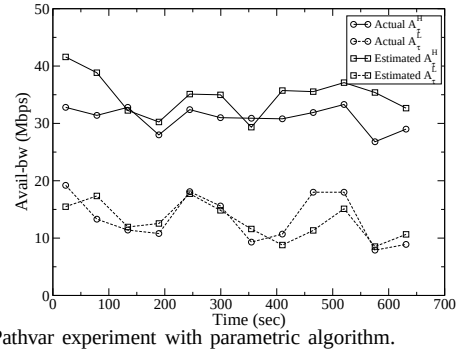

Fig. 10. Pathvar experiment with parametric algorithm.

In Figure 10, we use the parametric algorithm instead. The actual and estimated 10%-90% percentile range are measured in a time scale τ=250msec. The traffic load is non-stationary (generated from replaying multiple times a 90-sec NLANR trace), but with a marginal distribution that is quite close to the Gaussian model.

A first observation is that the tool needs about $2N(\tau + RTT)$=55 seconds to generate each estimate of the variation range. Notice that this large latency is not an intrinsic characteristic of the parametric algorithm, but it is due to the large

measurement time scale τ and the associated long duration of each probing stream. Second, the estimation error is in the order of 10-20%. The reader should not conclude that the parametric algorithm is less accurate than the non-parametric algorithm. In general, the accuracy of the two algorithms is comparable when they are both applied on the same traffic and with the same value of τ, as long as the traffic process is stationary and Gaussian.

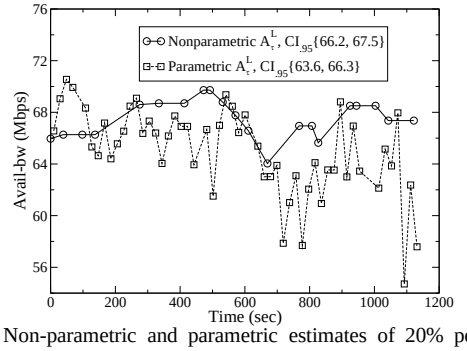

Fig. 11. Non-parametric and parametric estimates of 20% percentile for τ=40msec.

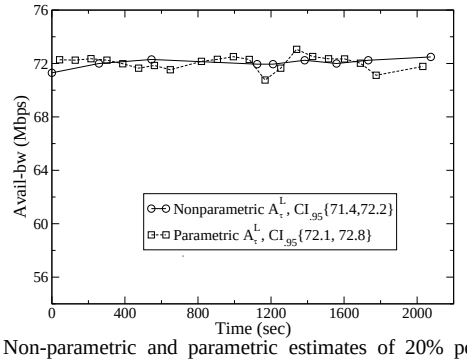

Fig. 12. Non-parametric and parametric estimates of 20% percentile for τ=140msec.

Next, we compare the accuracy of the parametric and non-parametric algorithms under the same traffic load. In the following graphs, we show a single avail-bw percentile, rather than a variation range, to avoid cluttering. For each algorithm, we show the time series of the 20-th percentile avail-bw estimates, as well as the 95-th Confidence Intervals (CI) of those estimates.

Figures 11 and 12 show the effect of the measurement time scale τ on the accuracy of the two algorithms. The traffic is generated by replaying the NLANR trace MRA-1062182531. The 20-th percentile of the avail-bw distribution during the entire trace is 67.0Mbps at τ=40msec, and 72.1Mbps at τ=140msec. The non-parametric algorithm estimates this percentile quite accurately in both measurement time scales. The parametric algorithm, on the other hand, is accurate when τ=140msec, but it underestimates the given percentile when τ=40msec. The reason is that in that shorter measurement time scale, the traffic deviates significantly from the Normal distribution.

Figures 13 and 14 show the effect of the degree of statistical multiplexing ("vertical aggregation") on the accuracy of the two algorithms. To generate traffic with a lower degree of multiplexing we replay 20 large flows extracted from an NLANR trace, and to generate traffic with higher multiplexing

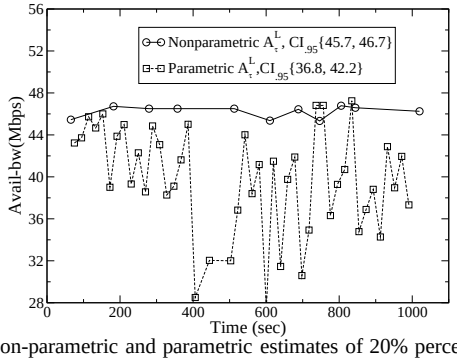

Fig. 13. Non-parametric and parametric estimates of 20% percentile for low vertical aggregation.

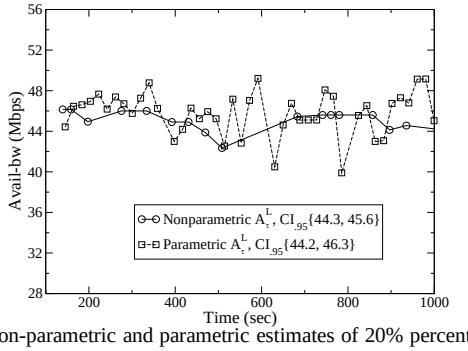

Fig. 14. Non-parametric and parametric estimates of 20% percentile for high vertical aggregation.

we replay approximately 3000 smaller flows from the same trace. The 20-th percentile of the avail-bw distribution during the entire trace is 45.9Mbps and 44.9Mbps, respectively. In both cases, the average traffic rate (and avail-bw) is about the same. The non-parametric algorithm produces accurate estimates of 20-th percentile with both degrees of multiplexing. The parametric algorithm, on the other hand, is accurate only when the traffic is highly aggregated. The reason is that in the latter the traffic deviates significantly from the Normal distribution.

To summarize the experiments of this section, the parametric algorithm performs better than the non-parametric algorithm under non-stationary conditions (especially level shifts and traffic spikes) because it is not iterative. On the other hand, if the traffic is not Gaussian because of low horizontal or vertical aggregation, then the non-parametric algorithm performs better. Obviously, the accuracy of Pathvar is worse when both previous assumptions do not hold, i.e., with non-stationary and non-Gaussian traffic. This may be the case in paths where the tight link is the host network interface or a LAN link. In such environments, the traffic load is sporadic, generated by only a few high-throughput flows, and so the resulting avail-bw process can be both non-stationary and non-Gaussian.

B. Internet Experiments

We have also used Pathvar to measure the avail-bw variation range in several Internet paths. The objective of these experiments is not to perform validation, given that we do not know the actual avail-bw distribution, but to observe how the avail-bw variation range changes with time in real Internet paths. In this section, we show some preliminary results from

two Internet paths between Georgia Tech (in Atlanta GA) and two universities in Greece (in Ioannina and Heraclion-Crete). In both cases, we have evidence that the tight link is the campus access link of the Greek universities (based on the corresponding MRTG graphs).

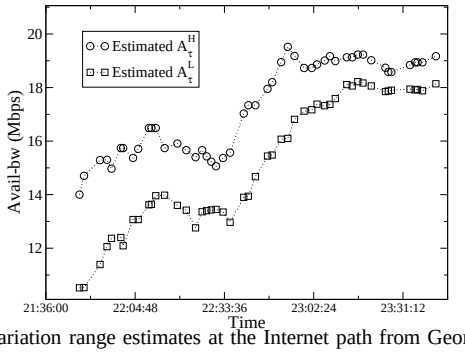

Fig. 15. Variation range estimates at the Internet path from Georgia Tech to University of Ioannina.

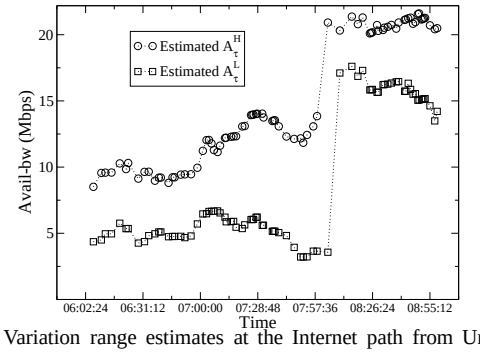

Fig. 16. Variation range estimates at the Internet path from University of Crete to Georgia Tech.

Figure 15 shows the estimated 20%-80% variation range of the avail-bw for the path from Georgia Tech to University of Ioannina over a two-hour time period. The time reported in the x-axis refers to local time in Greece. The measurement timescale is 40msec and the estimates are obtained using the non-parametric algorithm. A first observation is that the avail-bw gradually increases, especially after 22:30pm. A second, more interesting observation is that the variation range decreases as the average avail-bw increases. We discuss the relation between avail-bw variability and tight link utilization in the next section.

Figure 16 shows the estimated 20%-80% variation range for the avail-bw for a path from University of Crete (UoC) in Heraclion to Georgia Tech over a three-hour window. The time reported in the x-axis refers to local time in Greece. The measurement timescale is 40msec and the estimates are obtained using the non-parametric algorithm. A first observation is that the avail-bw shows a sharp increase at 8:00am. We confirmed this unusual behavior with the MRTG graph of the UoC campus access link. One possible explanation is that certain applications (e.g., p2p file transfers) are blocked during working hours. Another interesting observation is that even though the average avail-bw is roughly constant between 6:00am and 8:00am, the variation range fluctuates significantly. This illustrates that estimating only the average avail-bw may be an insufficient indicator for the load of a network path.

VII. Variability factors

The previous sections focused on the estimation of the avail-bw variation range through end-to-end measurements. Which are the factors, however, that affect the variability of the avail-bw distribution? Why does the traffic appear to be more "bursty" in some paths than in other paths? Two pieces of conventional wisdom are that "heavier load conditions also produce wider traffic variations" and that "a higher degree of multiplexing makes the traffic smoother". Under which conditions, however, are these statements true?

In this section, we focus on four different factors, and show how they affect the variability of the avail-bw distribution. These factors are the traffic load at the tight link, the number of competing flows, the rate of competing flows, and of course the measurement time scale. The first three factors are related to the traffic characteristics at the tight link, while the last factor is related to the way the avail-bw is measured. Even though these factors have been examined in different contexts before, our focus here is specifically on the way these factors affect the variation range of the avail-bw distribution.

The following results are based on a simulation study in which we measure the avail-bw variation range as we vary each of the previous four factors. Specifically, we have implemented an NS module for Pathvar that is identical to the actual prototype described in §VI. Unless noted otherwise, we use the following parameters in the simulation: τ=50msec and N=40 streams. The simulation topology includes a tight link with capacity C_t=50Mbps. The traffic at the tight link is generated by a large number of edge nodes, and it resembles short HTTP flows running over TCP NewReno. Each such flow transfers 10-15 packets from a server to a client through the tight link, sleeps for a random time interval (that is adjusted based on the desired average load), and then repeats the previous cycle.

During each simulation Pathvar runs M=25 consecutive times, estimating a 10%-90% variation range $[A_\tau^L(i), A_\tau^H(i)]$, for $i = 1 \dots M$, after each run. To summarize the M ranges into a single figure, we calculate the average width $\hat{V}$ of the estimated variation ranges as follows

$$\hat{V} = \frac{\sum_{i=1}^{M} (A_\tau^H(i) - A_\tau^L(i))}{M} \tag{13}$$

We also calculate the standard deviation $\hat{E}$ of these M samples, to quantify their dispersion around $\hat{V}$.

The following simulations also serve as a validation study of Pathvar. To do so, we collect a traffic trace at the tight link during the simulation and then measure the width $V = A_\tau^H - A_\tau^L$ of the actual variation range $[A_\tau^L, A_\tau^H]$. The comparison of $\hat{V}$ with V indicates whether Pathvar can successfully estimate the avail-bw variation range width.

A. Effect of tight link utilization

The first factor we consider is the average utilization u at the tight link. From queueing theory we know that the variance of the queueing delay or backlog in most queueing systems increases as the utilization increases [17]. How does the utilization affect the avail-bw variability however?

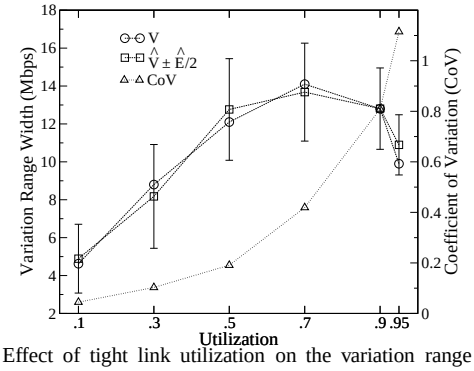

Fig. 17. Effect of tight link utilization on the variation range width (left Y-axis) and CoV (right Y-axis).

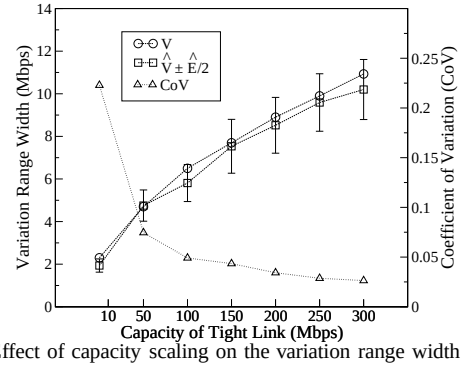

Fig. 18. Effect of capacity scaling on the variation range width (left Y-axis) and CoV (right Y-axis).

In Figure 17, we show the estimated and the actual variation range width for six values of u. The utilization is controlled by adjusting the number of TCP clients. The measurement time scale is τ=30msec. The first observation, in terms of validating Pathvar, is that the actual variation range width V is very close to the estimation $\hat{V}$, and always within a range of $\pm\hat{E}/2$.

Second, the variation range width increases with u up to a certain point. After that point, the variation range width decreases with u. The point of the maximum variability in this particular simulation occurs when the utilization is around 70%, but this depends on the measurement time scale and on the characteristics of the traffic mix. What is the reason for this non-monotonic variation of V with u? Intuitively, when u is relatively low, i.e., in light load conditions, V increases with u because of the increasing variability in the offered load. With Poisson traffic, the variance of the offered load increases linearly with the average traffic rate. As u approaches 100%, however, the tight link often becomes saturated. During the time periods that the tight link is saturated, the departure rate at the output of the tight link remains constant, and so the avail-bw variability drops. In the extreme case that the tight link is always fully utilized, the avail-bw remains constantly zero, and so its variability is also zero. Note that it is important to distinguish between the traffic variability at the input of a link versus at the output. Even if the input rate has high traffic spikes, the traffic rate at the output is essentially "clamped" by the link capacity. It is this clamping effect that causes the variation range reduction at high loads. This effect has been also studied in [18], examining the relation between load and traffic variance.

It is interesting that the Coefficient of Variation (CoV) of the traffic at the output of the tight link follows a different trend than the variation range (see Figure 17). Specifically, the CoV, which is defined as the standard deviation of the avail-bw over the average avail-bw, increases monotonically with u. This trend should not be misinterpreted as an indication that heavier loads cause wider traffic variability. This is only true in relative terms, when the avail-bw variability is normalized by the average avail-bw. In absolute terms, instead, the avail-bw variability reaches its maximum when the link is significantly loaded but not congested.

B. Statistical multiplexing effects and scaling models

Another conventional wisdom is that a higher degree of statistical multiplexing, under the same load conditions, makes the traffic smoother. To examine the validity of this statement, we first need to clarify what it means to increase the degree of multiplexing at a link.

We distinguish between two scaling models. In the first, referred to as "capacity scaling", we increase the capacity of the tight link proportionally to the number of competing flows. The average rate of each flow, as well as the utilization of the tight link, remain constant. In the second, referred to as "flow scaling", we increase the number of competing flows at the tight link while proportionally decreasing their average rate; the capacity and utilization of the tight link remain constant. Note that in both scaling models the number of competing flows at the tight link increases, while the utilization of that link remains the same.

1) Capacity Scaling: To simulate capacity scaling, we increase the number of TCP clients U proportionally to the capacity C_t of the tight link. Specifically, U is increased from 3 to about 90, C_t is increased from 10Mbps to 300Mbps, while the tight link utilization is kept constant at 50%. Each TCP client transmits an average of 1000 packets, then sleeps for a random time interval between 2-5 secs, and then repeats the previous cycle. The capacity of the access link of each client is fixed to 2Mbps, and the average rate of each TCP flow also remains constant.

Figure 18 shows the effect of capacity scaling on the avail-bw variation range width and CoV. Interestingly, capacity scaling has a different effect on the avail-bw variability, depending on whether we look at the variation range width or at the CoV. The former increases with C_t, while the latter decreases. To understand why, suppose that X_i is the traffic process generated by flow i, while Y=$\sum_{i=1}^{U} X_i$ is the aggregate traffic process at the tight link generated by U flows. If the U flows are independent, which is a reasonable assumption when the tight link is not congested, and if we assume for simplicity that the flows are identically distributed with $\text{Var}[X_i] = \text{Var}[X]$, then we have that $\text{Var}[Y] = U\text{Var}[X]$. So, the variation range width of Y will increase with U. Actually, if Y is Gaussian, then the width V of a symmetric variation range is proportional

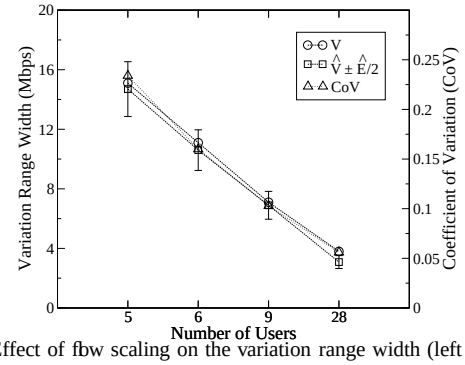

Fig. 19. Effect of fbw scaling on the variation range width (left Y-axis) and CoV (right Y-axis).

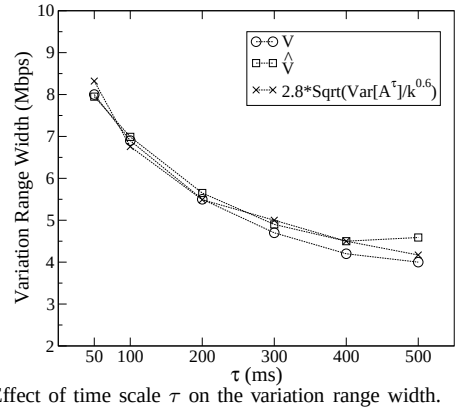

Fig. 20. Effect of time scale τ on the variation range width.

to the standard deviation of Y.[2] In that case, V increases proportionally to $\sqrt{U}$. The CoV of the avail-bw, on the other hand, is equal to

$$\text{CoV} = \frac{\sqrt{C - U\text{Var}[X]}}{C - UE[X]} = \frac{\sqrt{U\text{Var}[X]}}{C - UE[X]} \tag{14}$$

In capacity scaling, U increases proportionally with C, and so the CoV decreases as $1/\sqrt{C}$. So, the relative variability of the avail-bw decreases with capacity scaling, even though the absolute width of the variation range increases.

2) Flow scaling: To simulate flow scaling, we increase the number of users U (TCP clients) decreasing proportionally the average traffic rate of each user. This rate reduction is achieved by decreasing the capacity C_e of the edge link that connects each user to the tight link. The throughput of the TCP transfers is determined by C_e in these simulations. Figure 19 shows the effect of capacity scaling on the avail-bw variation range width and CoV. In the case of flow scaling, note that both the absolute variation range as well as the CoV decrease as the number of users increases.

An interesting question is, why does the variation range width decrease with flow scaling, but it increases with capacity scaling? Consider again the simple model of the previous paragraph. The variance of Y is $\text{Var}[Y] = U\text{Var}[X]$, assuming independence among the U users. The difference with capacity scaling, however, is that in flow scaling the variance $\text{Var}[X]$ of each flow decreases as U increases. This is at least the case for most traffic processes: their variance decreases as the average rate decreases. In the Poisson process, the variance is simply equal to the average rate. In the Poisson Pareto Burst Process [19], which creates self-similar traffic, the variance is proportional the square of the average rate. As long as $\text{Var}[X]$ decreases faster than the increase in U, the variance $\text{Var}[Y]$ will decrease as we increase the number of users. This is the case for the traffic mix that we simulate in Figure 19, or for the Poisson Pareto Burst Process. On other hand, this would not be the case for the Poisson process, in which $\text{Var}[Y]$ remains constant as we increase U.

The fact that the CoV decreases with flow scaling also depends on the relation between U and $\text{Var}[X]$. As in the previous paragraph, the avail-bw CoV is given by (14). Since the denominator (average avail-bw) remains constant, the CoV decreases if the variance of individual flows decreases faster than the flow average rate.

[2]For instance, if Y is Gaussian, then it is easy to calculate that the 10%-90% variation range width is equal to 2.56σ, where σ is the std-deviation of Y.

C. Effect of measurement timescale

As mentioned in the Introduction, the variability of the avail-bw decreases with τ. The rate of decrease, however, can be very different depending on the correlation structure of the underlying traffic process. If $A_\tau(t)$ is an IID random process, then the variance decreases inversely proportional with the length of the averaging time scale

$$\text{Var}[A_{k\tau}] = \frac{\text{Var}[A_\tau]}{k} \tag{15}$$

On the other hand, if $A_\tau(t)$ is an exactly self-similar process with Hurst parameter $0.5 < H < 1$, the variance decreases slower

$$\text{Var}[A_{k\tau}] = \frac{\text{Var}[A_\tau]}{k^{2(1-H)}} \tag{16}$$

A tool such as Pathvar can estimate the variation range in different time scales. Consequently, it is possible to infer through end-to-end measurements whether the avail-bw process is an IID or a self-similar process, and in the latter, to measure the local Hurst parameter in a certain range of time scales.

Figure 20 shows the actual and the estimated variation range width of the avail-bw in six measurement time scales: τ=50, 100, 200, 300, 400, and 500msec. As we expected, V decreases with τ. More interestingly, however, the decrease rate is consistent with a self-similar process with Hurst parameter H=0.7. Of course this scaling behavior is valid locally in the previous range of τ; the Hurst parameter may be different in larger time scales.

VIII. Conclusions

This paper focused on the estimation of the avail-bw variation range using end-to-end network measurements. To the extent of our knowledge, this is the first work that aimed to measure the variability of the avail-bw, rather than its mean. We developed and evaluated two complementary estimation algorithms. The selection among the two algorithms depends on the measurement time scale, the degree of multiplexing at the tight link, and the stationarity of the traffic at the measured path. The accuracy of the proposed algorithms will probably be satisfactory for most applications, with relative errors up to 10-20%.

Several important problems remain open for future work. First, it is not clear whether the avail-bw variability estimation is feasible when the traffic at the tight link is both non-stationary and non-Gaussian. Second, the proposed estimation techniques cannot be applied for larger measurement time scales (say more than one second) because the corresponding probing streams would be very network intrusive and they would probably cause packet drops. Finally, the presence of multiple bottlenecks at a path causes underestimation errors in all avail-bw estimation techniques, and it would also affect the algorithms presented in this paper.

ACKNOWLEDGMENTS

We are grateful to E. Markatos and L. Tassiulas for providing us with computer accounts at Greek universities. This work has benefited from the NLANR MOAT traffic collection project, which has been supported by the NSF cooperative agreements ANI-0129677 and ANI-9807479.

REFERENCES

[1] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk, "Multifractal Cross-Traffic Estimation," in *Proceedings ITC Specialist Seminar on IP Traffic Measurement, Modeling, and Management*, Sept. 2000.

[2] B. Melander, M. Bjorkman, and P. Gunningberg, "A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks," in *IEEE Global Internet Symposium*, 2000.

[3] M. Jain and C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 537–549, Aug. 2003.

[4] N. Hu and P. Steenkiste, "Evaluation and Characterization of Available Bandwidth Probing Techniques," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, pp. 879–894, Aug. 2003.

[5] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths," in *Proceedings of Passive and Active Measurements (PAM) workshop*, Apr. 2003.

[6] J. Strauss, D. Katabi, and F. Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools," in *Proceedings of ACM/USENIX Internet Measurement Conference (IMC)*, 2003.

[7] A. Akella, S. Seshan, and A. Shaikh, "An Empirical Evaluation of Wide-Area Internet Bottlenecks," in *Proceedings of ACM/USENIX Internet Measurement Conference (IMC)*, 2003.

[8] M. Jain and C. Dovrolis, "Ten Fallacies and Pitfalls in End-to-End Available Bandwidth Estimation," in *Proceedings of ACM/USENIX Internet Measurement Conference (IMC)*, Oct. 2004.

[9] K. Park and W. Willinger (editors), *Self-Similar Network Traffic and Performance Evaluation*. John Willey, 2000.

[10] R. S. Prasad, M. Murray, C. Dovrolis, and K. Claffy, "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools," *IEEE Network*, Nov. 2003.

[11] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[12] J. Kilpi and I. Norros, "Testing the Gaussian Approximation of Aggregate Traffic," in *Proceedings of ACM/USENIX Internet Measurement Workshop (IMW)*, Nov. 2002.

[13] NLANR MOAT, "Passive Measurement and Analysis," http://pma.nlanr.net/PMA/.

[14] X. Liu, K. Ravindran, B. Liu, and D. Loguinov, "Single-Hop Probing Asymptotics in Available Bandwidth Estimation: A Sample-Path Analysis," in *Proceedings of ACM Internet Measurement Conference*, 2004.

[15] M. Jain and C. Dovrolis, "Pathload: A Measurement Tool for End-to-End Available Bandwidth," in *Proceedings of Passive and Active Measurements (PAM) Workshop*, Mar. 2002, pp. 14–25.

[16] J. Spall, *Introduction to Stochastic Search and Optimization*. Willey Interscience, 2003.

[17] G. Bolch, S.Greiner, H.Meer, and K.S.Trivedi, *Queueing Networks and Markov Chains*. John Wiley and Sons, 1999.

[18] J. W. X. Tian and C. Ji, "A Unified Framework for Understanding Network Traffic Using Independent Wavelet Models," in *Proceedings of IEEE INFOCOM*, June 2002.

[19] M. Zukerman, T. D. Neame, and R. G. Addie, "Internet Traffic Modeling and Future Technology Implications," in *Proceedings of IEEE INFOCOM*, 2003.