Ten Fallacies and Pitfalls on
End-to-End Available Bandwidth Estimation [*]

Manish Jain
Georgia Tech
jain@cc.gatech.edu

Constantinos Dovrolis
Georgia Tech
dovrolis@cc.gatech.edu

ABSTRACT

The area of available bandwidth (avail-bw) estimation has attracted significant interest recently, with several estimation techniques and tools developed during the last 2-3 years. Unfortunately, some key issues regarding the avail-bw definition, estimation, and validation remain vague or misinterpreted. In this note, we first review the previous work in the area and classify the existing techniques in two classes: direct probing and iterative probing. We then identify ten misconceptions, in the form of fallacies or pitfalls, that we consider as most important. Some misconceptions relate to basic statistics, such as the impact of the population variance on the sample mean, the variability of the avail-bw in different time scales, and the effect of the probing duration. Other misconceptions relate to the queueing model underlying these estimation techniques. For instance, ignoring that traffic burstiness or the presence of multiple bottlenecks can cause significant underestimation errors. Our objective is not to debunk previous work or to claim that some estimation techniques are better than others, but to clarify a number of important issues that cover the entire area of avail-bw estimation so that this important metric can be better understood and put in practical use.

Categories and Subject Descriptors: C.2.3 [Network Operations]: *Network Monitoring*

General Terms: Measurement, Performance

Keywords: Active measurements, available bandwidth, bandwidth estimation, packet pairs and trains, measurement tools

1. INTRODUCTION

Estimating the available bandwidth (avail-bw) of end-to-end network paths with active measurements has attracted significant interest recently. Several estimation techniques and software tools have been developed, including Delphi

[1], TOPP [2], Pathload [3], IGI [4], Pathchirp [5], Spruce [6], and Bfind [7], while several more techniques are currently being reviewed or developed. On the positive side, this work reflects that the networking community recognizes the importance of the avail-bw metric and its many practical applications. On the negative side, however, we find that some key issues regarding the avail-bw definition, estimation, and validation remain vague or misinterpreted in both the general understanding of our community, but also in published work. The objective of this note is to clarify what we think of as the ten most important misconceptions about avail-bw estimation. Our objective is not to debunk previous work or to claim that some estimation techniques (or our estimation technique [3]) are better than others. Actually, the clarification of the following issues may lead to better estimation techniques, and also, it may allow a more fair comparison between existing measurement tools.

Definitions

We first define the avail-bw of a network link and then of an end-to-end path. Consider a store-and-forward network link i with capacity C_i. Let $u_i(t)$ be the instantaneous utilization of the link at time t, i.e., 0 if the link is idle and 1 if the link transmits a packet at t. The average utilization $u_i(t, t+\tau)$ of link i during the time interval $(t, t+\tau)$ is

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

The avail-bw $A_i(t, t+\tau)$ of link i during the time interval $(t, t+\tau)$ is defined as the average unutilized capacity in that duration,

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

Consider now a network path that goes through a given set of H links. By convention, the end-to-end avail-bw $A(t, t+\tau)$ of the network path during $(t, t+\tau)$ is defined as the minimum of all the link available bandwidths in the same interval,

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

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.

Since the avail-bw varies with time, it can be viewed as a random process $A^\tau(t)$, where τ is the *averaging time scale* in (3). Typically, and especially in the traffic modeling literature [8], the traffic process is assumed to be stationary, at least over intervals that span several minutes to a few hours,

modeling any variations around the mean with second-order statistics. We adopt the same assumption here, and assume that $A^\tau(t)$ is a stationary random process with mean $\mu_A = E[A^\tau(t)]$ and variance $\text{Var}[A^\tau]$. The mean avail-bw μ_A does not depend on the averaging time scale. The variance, however, is strongly dependent on the averaging time scale, and generally decreases with τ. Additionally, the rate of decrease of $\text{Var}[A^\tau]$ depends on the correlation structure of $A^\tau(t)$. If $A^\tau(t)$ is an IID random process, then the variance decreases inversely proportional with the length of the averaging time scale

$$\text{Var}[A^{k\tau}] = \frac{\text{Var}[A^\tau]}{k} \qquad (4)$$

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

$$\text{Var}[A^{k\tau}] = \frac{\text{Var}[A^\tau]}{k^{2(1-H)}} \qquad (5)$$

As will be discussed later, the relation between the variance of $A^\tau(t)$ and τ is very important, and it has been largely ignored so far in the avail-bw estimation literature.

Avail-bw estimation at a single link with fluid traffic

All the existing avail-bw estimation techniques are based, at least in terms of their basic idea, on a single-link model with fluid cross traffic of (constant) rate. Let C_t be the capacity of that link and R_c be the rate of cross traffic, with the avail-bw being $A = C_t - R_c > 0$. The fluid assumption means that during any time interval of length τ, the amount of arriving cross traffic at the link is $R_c\tau$.

Suppose that we send to this link a periodic probing stream with rate R_i and packet size L; the length of the stream does not matter at this point. The interarrival between two consecutive probing packets is $\delta_i = L/R_i$. The amount of cross traffic that arrives at the link during any interval of δ_i is $X_c = R_c\delta_i$, and the total amount of traffic that arrives at the link in the same interval is $L + X_c$ (including one probing packet). On the other hand, the maximum amount of traffic the link can transmit in the same interval is $C_t\delta_i$. It is straightforward to show that if $R_i > A$, then the link receives more traffic than it can service, i.e. $L + X_c > C_t\delta_i$.

In that case, when $R_i > A$, the extra arriving traffic at the link accumulates at the link's buffers, increasing the queue size with every probing packet we send. It is also easy to show that the queue size increase Δq during an interval of length δ_i is

$$\Delta q = \delta_i(R_i - A) = L\frac{R_i - A}{R_i} \qquad \text{if} \;\; R_i > A \qquad (6)$$

Two ways to detect that $R_i > A$ from end-to-end measurements are through increase in the One-Way Delay (OWD) of the probing packets, and through a lower output rate R_o at the receiver compared to the input rate R_i at the sender. Specifically, the OWD increase Δd between two successive probing packets during an interval of length δ_i is determined by the queue size increase in the same interval

$$\Delta d = \frac{\Delta q}{C_t} = \frac{L}{C_t}\frac{R_i - A}{R_i} \qquad \text{if} \;\; R_i > A \qquad (7)$$

Also, the interarrival between two consecutive probing packets after the link (at the receiver) is $\delta_o = \delta_i + \Delta d$, and so the

rate with which the probing stream arrives at the receiver is

$$R_o = \frac{L}{\delta_o} = \frac{R_iC_t}{C_t + (R_i - A)} \qquad \text{if} \;\; R_i > A \qquad (8)$$

The key point here is that the output rate is less than the input rate $R_o < R_i$ when $R_i > A$.

On the other hand, if $R_i \leq A$ the link can transmit all the received traffic and so $\Delta q = \Delta d = 0$, meaning that the OWDs do not increase, and $R_o = R_i$ meaning that the output rate is equal to the input rate.

We next identify two different avail-bw estimation approaches that are based on the previous model. All the existing estimation techniques and tools are based on these general approaches.

Direct probing

In direct probing, each probing stream results in a sample of the avail-bw. The sender transmits a periodic probing stream of rate R_i and receiver measures the output rate R_o. The basic idea is that, if R_i is larger than the avail-bw, Equation (8) can be solved for the only unknown A

$$A = C_t - R_i(\frac{C_t}{R_o} - 1) \qquad (9)$$

Note that direct probing samples the avail-bw process with each packet train, as long as the input rate is sufficiently high. The main assumption in the direct probing approach, however, is that the tight link capacity C_t is known. We will return to this point in §3.

Iterative probing

In iterative probing, we do not need to know the capacity of the tight link. The sender transmits a periodic probing stream k with rate $R_i(k)$. The rate $R_i(k)$ varies either linearly, or as a function of the outcome of previous streams. If the k'th stream gives $R_o(k) < R_i(k)$, or if the OWDs of that stream are increasing, then we know that $R_i(k) > A$; otherwise, it is $R_i(k) \leq A$. The basic idea is that, through a sequence of streams with different rates, iterative probing can converge to A. Thus, the basic equation behind iterative probing is:

$$R_o(k) < R_i(k) \quad \text{if} \; R_i(k) > A; \quad \text{else} \; R_i(k) \leq A \qquad (10)$$

A key point about iterative probing is that it does not sample the avail-bw process; instead, it only samples whether a rate is larger than the avail-bw or not.

2. CLASSIFICATION

We next briefly review and classify each existing avail-bw estimation technique. The techniques are presented in the order in which they were published.

Delphi: Delphi is the canonical example of direct probing [1]. An interesting point about that work is that it assumes a multifractal model for the path's cross traffic. Similar to other direct probing techniques, Delphi assumes that the avail-bw is limited by a single tight link and that the capacity of the tight link is known or it can be estimated.

Trains of packet pairs (TOPP): TOPP is the canonical example of iterative probing [2, 9]. A probing stream at a rate R_i consists of several packet pairs with that rate. The rate increases linearly in successive streams. TOPP estimates the avail-bw as the maximum input rate R_i that is

not larger than R_o. An interesting point about TOPP is that it attempts to also estimate the capacity of the tight link, as well as the avail-bw and capacities of secondary bottlenecks in the path.

Pathload: Pathload is an estimation tool that we developed, and it is based on iterative probing [3]. A difference with TOPP is that Pathload varies the probing rate in a "binary search" manner, rather than linearly. Pathload was the first tool to consider the variability of the avail-bw process, and this is why it estimates a *variation range* rather than a single estimate. Furthermore, it infers the relation between the input rate and the avail-bw based on statistical analysis of the OWD trends, rather than based on the ratio R_o/R_i. Also, in [3] we considered the effect of the probing stream length on the variation range of the avail-bw, and explained the differences between TCP throughput and avail-bw. We will return to the previous points in §3.

Pathchirp and S-chirp: Pathchirp is based on iterative probing [5]. Instead of sending periodic packet streams, Pathchirp sends "chirps", i.e., N packets with exponentially increasing interarrivals between successive packets. Therefore, the packet pairs of a single chirp probe the network at a wide range of rates, while a single chirp of length N probes $N-1$ different rates. Note that Pathchirp's efficiency in terms of the number of probing rates per packet is due to the use of consecutive packet pairs. Pathchirp reports a single estimate of the avail-bw during a sequence of chirps.

A similar approach, referred to as *Smoothed-chirp* (S-chirp), was proposed in [10] and it is also based on iterative probing.

IGI/PTR: Hu and Steenkiste proposed two estimation techniques (IGI and PTR) in [4]. PTR is quite similar with TOPP, in the sense that it is also based on iterative probing, performs linear rate variation, and reports a single estimate. A difference with TOPP is that it uses packet trains of 60 packets rather than packet pairs.

IGI is harder to classify. It uses an equation that is similar to (9), and so it can be viewed as direct probing. The input rate R_i, however, is chosen in an iterative manner, as in TOPP and PTR. It is not clear what is the benefit of also using the direct probing equation, given that at the end of the iterative phase IGI already has an estimate of A. IGI also uses packet trains of 60 packets. The capacity of the tight link is estimated using *bprobe*, which is an end-to-end capacity estimation tool [11].

Spruce: Spruce is also based on direct probing [6]. Spruce uses 100 packet pairs, instead of packet trains, to collect its avail-bw samples. The input rate is chosen to be roughly equal to the tight link capacity, which is assumed to be known. To emulate *Poisson sampling*, different packet pairs are spaced with exponential interarrivals.

Bfind: BFind is also based on iterative probing [7]. It differs from previous tools in that it does not require control at both ends of the path but it is based on ICMP TTL-expired responses from the intermediate routers in the path. Specifically, the sender probes the end-to-end path with UDP streams of gradually increasing rate, while monitoring at the same time the RTTs to successive routers with repeated traceroute measurements. An increase in the RTT to a particular link reveals an increasing queue size, which implies that the corresponding input rate is larger than the avail-bw at that link.

3. FALLACIES AND PITFALLS [1]

Pitfall: Ignoring the variability of the avail-bw process.

Suppose that we collect k independent samples of the avail-bw process $A^\tau(t)$. Even if we could measure each sample with perfect accuracy, the variability of $A^\tau(t)$ means that the sample mean $m_A(k)$ will be different than the population mean μ_A. The variance of $m_A(k)$ is

$$\mathrm{Var}[m_A(k)] = \frac{\mathrm{Var}[A^\tau]}{k} \tag{11}$$

where the population variance $\mathrm{Var}[A^\tau]$ of the avail-bw process depends on the averaging time scale as discussed in §1. The fact that the variance of the sample mean decreases with the number of samples, or the fact that the population variance decreases with the time scale τ are basic statistical facts. Unfortunately, the avail-bw estimation literature includes accuracy comparisons between estimation techniques that use a different number of samples, or between estimation techniques that use different averaging time scales.
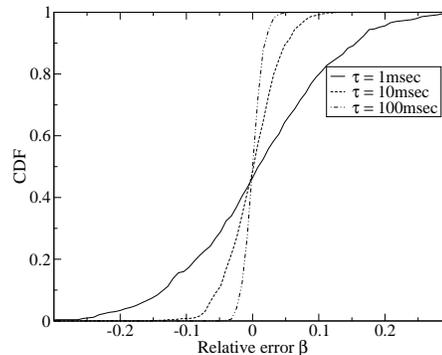

Figure 1: Relative error β of the sample mean m_A for three averaging time scales.

To illustrate the magnitude of the errors that can be caused simply due to sampling, even with perfect measurement of each sample, we analyze an NLANR packet trace from an internet link[2]. The trace provides us with accurate knowledge of the avail-bw process in a wide range of time scales τ. Our experiment consists of collecting sets of $k=20$ avail-bw samples from the trace, using Poisson sampling, calculating the sample mean m_A, and then computing the relative error $\beta = \frac{m_a - \mu_A}{\mu_A}$, where μ_A is the mean avail-bw in the trace. Figure 1 shows the CDF of β for three averaging time scales. Note that unless if τ is 10ms or more, significant errors should be expected with 20 samples, simply due to the variability of the avail-bw process. Especially in short time scales, say 1ms, the number of samples that is required for a reasonably low β (say $< 5\%$) can be in hundreds.

Pitfall: Ignoring the relation between probing stream duration and averaging time scale.

The averaging time scale τ is an important parameter in avail-bw estimation, and we should expect that different ap-

plications will be interested in different values of τ. With the exception of [3], the intimate relation between τ and the probing stream duration has been ignored.
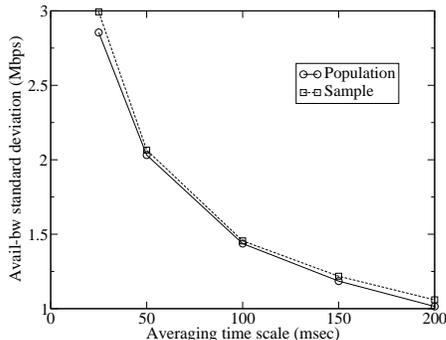

Figure 2: The probing stream duration controls the averaging time scale τ.

It is important to understand that we can control the averaging time scale with the probing stream duration, since the latter determines the length of the time interval in which we interact with the cross traffic at the tight link. To illustrate, we performed a single-hop simulation of direct probing. The capacity of the link is 50Mbps, the average avail-bw is 25Mbps, the cross traffic is Poisson, and the input probing rate is 40Mbps. An avail-bw estimate is obtained using direct probing with each stream. The simulations are repeated five times, for different values of the stream duration (25ms, 50ms, 100ms, 150ms, and 200ms). We then compare the standard deviation of 100 avail-bw samples with the population standard deviation (derived from the simulation packet trace) for the corresponding averaging time scale. Figure 2 shows that the population and sample standard deviations are almost equal. In summary, the probing duration should not be viewed as an "implementation parameter", but as the knob that controls the averaging time scale.

Fallacy: Faster estimation is better.

Some avail-bw estimation tools have been proposed on the premise that they are faster than other tools. Using fewer streams or shorter streams, however, reduces the estimation latency with a penalty in terms of accuracy. For instance, decreasing the probing stream duration decreases the averaging time scale. Since the variance of the avail-bw process increases with a smaller averaging time scale, it is clear from (11) that the variance of the avail-bw sample mean would be increased (for the same number of samples). In general, the stream duration and the number of streams should be inputs to an estimation technique, as knobs that control the estimation accuracy and the measurement overhead/intrusiveness. Additionally, comparisons between different techniques should take into account the tradeoff between estimation latency and accuracy.

Fallacy: Packet pairs are as good as packet trains.

It is true that with fluid cross traffic there is no difference between using packet pairs or longer trains. In practice, however, the cross traffic consists of packets with discrete packet sizes, commonly following a strongly modal distribution. Consequently, the amount of cross traffic that interferes between each probing packet pair can often take

just some discrete values, such as one 1500B packet, two 40B packets, etc. For a given number of samples, and for a given avail-bw process, the avail-bw estimation error will be higher when the cross traffic consists of a few large packets rather than many small packets. In the extreme, if we could reduce the cross traffic packet size L_c to almost zero, the traffic would behave as a fluid model.

	$k{=}10$	$k{=}20$	$k{=}50$	$k{=}100$
$L_c{=}40$B	≈ 0	≈ 0	≈ 0	≈ 0
$L_c{=}512$B	31%	8%	5%	2.5%
$L_c{=}1500$B	40%	20%	8%	2%

Table 1: Effect of cross traffic packet size L_c on the relative error β for four sample sizes k.

To illustrate this point, we performed single-hop simulations (with the same topology and parameters as in the previous paragraph) with three different values of L_c, while keeping the average avail-bw constant. The probing packet size is $L{=}1500$B. Table 1 shows the relative error β for four sample sizes. Even though packet pairs are able to provide good accuracy when the cross traffic consists of small packets (40B), the presence of larger packets makes packet pairs significantly inaccurate.

Pitfall: Estimating the tight link capacity with end-to-end capacity estimation tools.

As previously mentioned, direct probing techniques require the knowledge of the tight link capacity C_t. It is often assumed that C_t can be estimated with end-to-end capacity estimation tools. The latter, however, would estimate the capacity C_n of the narrow link, which may be less than C_t. This is often the case, for instance, when C_n is limited by a Fast Ethernet interface while the tight link is an OC-3 or OC-12 link that has less avail-bw than the narrow link.

Pitfall: Ignoring the effects of cross traffic burstiness.

With the fluid model of §1, we have that $R_o < R_i$ if and only if $R_i > A$. Due to the cross traffic burstiness, however, a queue can build up at the tight link during the probing stream even if $R_i < A$. This should not be a surprise. We know from basic queuing theory that queues build up even before a server is 100% utilized, depending on the burstiness (variance) and correlation structure of the arriving load.
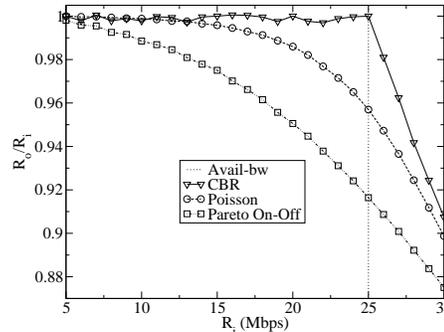

Figure 3: Effect of cross traffic burstiness.

Returning to the previous single-hop simulation setting, we measured the ratio R_o/R_i as a function of R_i for three different cross traffic models: Constant-Bit-Rate (periodic),

Poisson, and Pareto ON-OFF[3]. The mean avail-bw is set to 25Mbps. Figure 3 shows the resulting average R_o/R_i ratio from 500 samples. With CBR traffic, we are very close to the fluid model and so the ratio R_o/R_i drops to less than 1.0 only after $R_i > A$. With the two other cross traffic models, however, the burstiness of the arriving traffic causes $R_o/R_i < 1$ well before we reach the avail-bw point.

It is clear that the cross traffic burstiness can cause significant underestimation errors in both direct and iterative probing techniques. One may think that the burstiness can be taken into account by using certain thresholds; for instance, to say that $R_i > A$ if $R_o/R_i < 0.96$. These thresholds, however, depend strongly on the measured path and on the cross traffic burstiness, as can be seen comparing the Poisson model with the Pareto ON-OFF model. The fact that traffic burstiness causes underestimation errors in avail-bw estimation has been formally shown recently by Liu *et al.* [12], modeling direct probing in a single-hop scenario.

Pitfall: Ignoring the effects of multiple bottlenecks.

A similar underestimation error can be introduced in the presence of multiple bottlenecks that have approximately equal avail-bw (multiple tight links). The reason is that, in such paths, the more links the probing streams go through, the higher the probability that these streams will interact with cross traffic there, and so the lower their output rate will be.
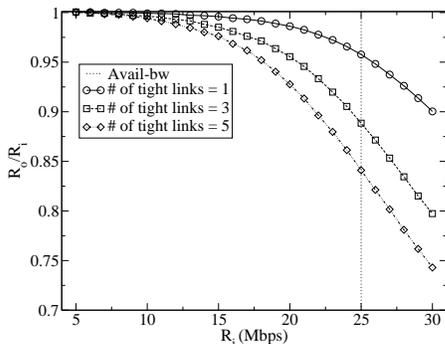
Figure 4: Effect of multiple tight links.

Figure 4 shows simulation results for the average ratio R_o/R_i for 500 samples with Poisson cross traffic. The cross traffic is one-hop persistent, i.e. it enters the link i and exits at link $i+1$. The main observation is that as the number of tight link increases, the ratio R_o/R_i at the point $R_i = A$ decreases.

The underestimation errors caused by both traffic burstiness and multiple bottlenecks can be viewed as artifacts of the simplistic avail-bw definition in (3). If the avail-bw was defined so that it decreases with the burstiness of the traffic, or with the number of tight links, then the previous underestimation errors could potentially be reduced or avoided. For instance, the *effective bandwidth* metric of [13] captures the maximum load in a queue that does not violate a certain delay or loss rate constraint, considering the traffic burstiness. Such different avail-bw definitions, however, would be less practical, because they would involve the characteristics of the traffic and the load at each link of the path.

---

[3]OFF shape parameter=1.5, ON duration uniformly between 1-10 packets.

Fallacy: Increasing One-Way Delays is equivalent to $R_o < R_i$.

In the fluid model, increasing OWDs is equivalent to $R_o < R_i$; see Equations (7) and (8). With real cross traffic, however, the time series of OWD measurements carries much more information than the ratio of the average input and output rates. The reason is rather obvious: the OWDs are many values, and so they can be analyzed with statistical tools to detect trends, measurement noise, level shifts, etc. The ratio R_o/R_i, on the other hand, is a single number, hiding much information about what happened during that probing stream.
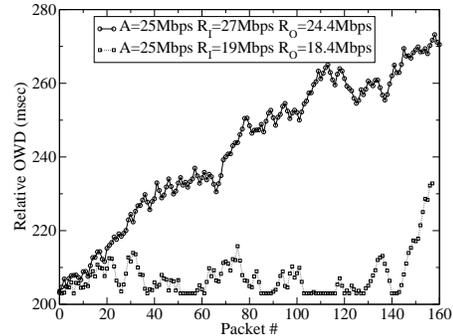
Figure 5: OWDs for two probing streams of 160 packets.

To illustrate, Figure 5 shows the OWDs for two probing streams of 160 packets. In the lower time series, we have that $R_o < R_i$ even though $R_i < A$. The decreased output rate is due to a sudden increase in the OWDs at the very end of the stream (probably due to a cross traffic burst). On the other hand, even a simplistic analysis of the OWDs in that stream would detect that there is no OWD *increasing trend*, inferring correctly that $R_i < A$.

The time series at the top is an example of a clearly increasing OWD trend. In that case, both the OWD trend and the ratio R_o/R_i would correctly infer that $R_i > A$.

Fallacy: Iterative probing converges to a single avail-bw estimate (as opposed to a variation range).

The fact is that iterative probing converges to an avail-bw range rather than to a single value. To see this point, recall that with iterative probing each stream of duration τ and rate $R_i(k)$ can only show if $R_i(k) > A(t, t+\tau)$ or not, where t is the arrival time of the probing stream at the tight link. As we iterate, sending streams with different rates, the avail-bw process $A^\tau(t)$ changes. Consequently, even though a rate $R_i(k)$ may be larger than $A(t_1, t_1 + \tau)$, it may be less than $A(t_2, t_2 + \tau)$ at some later time t_2. Eventually, after probing at several rates, we can only determine a range (R_L^τ, R_H^τ) in which the avail-bw process $A^\tau(t)$ varies with time. Here, R_L^τ is the lowest probing rate that lead to increasing OWDs, while R_H^τ was the highest probing rate that did not lead to increasing OWDs. The range (R_L^τ, R_H^τ) can be viewed as an estimate of the absolute variation range of the process $A^\tau(t)$ during the measurements.

To illustrate, Figure 6 shows a sample path of avail-bw measurements using the previously mentioned packet trace. A (passive) measurement is obtained in every τ=10ms. Note that the avail-bw in that time scale varies, with significant probability, between 60Mbps and 110Mbps. This range can
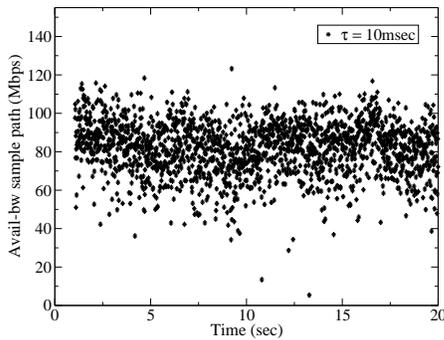
Figure 6: Variation range of an avail-bw sample path.

be estimated with iterative probing, providing us with useful information regarding, not just the average avail-bw, but the range in which the avail-bw process varies in the time scale that corresponds to the stream duration. Pathload is the only tool that reports such a variation range [3]. Unfortunately, the Pathload range is often misinterpreted as a confidence interval, or as a range-estimate for the average avail-bw.

Pitfall: Evaluating the accuracy of avail-bw estimation through comparisons with bulk TCP throughput.

The related literature makes comparisons between the estimated avail-bw and measured bulk TCP throughput, as a way to verify or evaluate avail-bw estimation techniques. These two metrics however are very different, and they should not be expected to be equal. The throughput of a bulk TCP transfer depends on a number of factors, including socket buffer sizes at the sender and receiver, avail-bw, amount of buffering in the tight link, type of competing cross traffic, round-trip time, loss rate, and several more. Even though the avail-bw is one of these factors, it is certainly not the case that it dominates the others.
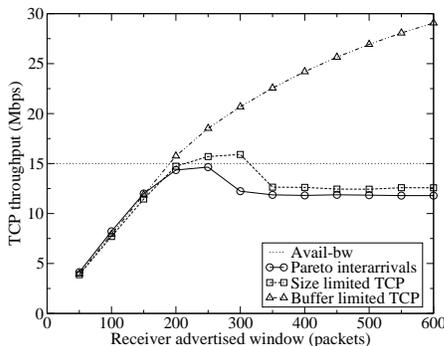

Figure 7: TCP throughput compared to avail-bw.

We next show some simulation results on the effect of the receiver's advertised window W_r, as well as on the effect of the cross traffic *congestion responsiveness* on TCP throughput. Figure 7 shows the throughput of a bulk TCP transfer as a function of W_r for three types of cross traffic. The avail-bw is 15Mbps. The cross traffic types are: UDP sources with Pareto interarrivals, a few persistent TCP transfers limited by their advertised windows, and an aggregate of many short TCP transfers. Note that the difference between the avail-bw and the TCP throughput can be positive or negative, and it strongly depends on the congestion responsiveness of

the cross traffic, and on W_r. An extended study of the relation between avail-bw and TCP throughput can be found in [14].

4. SUMMARY

This note identified a number of issues about avail-bw estimation that need to be examined more thoroughly. An interesting open question is whether some of the problems we presented, such as the underestimation errors that can be caused due to traffic burstiness or multiple bottlenecks, can be addressed through improved estimation techniques or through statistical processing of the resulting delay variations. Another important task is to compare and evaluate the existing estimation techniques under reproducible and controllable conditions, and with the same configuration parameters (e.g., number of streams or probing duration). Finally, it is important to integrate avail-bw estimation techniques with actual applications, and then examine the effectiveness of these techniques given the actual accuracy and latency constraints of real applications.

5. 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 SIGCOMM conference on Internet measurement*, 2003.

[7] A. Akella, S. Seshan, and A. Shaikh, "An empirical evaluation of wide-area internet bottlenecks," in *Proceedings of ACM SIGCOMM conference on Internet measurement*, 2003.

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

[9] B. Melander, M. Bjorkman, and P. Gunningberg, "Regression-Based Available Bandwidth Measurements," in *International Symposium on Performance Evaluation of Computer and Telecommunications Systems*, 2002.

[10] A. Pasztor, *Accurate Active Measurement in the Internet and its Applications*, Ph.D. thesis, Department of Electrical and Electronic Engineering, The University of Melbourne, Feb. 2003.

[11] R. L. Carter and M. E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," *Performance Evaluation*, vol. 27,28, pp. 297–318, 1996.

[12] 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 SIGCOMM conference on Internet measurement*, 2004.

[13] F. Kelly, *Stochastic Networks: Theory and Applications*, chapter Notes on effective bandwidths, pp. 141–168, Oxford University Press, 1996.

[14] M. Jain, R. S. Prasad, and C. Dovrolis, "The TCP Bandwidth-Delay Product Revisited: Network Buffering, Cross Traffic, and Socket Buffer Auto-Sizing," Tech. Rep. GIT-CERCS-03-02, Georgia Tech, Feb. 2003, Available at http://www.cercs.gatech.edu/tech-reports/.