

Packet-Pair Bandwidth Estimation: Stochastic Analysis of a Single Congested Node

¹Seong-ryong Kang, ²Xiliang Liu, ¹Min Dai, and ¹Dmitri Loguinov

¹Texas A&M University, College Station, TX 77843, USA

{skang, dmitri}@cs.tamu.edu, min@ee.tamu.edu

²City University of New York, New York, NY 10016, USA

xliu@gc.cuny.edu

Abstract— In this paper, we examine the problem of estimating the capacity of bottleneck links and available bandwidth of end-to-end paths under *non-negligible* cross-traffic conditions. We present a simple stochastic analysis of the problem in the context of a single congested node and derive several results that allow the construction of asymptotically-accurate bandwidth estimators. We first develop a generic queuing model of an Internet router and solve the estimation problem assuming renewal cross-traffic at the bottleneck link. Noticing that the renewal assumption on Internet flows is too strong, we investigate an alternative filtering solution that asymptotically converges to the desired values of the bottleneck capacity and available bandwidth under arbitrary (including non-stationary) cross-traffic. This is one of the first methods that simultaneously estimates both types of bandwidth and is provably accurate. We finish the paper by discussing the impossibility of a similar estimator for paths with two or more congested routers.

I. INTRODUCTION

Bandwidth estimation has recently become an important and mature area of Internet research [1], [2], [4], [5], [6], [7], [10], [11], [12], [14], [15], [16], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27]. A typical goal of these studies is to understand the characteristics of Internet paths and those of cross-traffic through a variety of end-to-end and/or router-assisted measurements. The proposed techniques usually fall into two categories – those that estimate the bottleneck bandwidth [4], [5], [6], [12], [13] and those that deal with the available bandwidth [11], [19], [24], [26]. Recall that the former bandwidth metric refers to the capacity of the slowest link of the path, while the latter is generally defined as the smallest average unused bandwidth among the routers of an end-to-end path.

The majority of existing bottleneck-bandwidth estimation methods are justified assuming no cross-traffic along the path and are usually examined in simulations/experiments to show that they can work under realistic network conditions [4], [5], [6], [7], [12]. With available bandwidth estimation, cross-traffic is essential and is usually taken into account in the analysis; however, such analysis predominantly assumes a fluid model for all flows and implicitly requires that such models be accurate in non-fluid cases. Simulations/experiments are again used to verify that the proposed methods are capable of dealing with bursty conditions of real Internet cross-traffic [11], [19], [24], [26].

To understand some of the reasons for the lack of stochastic modeling in this field, this paper studies a *single-node* bandwidth measurement problem and derives a closed-form estimator for both capacity C and available bandwidth $A = C - \bar{r}$, where \bar{r} is the average rate of cross-traffic at the link. For an arbitrary cross-traffic arrival process $r(t)$, we define \bar{r} as the asymptotic time-average of $r(t)$ and assume that it exists and is finite:

$$\bar{r} = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t r(u) du < \infty. \quad (1)$$

Notice that the existence of (1) does not require stationarity of cross-traffic, nor does it impose any restrictions on the arrival of individual packets to the router. While other definitions of available bandwidth A and the average cross-traffic rate \bar{r} are possible, we find that (1) serves our purpose well as it provides a clean and mathematically tractable metric.

The first half of the paper deals with bandwidth estimation under *i.i.d.* renewal cross-traffic and the analysis of packet-pair/train methods. We first show that under certain conditions and even the simplest *i.i.d.* cross-traffic, histogram-based methods commonly used in prior work (e.g., [5]) can be misled into producing inaccurate estimates of C . We overcome this limitation by developing an asymptotically accurate model for C ; however, since this approach eventually requires ergodicity of cross-traffic, we later build another model that imposes more restriction on the sampling process (using PASTA principles suggested in [26]), but allows cross-traffic to exhibit arbitrary characteristics.

Unlike previous studies [26], we prove that the corresponding PASTA-based estimators converge to the correct values and show that they can be used to simultaneously measure capacity C and available bandwidth A . To our knowledge, this is the first estimator that measures both C and A without congesting the link, assumes non-negligible, non-fluid cross-traffic in the derivations, and applies to non-stationary $r(t)$. Note that while this estimator can measure A over multiple links, *its inherent purpose is not to become another measurement tool or to work over multi-node paths, but rather to understand the associated stochastic models and disseminate the knowledge obtained in the process of writing this paper.*

We conclude the paper by discussing the impossibility of building an asymptotically optimal estimator of both C and A for a two-node case. While estimation of A may be possible for multi-node paths, our results suggest that provably-convergent estimators of C do not exist in the context of end-to-end measurement. Thus, the problem of deriving a *provably accurate* estimator for a two-node case with arbitrary cross-traffic remains open; however, we hope that our initial work in this direction will stimulate additional research and prompt others to prove/disprove this conjecture.

II. STOCHASTIC QUEUING MODEL

In this section, we build a simple model of a router that introduces random delay noise into the measurements of the receiver and use it to study the performance of packet-pair bandwidth-sampling techniques. Note that we depart from the common assumption of negligible and/or fluid cross-traffic and specifically aim to understand the effects of random queuing delays on the bandwidth sampling process. First consider an unloaded router with no cross-traffic. The packet-pair mechanism is based on an observation that if two packets arrive at the bottleneck link with spacing x smaller than the transmission delay Δ of the second packet over the link, their spacing after the link will be exactly Δ . In practice, however, packets from other flows often queue between the two probe packets and increase their spacing on the exit from the bottleneck link to be larger than Δ .

Assume that packets of the probe traffic arrive to the bottleneck router at times a_1, a_2, \dots and that inter-arrival times $a_n - a_{n-1}$ are given by a random process x_n determined by the server's initial spacing. Further assume that the bottleneck node delays arriving packets by adding a random processing time ω_n to each received packet n . For the remainder of the paper, we use constant¹ packet size q for the probing flow and arbitrarily-varying packet sizes for cross-traffic. Furthermore, there is no strict requirement on the initial spacing x_n as long as the modeling assumptions below are satisfied. This means that both isolated packet pairs or bursty packet trains can be used to probe the path. Let the transmission delay of each application packet through the bottleneck link be $q/C = \Delta$, where C is the transmission capacity of the link. Under these assumptions, packet departure times d_n are expressed by the following recurrence²:

$$d_n = \begin{cases} a_1 + \omega_1 + \Delta & n = 1 \\ \max(a_n, d_{n-1}) + \omega_n + \Delta & n \geq 2 \end{cases} \quad (2)$$

In this formula, the dependence of d_n on departure time d_{n-1} is a consequence of FIFO queuing (i.e., packet n cannot depart before packet $n-1$ is fully transmitted). Furthermore, packet n cannot start transmission until it has fully arrived (i.e., time a_n). The value of the noise term ω_n is proportional to the number of packets generated by cross-traffic and queued

¹Methods that vary the probing packet size also exist (e.g., [6], [10]).

²Times d_n specify when the *last* bit of the packet leaves the router. Similarly, times a_n specify when the last bit of the packet is fully received and the packet is ready for queuing.

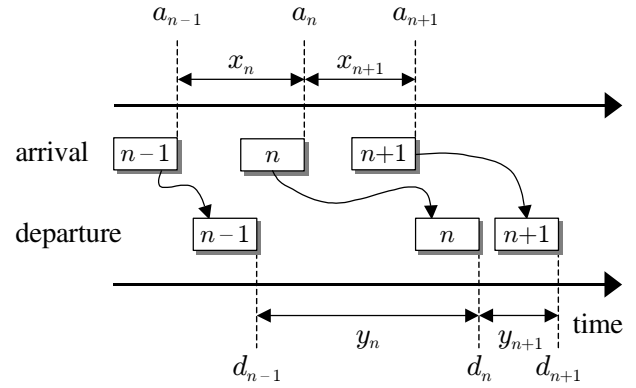


Fig. 1. Departure delays introduced by the node.

in front of packet n . The final metric of interest is inter-departure delay $y_n = d_n - d_{n-1}$ as the packets leave the router. The various variables and packet arrival/departure are schematically shown in Figure 1.

Even though the model in (2) appears to be simple, it leads to fairly complicated distributions for y_n if we make no prior assumptions about cross-traffic. We next examine several special cases and derive important asymptotic results about process y_n .

III. RENEWAL CROSS-TRAFFIC

A. Packet-Pair Analysis

We start our analysis with a rather common assumption in queuing theory that cross-traffic arrives into the bottleneck link according to some renewal process (i.e., delays between cross-traffic packets are *i.i.d.* random variables). In what follows in the next few subsections, we show that modeling of this direction requires stationarity (more specifically, ergodicity) of cross-traffic. However, since neither the *i.i.d.* assumption nor stationarity holds for regular Internet traffic, we then apply a different sampling methodology and a different analytical direction to derive a provably robust estimator of capacity C and average cross-traffic rate \bar{r} .

The goal of bottleneck bandwidth sampling techniques is to queue probe packets directly behind each other at the bottleneck link and ensure that spacing y_n on the exit from the router is Δ . In practice, however, this is rarely possible when the rate of cross-traffic is non-negligible. This does present certain difficulties to the estimation process; however, assuming a single congested node, the problem is asymptotically tractable given certain mild conditions on cross-traffic. We present these results below.

To generate measurements of the bottleneck capacity, it is commonly derived that the server must send its packets with initial spacing no more than Δ (i.e., no slower than C). This is true for *unloaded* links; however, when cross-traffic is present, the probes may be sent arbitrarily slower as long as each packet i arrives to the router before the departure time of packet $i-1$. This condition translates into $a_n \leq d_{n-1}$ and

(2) expands to:

$$d_n = \begin{cases} a_1 + \omega_1 + \Delta & n = 1 \\ d_{n-1} + \omega_n + \Delta & n \geq 2 \end{cases} \quad (3)$$

From (3), packet inter-departure times y_n after the bottleneck router are given by:

$$y_n = d_n - d_{n-1} = \Delta + \omega_n, \quad n \geq 2. \quad (4)$$

Notice that random process ω_n is defined by the arrival pattern of cross-traffic and also by its packet-size distribution. Since this process is a key factor that determines the distribution of sampled delays y_n , we next focus on analyzing its properties. Assume that inter-packet delays of cross-traffic are given by independent random variables $\{X_i\}$ and the actual arrivals occur at times $X_1, X_1 + X_2, \dots$. Thus, the arrival pattern of cross-traffic defines a renewal process $M(t)$, which is the number of packet arrivals in the interval $[0, t]$. Using common convention, further assume that the mean inter-arrival delay $E[X_i]$ is given by $1/\lambda$, where $\lambda = \bar{r}$ is the mean arrival rate of cross-traffic in packets per second.

To allow random packet sizes, assume that $\{S_j\}, j = 1, 2, \dots$ are independent random variables modeling the size of packets in cross-traffic. We further assume that the bottleneck link is probed by a sequence of packet-pairs, in which the delay between the packets *within each pair* is small (so as to keep their rate higher than C) and the delay *between the pairs* is high (so as not to congest the link). Under these assumptions, the amount of cross-traffic data received by the bottleneck link between probe packets $2m - 1$ and $2m$ (i.e., in the interval (a_{2m-1}, a_{2m})) is given by a cumulative reward process:

$$v_n = \sum_{j=1}^{M(a_n) - M(a_{n-1})} S_j, \quad n = 2m, \quad (5)$$

where n represents the sequence number of the second packet in each pair. For now, we assume a general (not necessarily equilibrium) process $M(t)$ and re-write (4) as:

$$y_n = \Delta + \frac{v_n}{C} = \Delta + \frac{\sum_{j=1}^{M(a_n) - M(a_{n-1})} S_j}{C}, \quad n = 2m. \quad (6)$$

Since classical renewal theory is mostly concerned with limiting distributions, y_n by itself does not lead to any tractable results because the observation period of the process captured by each of y_n is very small.

Define a time-average process W_n to be the average of $\{y_i\}$ up to time n :

$$W_n = \frac{1}{n} \sum_{i=1}^n y_i. \quad (7)$$

Then, we have the following result.

Claim 1: Assuming ergodic cross-traffic, time-average process W_n converges to:

$$\lim_{n \rightarrow \infty} W_n = \Delta + \frac{\lambda E[x_n] E[S_j]}{C} = \Delta + E[\omega_n], \quad (8)$$

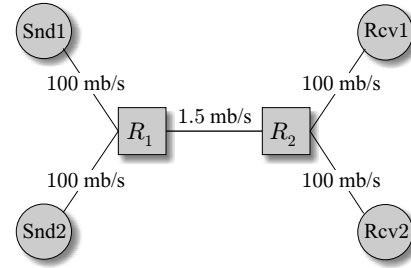


Fig. 2. Single-link simulation topology.

where $E[x_n]$ is the mean inter-probe delay of packet-pairs.

Proof: Process W_n samples a much larger span of $M(t)$ and has a limiting distribution as we demonstrate below. Applying Wald's equation to (6) [28]:

$$E[y_n] = \Delta + \frac{E[M(a_n) - M(a_{n-1})] E[S_j]}{C}. \quad (9)$$

The last result holds since $M(t)$ and S_j are independent and $M(a_n) - M(a_{n-1})$ is a stopping time for sequence $\{S_j\}$. Equation (9) can be further simplified by noticing that $E[M(t)]$ is the renewal function $m(t)$:

$$E[y_n] = \Delta + \frac{(m(a_n) - m(a_{n-1})) E[S_j]}{C}. \quad (10)$$

Assuming stationary cross-traffic, (10) expands to [28]:

$$E[y_n] = \Delta + \frac{\lambda E[x_n] E[S_j]}{C}. \quad (11)$$

Finally, assuming ergodicity of cross-traffic (which implies that of process y_n), we can obtain (11) using a large number of packet pairs as the limit of W_n in (7) as $n \rightarrow \infty$. ■

Notice that the second term in (8) is strictly positive under the assumptions of this paper. This leads to an interesting observation that the filtering problem we are facing is quite challenging since the sampled process y_n represents signal Δ corrupted by a *non-zero-mean* noise ω_n . This is a drastic departure from the classical filter theory, which mostly deals with zero-mean additive noise. It is also interesting that the only way to make the noise zero-mean is to either send probe traffic with $E[x_n] = 0$ (i.e., infinitely fast) or to have no cross-traffic at the bottleneck link (i.e., $\lambda = \bar{r} = 0$). The former case is impossible since x_n is always positive and the latter case is a simplification that we explicitly want to avoid in this work.

We will present our analysis of packet-train probing shortly, but in the mean time, discuss several simulations to provide an intuitive explanation of the results obtained so far.

B. Simulations

Before we proceed to estimation of C , let us explain several observations made in previous work and put them in the context of our model in (6) and (8). For the simulations in this section, we used the ns2 network simulator with the topology shown in Fig. 2. In the figure, the source of probe packets Snd1 sends its data to receiver Rcv1 across two routers R_1 and R_2 . The speed of all access links is 100 mb/s (delay 5 ms),

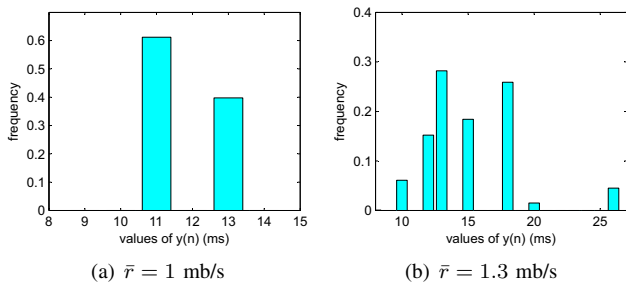


Fig. 3. The histogram of measured inter-arrival times y_n under CBR cross-traffic.

while the bottleneck link $R_1 \rightarrow R_2$ has capacity $C = 1.5$ mb/s and 20 ms delay. Note that there are five cross-traffic sources attached to Snd2.

We next discuss simulation results obtained under UDP cross-traffic. In the first case, we initialize all five sources in Snd2 to be CBR streams, each transmitting at 200 kb/s ($\bar{r} = 1$ mb/s total cross-traffic). Each CBR flow starts with a random initial delay to prevent synchronization with other flows and uses 500-byte packets. The probe flow at Snd1 sends its data at an average rate of 500 kb/s for the probing duration, which results in 100% utilization of the bottleneck link. In the second case, we lower packet size of cross-traffic to 300 bytes and increase its total rate to 1.3 mb/s to demonstrate more challenging scenarios when there is packet loss at the bottleneck.

These simulation results are summarized in Fig. 3, which illustrates the distribution of the measured samples y_n based on each pair of packets sent with spacing $x_n \leq \Delta$ (the results exclude packet pairs that experienced loss). Given capacity $C = 1.5$ mb/s and packet size $q = 1,500$ bytes, the value of Δ is 8 ms. Fig. 3 shows that none of the samples are located at the correct value of 8 ms and that the mean of the sampled signal (i.e., W_n) has shifted to 11.7 ms for the first case and 14.5 ms for the second one.

Next, we employ TCP cross-traffic, which is generated by five FTP sources attached to Snd2. The TCP flows use different packet sizes of 540, 640, 840, 1,040, and 1,240 bytes, respectively. The histogram of y_n for this case is shown in Fig. 4 for two different average cross-traffic rates $\bar{r} = 750$ kb/s and $\bar{r} = 1$ mb/s. As seen in the figure, even though some of the samples are located at 8 ms, the majority of the mass in the histogram (including the peak modes) are located at the values much higher than 8 ms.

Recall from (8) that W_n of the measured signal tends to $\Delta + E[\omega_n]$. Under CBR cross-traffic, we can estimate the mean of the noise $E[\omega_n]$ to be approximately $11.7 - 8 = 3.7$ ms in the first case and $14.5 - 8 = 6.5$ ms in the second one. The two naive estimates of C based on W_n are $\tilde{C} = q/W_n = 1,025$ kb/s and $\tilde{C} = 827$ kb/s, respectively. Likewise, for the TCP case, the measured averages (i.e., 12.2 and 13.3 ms each) of samples y_n lead to incorrect naive estimates $\tilde{C} = 983$ kb/s and $\tilde{C} = 902$ kb/s, respectively.

In order to understand how \tilde{C} and the value of W_n evolve,

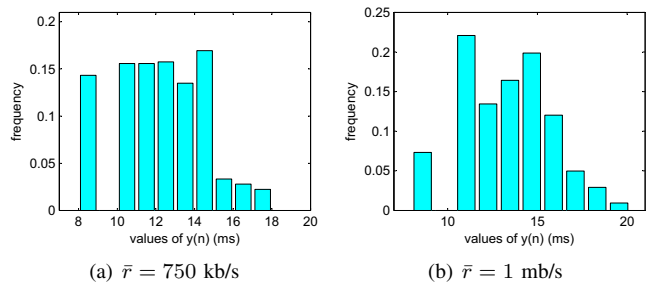


Fig. 4. The histogram of measured inter-arrival times y_n under TCP cross-traffic.

we run another simulation under 1 mb/s total TCP cross-traffic and plot the evolutions of the absolute error $|\tilde{C} - C|$ and that of W_n in Fig. 5. As Fig. 5(a) shows, the absolute error between C and \tilde{C} converges to a certain value after 5,000 samples y_n , providing a rather poor estimate $\tilde{C} \approx 1,010$ kb/s. Fig. 5(b) illustrates that W_n in fact converges to $\Delta + E[\omega_n]$, where the mean of the noise is $W_n - \Delta \approx 11.9 - 8 = 3.9$ ms.

Previous work [1], [4], [5], [15], [22] focused on identifying the peaks (modes) in the histogram of the collected bandwidth samples and used these peaks to estimate the bandwidth; however, as Figs. 3 and 4 show, this can be misleading when the distribution of the noise is not known a-priori. For example, the tallest peak on the right side of Fig. 3 is located at 13 ms ($\tilde{C} = 923$ kb/s), which is only a slightly better estimate than 827 kb/s derived from the mean of y_n . Moreover, the tallest peak in Fig. 4(a) is located at 14.5 ms, which leads to a worse estimate $\tilde{C} = 827$ kb/s compared to 983 kb/s computed from the mean of y_n .

To combat these problems, existing studies [1], [4], [5], [15], [22] apply numerous empirical methods to find out which mode is more likely to be correct. This may be the only feasible solution in multi-hop networks; however, one must keep in mind that it is possible that *none* of the modes in the measured histogram corresponds to Δ as evidenced by both graphs in Fig. 3.

C. Packet-Train Analysis

Another topic of debate in prior work was whether *packet-train* methods offer any benefits over *packet-pair* methods. Some studies suggested that packet-train measurements converge to the *available* bandwidth³ for sufficiently long bursts of packets [1], [4]; however, no analytical evidence to this effect has been presented so far. Other studies [5] employed packet-train estimates to increase the measurement accuracy of bottleneck bandwidth estimation, but it is not clear how these samples benefit asymptotic convergence of the estimation process.

We consider a hypothetical packet-train method that transmits probe traffic in bursts of k packets and averages the inter-packet arrival delays within each burst to obtain individual

³Even though this question appears to have been settled in some of the recent papers, we provide additional insight into this issue.

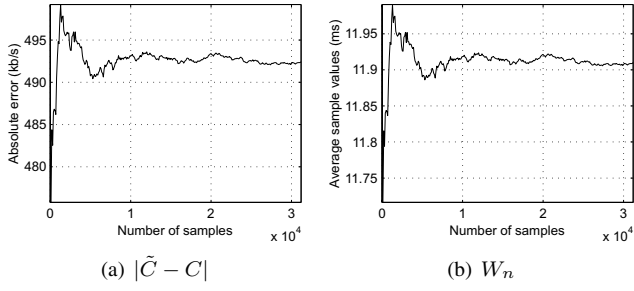


Fig. 5. (a) The absolute error of the packet-pair estimate under TCP cross-traffic of $\bar{r} = 1$ mb/s. (b) Evolution of W_n .

samples $\{Z_n^k\}$, where n is the burst number. For example, if $k = 10$, samples y_2, \dots, y_{10} define Z_1^{10} , samples y_{12}, \dots, y_{20} define Z_2^{10} , and so on. The reason for excluding samples $y_1, y_{n+1}, \dots, y_{nk+1}$ is because they are based on the leading packets of each burst, which encounter large inter-burst gaps in front of them and do not follow the model developed so far.

In what follows in this section, we derive the distribution of $\{Z_n^k\}$ as $k \rightarrow \infty$.

Claim 2: For sufficiently large k , constant $x_n = x$, and a regenerative processes $M(t)$, packet-train samples converge to the following Gaussian distribution for large n :

$$\{Z_n^k\} \xrightarrow{D} N\left(\Delta + \frac{\lambda x E[S_j]}{C}, \frac{\lambda x \text{Var}[S_j - \lambda E[S_j]X_i]}{(k-1)C^2}\right), \quad (12)$$

where \xrightarrow{D} denotes convergence in distribution, $N(\mu, \sigma)$ is a Gaussian distribution with mean μ and standard deviation σ , and X_i are inter-packet arrival delays of cross-traffic.

Proof: First, define a k -sample version of the cumulative reward process in (5):

$$V_n^k = \sum_{j=1}^{M(a_{kn}) - M(a_{k(n-1)+1})} S_j, \quad n = 1, 2, \dots \quad (13)$$

Process V_n^k is also a counting process, however, its time-scale is measured in bursts instead of packets. Thus, V_n^k determines the amount of cross-traffic data received by the bottleneck link during an entire burst n . Equation (13) shows that Z_n^k can be asymptotically interpreted as the *reward rate* of the reward-renewal process V_n^k :

$$Z_n^k = \Delta + \frac{V_n^k}{(k-1)C}, \quad (14)$$

where $k-1$ is the number of inter-packet gaps in a k -packet train of probe packets. Assuming $M(t)$ is regenerative and for sufficiently large k , we have [28]:

$$Z_n^k = \Delta + \frac{V_1^k}{(k-1)C} + o(1). \quad (15)$$

Applying the regenerative central limit theorem, constraining the rest of the derivations in this section to constant

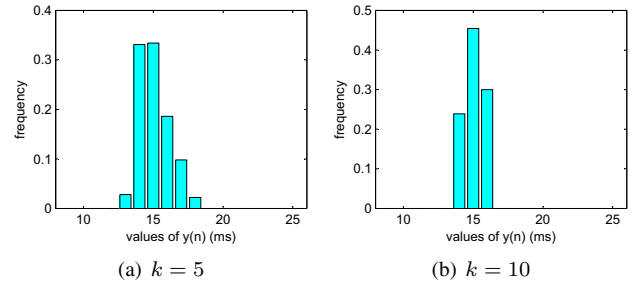


Fig. 6. The histogram of measured inter-arrival times Z_n^k based on packet trains and CBR cross-traffic of $\bar{r} = 1.3$ mb/s.

$x_n = x$, and assuming $E[X_i] < \infty$ [28]:

$$\left\{ \frac{V_1^k}{k-1} \right\} \xrightarrow{D} N\left(\lambda x E[S_j], \frac{\lambda x \text{Var}[S_j - \lambda E[S_j]X_i]}{k-1}\right). \quad (16)$$

Combining (15) and (16), we get (12). ■

First, notice that the mean of this distribution is the same as that of samples $\{y_n\}$ in (11), which, as was intuitively expected, means that both measurement methods have the same expectation. Second, it is also easy to notice that the variance of Z_n^k tends to zero as long as $\text{Var}[X_i]$ is finite.

Claim 3: If $\text{Var}[X_i]$ is finite, the variance of packet-train samples Z_n^k tends to zero for large k .

Proof: Since λ , x , and $E[S_j]$ are all finite and do not depend on k , using independence of S_j and X_i in (12), we have:

$$\text{Var}[Z_n^k] = \left(\frac{\lambda x \text{Var}[S_j] + \lambda^3 x E^2[S_j] \text{Var}[X_i]}{(k-1)C^2} \right)^2, \quad (17)$$

which tends to 0 for $k \rightarrow \infty$. ■

As a result of this phenomenon, longer packet trains will produce narrower distributions centered at $\Delta + E[\omega_n]$. The CBR case already studied in Fig. 3(b) clearly has finite $\text{Var}[X_i]$ and therefore samples $\{Z_n^k\}$ must exhibit decaying variance as k increases. One example of this convergence for packet trains with $k = 5$ and $k = 10$ is shown in Fig. 6.

D. Discussion

Now we address several observations of previous work. It is noted in [5] that while packet-pair histograms usually have many different modes, the histogram of packet-train samples becomes unimodal with increased k . This readily follows from (12) and the Gaussian shape of $\{Z_n^k\}$. Previous papers also noted (e.g., [5]) that as packet-train size k is increased, the distribution of samples $\{Z_n^k\}$ exhibits lower variance. This result follows from the above discussion and (17). Furthermore, [5] found that packet-train histograms for large k tend to a single mode whose location is “independent of burst size k .” Our derivations provide an insight into how this process happens and shows the location of this “single mode” to be $\Delta + E[\omega_n]$ in (12).

In summary, packet-train samples $\{Z_n^k\}$ represent a limited reward rate that asymptotically converges to a Gaussian distribution with mean $E[y_n]$. Perhaps it is possible to infer some

characteristics of $\{X_i\}$ by observing the variance of $\{Z_n^k\}$ and applying the result in (12); however, since there are several unknown parameters in the formula (such as $Var[X_i]$ and $E[S_j]$), this direction does not lead to any tractable results unless we assume a particular process $M(t)$.

Since $\{Z_n^k\}$ asymptotically tend to a very narrow Gaussian distribution centered at $\Delta + E[\omega_n]$, we find that there is no evidence that $\{Z_n^k\}$ measure the available bandwidth or offer any additional information about the value of Δ as compared to traditional packet-pair samples $\{y_n\}$.

IV. ARBITRARY CROSS-TRAFFIC

In this section, we relax the stationarity and renewal assumptions about cross-traffic and derive a robust estimator of C and \bar{r} . Assume an arbitrary arrival process $r(t)$ for cross-traffic, where $r(t)$ is its instantaneous rate at time t . We impose only one constraint on this process – it must have a finite time average \bar{r} shown in (1). The goal of the sampling process is to determine both C and \bar{r} . Since $\bar{r} > C$ imply constant packet loss and zero available bandwidth, we are generally interested in non-trivial cases of $\bar{r} \leq C$. Stochastic process $r(t)$ may be renewal, regenerative, a superposition of ON/OFF sources, self-similar, or otherwise. Furthermore, since packet arrival patterns in the current Internet commonly exhibit non-stationarity (due to day-night cycles, routing changes, link failure, etc.), our assumptions on $r(t)$ allow us to model a wide variety of such non-stationary processes and are much broader than commonly assumed in traffic modeling literature.

Next, notice that if the probing traffic can sample $r(t)$ using a Poisson sequence of probes at times t_1, t_2, \dots , the average of $r(t_i)$ converges to \bar{r} (applying the PASTA principle [28]):

$$\lim_{n \rightarrow \infty} \frac{r(t_1) + r(t_2) + \dots + r(t_n)}{n} = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t r(u) du = \bar{r}, \quad (18)$$

as long as delays $\tau_i = t_i - t_{i-1}$ are *i.i.d.* exponential random variables. In order to accomplish this type of sampling, the sender must emit packet-pairs at exponentially distributed intervals. Assuming that the i -th packet-pair arrives to the router at time t_i , it will sample a small segment of $r(t)$ by allowing g_i amount of data to be queued between the probes:

$$g_i = \int_{t_i}^{t_i + x_i} r(u) du \approx r(t_i) x_i, \quad (19)$$

where x_i is the spacing between the packets in the i -th packet-pair. Again, assuming that y_i is the i -th inter-arrival sample generated by the receiver, we have:

$$y_i = \Delta + \frac{g_i}{C} = \Delta + \frac{r(t_i) x_i}{C}. \quad (20)$$

Finally, fixing the value of $x_i = x$, notice that W_n has a

well-defined limit:

$$\begin{aligned} \lim_{n \rightarrow \infty} W_n &= \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \left(\Delta + \frac{r(t_i) x_i}{C} \right) \\ &= \Delta + \frac{x}{C} \lim_{n \rightarrow \infty} \sum_{i=1}^n \frac{r(t_i)}{n} = \Delta + \frac{x \bar{r}}{C}. \end{aligned} \quad (21)$$

In essence, this result⁴ is similar to our earlier derivations, except that (21) requires much weaker restrictions on cross-traffic and also shows that a single-node model is completely tractable in the setting of almost arbitrary cross-traffic. We next show how to extract both C and \bar{r} from (21).

A. Capacity

Observe that (21) is a linear function of x , where \bar{r} is the slope and Δ is the intercept⁵. Therefore, by injecting packet-pairs with two different spacings x_a and x_b , one can compute the unknown terms in (21) using two sets of measurements $\{y_i^a\}$ and $\{y_i^b\}$. To accomplish this, define the corresponding average processes to be W_n^a and W_n^b :

$$W_n^a = \frac{1}{n} \sum_{i=1}^n y_i^a, \quad W_n^b = \frac{1}{n} \sum_{i=1}^n y_i^b. \quad (22)$$

The simplest way to obtain both W_n^a and W_n^b using a *single* measurement is to alternate spacing x_a and x_b while preserving the PASTA sampling property. Using a one-bit header field, the receiver can unambiguously sort the inter-arrival delays into two sets $\{y_i^a\}$ and $\{y_i^b\}$, and thus compute their averages in (22).

While samples are being collected, the receiver has two running averages produced by (22). Subtracting W_n^b from W_n^a , we are able to separate \bar{r}/C from Δ :

$$\lim_{n \rightarrow \infty} (W_n^a - W_n^b) = \frac{(x_a - x_b) \bar{r}}{C}. \quad (23)$$

Next, denote by $\tilde{\Delta}_n$ the following estimate of Δ at time n :

$$\tilde{\Delta}_n = W_n^a - x_a \frac{W_n^a - W_n^b}{x_a - x_b}. \quad (24)$$

Taking the limit of (24), we have the following result.

Claim 4: Assuming a single congested bottleneck for which time-average rate \bar{r} exists, $\tilde{\Delta}_n$ converges to Δ :

$$\lim_{n \rightarrow \infty} \tilde{\Delta}_n = \Delta. \quad (25)$$

Proof: Re-writing (25):

$$\lim_{n \rightarrow \infty} \tilde{\Delta}_n = \Delta + \frac{x_a \bar{r}}{C} - x_a \frac{(x_a - x_b) \bar{r}}{C(x_a - x_b)} = \Delta, \quad (26)$$

which is obtained with the help of (21), (23), and (24). ■

Our next result shows a more friendly restatement of the previous claim.

⁴A similar formula has been derived in [3], [8], [19] and several other papers under a fluid assumption.

⁵For technical differences between this approach and previous work (such as TOPP [19]), see [17].

TABLE I
AVAILABLE BANDWIDTH ESTIMATION ERROR

Bottleneck capacity C (mb/s)	Relative error			
	Model (29)	Pathload	Spruce	IGI
1.5	8.6%	46.5%	27.9%	84.5%
5	8.3%	40.1%	23.4%	90.0%
10	10.1%	40.9%	26.9%	89.0%
15	7.7%	38.5%	24.5%	83.1%

Corollary 1: Assuming a single congested bottleneck for which time-average rate \bar{r} exists, estimate $\tilde{C}_n = q/\tilde{\Delta}_n$ converges to capacity C :

$$\begin{aligned} \lim_{n \rightarrow \infty} \tilde{C}_n &= \lim_{n \rightarrow \infty} \frac{q}{W_n^a - x_a \frac{W_n^a - W_n^b}{x_a - x_b}} \\ &= \lim_{n \rightarrow \infty} \frac{q(x_a - x_b)}{x_a W_n^b - x_b W_n^a} = C. \end{aligned} \quad (27)$$

B. Available Bandwidth

Notice that knowing an estimate of C in (27) and using \bar{r} in (23), it is easy to estimate the mean rate of cross-traffic:

$$\lim_{n \rightarrow \infty} \frac{(W_n^a - W_n^b)\tilde{C}_n}{x_a - x_b} = \bar{r}, \quad (28)$$

which leads to the following result.

Corollary 2: Assuming a single congested bottleneck for which time-average rate \bar{r} exists, the following converges to the available bandwidth $A = C - \bar{r}$:

$$\lim_{n \rightarrow \infty} q \left(\frac{x_a - x_b - W_n^a + W_n^b}{x_a W_n^b - x_b W_n^a} \right) = C - \bar{r} = A. \quad (29)$$

C. Simulations

We confirm these results and compare our models with several recent methods `spruce` [26], IGI [8], and `pathload` [12] through `ns2` simulations. Since the main theme of this paper is bandwidth estimation in heavily-congested routers, we conduct all simulations over a loaded bottleneck link in Fig. 2 with utilization varying between 82% and 92% (the exact value changes depending on C and the interaction of TCP cross-traffic with probe packets). Delays x_a and x_b are set to maintain the desired range of link utilization.

Define $E_A = |\tilde{A} - A|/A$ and $E_C = |\tilde{C} - C|/C$ to be the relative estimation errors of A and C , respectively, where A is the true available bandwidth of a path, \tilde{A} is its estimate using one of the measurement techniques, C is the true bottleneck capacity, and \tilde{C} is its estimate. Table I shows relative estimation errors E_A for `spruce`, IGI, and `pathload`. For `pathload`, we averaged the low and high values of the produced estimates \tilde{A} . In the IGI case, we used the estimates available at the end of IGI's internal convergence algorithm. Also note that we fed both `spruce` and IGI the exact bottleneck capacity C , while model (29) and `pathload` operated without this information.

As the table shows, `spruce` performs better than `pathload` in heavily-congested cases, which is expected

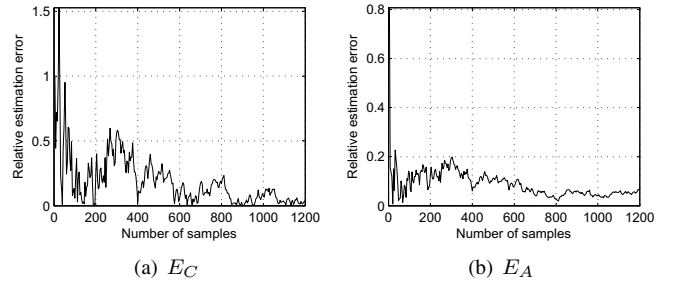


Fig. 7. Evolution of relative estimation errors of (27) and (29) over a single congested link with $C = 1.5$ mb/s and 85% link utilization.

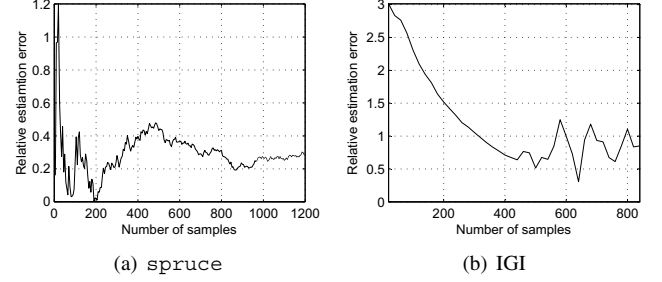


Fig. 8. Relative estimation errors E_A produced by `spruce` and IGI over a single congested link with $C = 1.5$ mb/s and 85% link utilization.

since it can utilize the known capacity information. Interestingly, however, IGI's estimates are worse than those of `pathload` even though IGI utilizes the true capacity C in its estimation algorithm. A similar result is observed in [26] under a relatively small amount of cross-traffic (20% to 40% link utilization).

Next, we examine models (27), (29) with a large number of samples to show their asymptotic convergence and estimation accuracy. We plot the evolution of relative estimation errors E_C and E_A in Fig. 7. As Fig. 7(a) shows, \tilde{C} converges to a value that is very close (within 3%) to the true value of C . In Fig. 7(b), the available bandwidth estimates quickly converge within 10% of A . For the purpose of comparison, we next plot estimation errors E_A produced by `spruce` and IGI in Fig. 8. As Fig. 8(a) shows, even with the exact value of C and after 1,000 samples, `spruce` exhibits an error of 27%. Furthermore, IGI's estimates are much worse than `spruce`'s as illustrated in Fig. 8(b), which plots the evolution of errors until IGI's internal algorithm terminates.

To better understand these results, we next study how the accuracy of capacity information provided to `spruce` and IGI affects their measurement accuracy.

D. Further Analysis of `spruce` and IGI

Since bottleneck capacities of Internet paths are not generally known, the use of `spruce` and IGI may be limited to a small number of known paths, unless these methods can obtain capacity measurements from other tools such as `nettimer` [15], [16], `pathrate` [5], or `CapProbe` [13]. These estimators of C are usually very accurate when the routers along the path are not highly utilized; however, as link utilization increases,

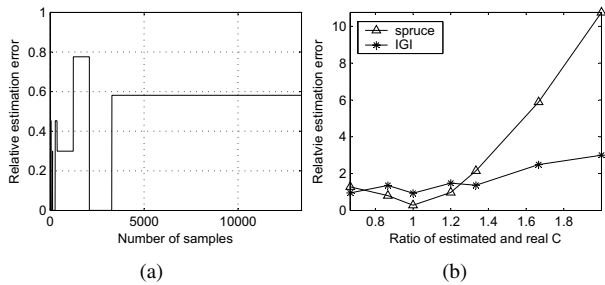


Fig. 9. (a) Relative capacity estimation error E_C measured by CapProbe. (b) Relative estimation errors E_A of *spruce* and IGI based on the ratio \tilde{C}/C .

their results often become inaccurate. Hence, a natural question is *how robust are spruce and IGI to inaccurate values of C provided to their estimation process?* We examine this issue below, but first discuss simulation results of CapProbe to illustrate the pitfalls that many estimators of C experience over heavily-loaded bottleneck links.

For the CapProbe simulation, we use the topology shown in Fig. 2 with a single bottleneck link whose utilization was kept at 85% using five TCP cross-traffic sources. Fig. 9(a) plots the evolution of E_C produced by CapProbe in this experiment. As the figure shows, CapProbe’s minimum filtering is sensitive to random queuing delays in front of the first packet of the pair and can eventually converge to a completely wrong value (60% error).

We next examine how the value of $\tilde{C} \neq C$ supplied to IGI and *spruce* affects their accuracy. We plot the relative estimation errors in Fig. 9(b), in which the accuracy of both methods deteriorates when \tilde{C}/C becomes large. To understand the exact effect of \tilde{C} on these methods, we have the following simple analysis.

According to [8], IGI first sends packet trains Z_i with inter-packet spacing x_i ($x_i < x_j$ for $i < j$) to determine the *turning point* \hat{x} . Each packet-train consists of k back-to-back packets and the turning point is the n -th inter-packet spacing x_n at which the receiving rate of probe traffic starts to match the sending rate:

$$\hat{x} = x_n = \frac{1}{k-1} \sum_{i=2}^k y_i. \quad (30)$$

Subsequently, IGI computes the average cross-traffic rate \bar{r} as following [8]:

$$\bar{r} = \frac{\sum_{y_i > \max(\Delta, \hat{x})} \tilde{C} (y_i - \Delta)}{(k-1)\hat{x}} \quad (31)$$

and estimates available bandwidth by subtracting (31) from its a-priori known value of \tilde{C} : $\hat{A} = \tilde{C} - \bar{r}$. Notice from (31) that the average cross-traffic \bar{r} depends on the capacity value provided to the estimation algorithm. This dependency explains the increased estimation inaccuracy when $\tilde{C}/C \neq 1$ as illustrated in Fig. 9(b).

Similarly, estimation accuracy of *spruce* also changes as a function of \tilde{C} . For example, even though *spruce* provides

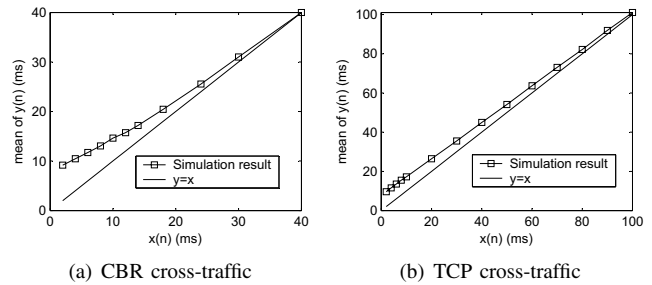


Fig. 10. Convergence of ω_n to zero-mean additive noise for large x_n under CBR and TCP cross-traffic.

better estimates than IGI with a correct capacity value $\tilde{C} = C = 1.5$ mb/s, it exhibits large estimation errors (over 200%) when \tilde{C} exceeds C by 33%. This means that the accuracy of *spruce* is also heavily dependent on the performance of the underlying bottleneck bandwidth estimation method.

To better understand this observation, we next analyze *spruce*’s estimation process. *Spruce* collects individual samples A_i [26]:

$$A_i = C \left(1 - \frac{y_i - \Delta}{\Delta} \right), \quad (32)$$

where y_i is the i -th measured packet spacing at the receiver. The algorithm averages samples A_i to obtain a running estimate of the available bandwidth:

$$A_n = \frac{1}{n} \sum_{i=1}^n A_i = 2C - \frac{C^2}{qn} \sum_{i=1}^n y_i. \quad (33)$$

Taking the limits of (33) and substituting W_n from (21) into (33), we get

$$\hat{A} = \lim_{n \rightarrow \infty} A_n = C - \frac{x}{\Delta} \bar{r}. \quad (34)$$

This brief analysis shows that the bandwidth estimation mechanism in *spruce* requires that the sender set its inter-packet spacing x to be $\Delta = q/C$. This is possible when C is known exactly; however, in cases when C is not correctly estimated, the initial spacing x cannot be set to Δ and *spruce* cannot converge to A . Also notice that if $\bar{r} \ll C$, estimation errors are generally small; however, as link utilization increases, any deviation of x/Δ from 1 will have a significant impact on \hat{A} .

V. MULTIPLE LINKS

In this section, we extend our single-node model to the case of multiple congested routers and conjecture that it is impossible to derive a closed-form solution that filters out the noise introduced by cross-traffic at several links of an end-to-end path.

A. Large Inter-Probe Delays

Consider the original model of a router in (2). This time, assume that *none* of the probe packets queue behind each other at the bottleneck router. This means that packet $n-1$ leaves

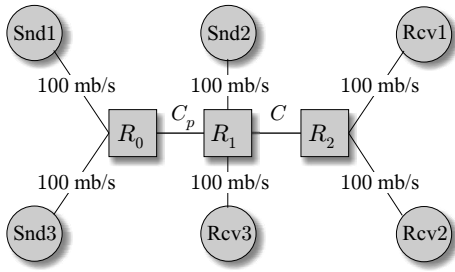


Fig. 11. Multi-link simulation topology.

the router before packet n arrives, which is expected if inter-packet spacing x_n at the source is very large compared to the transmission delay Δ . Under these assumptions, $d_{n-1} < a_n$ and (2) becomes:

$$d_n = a_n + \omega_n + \Delta, \quad n \geq 1. \quad (35)$$

Hence, inter-departure delays y_n are:

$$y_n = a_n - a_{n-1} + \omega_n - \omega_{n-1}, \quad n \geq 2. \quad (36)$$

Notice that the first term $a_n - a_{n-1}$ in (36) is the inter-arrival delay x_n of the probe traffic and the second term $\omega_n - \omega_{n-1}$ can be modeled as some zero-mean random noise. This can be explained intuitively by noticing that under the assumption of large x_n , each router delays probe packets (on average) by the same amount. Then the distance between each pair of subsequent packets fluctuates around the mean of x_n . Using an inductive argument, it is also easy to show the following.

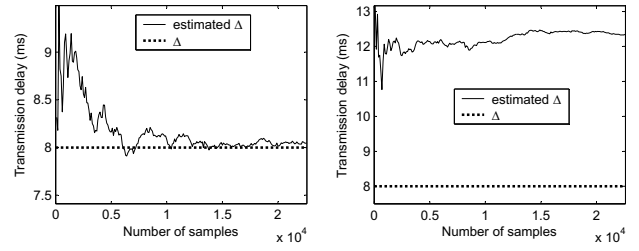
Claim 5: If the initial spacing x_n is larger (in a statistical sense) than queuing delays experienced by packets at each router of an N -hop end-to-end path, the mean of the sampled signal y_n is equal to $E[x_n]$ and the following holds for each router j :

$$E[\omega_n^{(j)} - \omega_{n-1}^{(j)}] = 0. \quad (37)$$

To confirm that the zero-mean model in (37) holds in practice, we run ns2 simulations with 85% utilization at the bottleneck link and varying packet sizes of CBR and TCP cross-traffic. The plots of $E[y_n]$ as a function of x_n for different values of $x_n = x$ are shown in Fig. 10. As the two figures show, $E[y_n]$ converges to x_n at 40 and 100 ms, respectively, at which time the noise at the bottleneck router becomes zero-mean-additive. Note that similar results hold for multiple congested routers, different traffic patterns, and different packet sizes. Also note that the point at which $E[y_n]$ converges to x is *not* necessarily the value of the available bandwidth as was suggested in prior work [8].

B. Recursive Model for Multi-Node Paths

Assuming multiple congested routers along a path, the result in (25) no longer holds. To better understand multi-link effects, we use the topology in Fig. 11, where the bottleneck link C has capacity 1.5 mb/s and pre-bottleneck link $C_p = 1.8$ mb/s. Five FTP sources are attached to each of snd2 and snd3



(a) Single congested link

(b) Two congested links

Fig. 12. The evolution of estimated transmission delay $\tilde{\Delta}_n$ for a single and two congested links.

and both links are utilized by cross-traffic at 85%. Fig. 12 illustrates the convergence process of $\tilde{\Delta}_n$ for the single-node and multi-node cases. As Fig. 12(b) shows, estimates $\tilde{\Delta}_n$ do not converge to the true value of $\Delta = 8$ ms, regardless of the number of samples y_i .

Notice that it is possible to recursively extend the original model in (26) to multiple congested links where the input x_n of each link is the output y_n of the previous link. However, this model becomes intractable as we show next. Add index j to each process and each random variable to indicate that it is local to router j along the path from the sender to the receiver. Further assume that $Q_i^{(j)}$ is the queuing delay (due to cross-traffic and other probe packets) in front of packet i inside router j and $\phi_i^{(j)}$ is some zero-mean noise process at router j at time i . Then, we define the following recursive model:

$$y_i^{(j)} = \begin{cases} \Delta_j + \omega_i^{(j)}, & x_i^{(j)} < Q_{i-1}^{(j)} \\ \Delta_j + \phi_i^{(j)}, & x_i^{(j)} \geq Q_{i-1}^{(j)} \end{cases}, \quad (38)$$

where $y_i^{(j)}$ is the departure spacing of the i -th packet-pair from router j , $\Delta_j = q/C_j$ is the transmission delay of the probe packets over link j , and $x_i^{(j)}$ is the arrival spacing between the probes at router j . Notice that if the arrival spacing between packets $i-1$ and i (which is simply $x_i^{(j)} = y_{i-1}^{(j)}$) is less than the time packet $i-1$ spends in the buffer (i.e., $Q_{i-1}^{(j)}$), the two packets queue behind each other and follow the model developed earlier in this paper. When the opposite holds, the packets do not queue behind each other and the router adds *i.i.d.* zero-mean noise $\phi_i^{(j)}$ to Δ_j .

C. Conjecture of Impossibility

One of the main difficulties of this situation is the stochastic mixture of the two types of noise in (38). While at this time we do not offer a complete treatment of this problem, we show that *even* when all links follow the first model (i.e., all packets queue behind each other), the problem appears intractable. Under these assumptions, (38) leads to:

$$W_n^{(j)} = \Delta_j + \frac{1}{nC_j} \sum_{i=1}^n r_j(t_i) x_i^{(j)}, \quad (39)$$

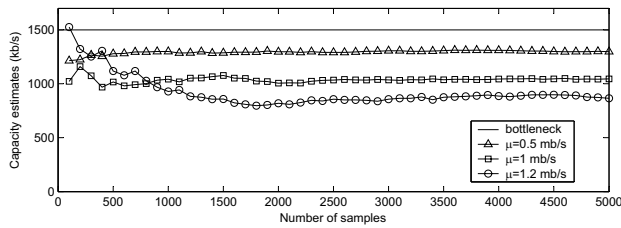


Fig. 13. Evolution of estimated bandwidth for two congested links.

where $r_j(t)$ is the cross-traffic process at router j . For a two congested-router case, (39) becomes:

$$\begin{aligned}
 W_n^{(2)} &= \Delta_2 + \frac{1}{nC_2} \sum_{i=1}^n r_2(t_i) y_i^{(1)} \\
 &= \Delta_2 + \frac{1}{nC_2} \sum_{i=1}^n r_2(t_i) \left[\Delta_1 + \frac{r_1(t_i)x}{C_1} \right] \\
 &\rightarrow \Delta_2 + \frac{\Delta_1 \bar{r}_2}{C_2} + \frac{x}{C_1 C_2} \int_0^\infty r_1(u) r_2(u) du. \quad (40)
 \end{aligned}$$

Since Poisson samples taken at the receiver lead to obtaining a time average of a mixed signal introduced by the two routers, it appears impossible to filter out the unknown cross-traffic statistics or the main integral in (40).

We next examine how the amount of cross-traffic in two congested links affects the estimation accuracy of the earlier models developed in the paper. We set the average cross-traffic rates $r_1(t)$ and $r_2(t)$ to be the same in both routers (i.e., $\bar{r}_1 = \bar{r}_2 = \mu$) and vary this value between 500 kb/s and 1.2 mb/s. We again use the topology in Fig. 11 and plot in Fig. 13 the value of \hat{C}_n computed using (27) when cross-traffic is simultaneously injected into both links ($C_p = 2$ mb/s, $C = 1.5$ mb/s). As the figure illustrates, the estimates of the bottleneck bandwidth converge to a value that is significantly less than the true capacity C and become progressively worse as μ increases.

VI. CONCLUSION

This paper examined the problem of estimating the capacity/available bandwidth of a single congested link and showed a simple stochastic analysis of the problem. Unlike previous approaches, our estimation did *not* rely on empirically-driven methods, but instead used a queuing model of the bottleneck router that specifically assumed non-negligible, non-fluid cross-traffic. It is also the first model to provide simultaneous asymptotically-accurate estimation of both C and A in the presence of arbitrary cross-traffic.

Our analysis in the multi-node cases suggests that the problem of obtaining provably-convergent estimates of C and A in such environments does not have a solution. This insight possibly explains the lack of analytical/stochastic justification for many of the proposed methods. Knowing that accurate estimation of C for the multi-node case is impossible will potentially provide an important foundation for the methods that rely on empirically-optimized heuristics.

REFERENCES

- [1] B. Ahlgren, M. Björkman, and B. Melander, "Network Probing Using Packet Trains," *Swedish Institute Technical Report*, March 1999.
- [2] M. Allman and V. Paxson, "On Estimating End-to-End Network Parameters," *ACM SIGCOMM*, August 1999.
- [3] J. Bolot, "End-to-End Packet Delay and Loss Behavior in the Internet," *ACM SIGCOMM*, September 1993.
- [4] R.L. Carter and M.E. Crovella, "Measuring Bottleneck Link Speed in Packet Switched Networks," *International Journal on Performance Evaluation* 27 & 28, 1996.
- [5] C. Dovrolis, P. Ramanathan, D. Moore, "What Do Packet Dispersion Techniques Measure?" *IEEE INFOCOM*, April 2001.
- [6] A.B. Downey, "Using PATHCHAR to Estimate Internet Link Characteristics," *ACM SIGCOMM*, August 1999.
- [7] K. Harfoush, A. Bestavros, and J. Byers, "Measuring Bottleneck Bandwidth of Targeted Path Segments," *IEEE INFOCOM*, March 2003.
- [8] N. Hu and P. Steenkiste, "Evaluation and Characterization of Available Bandwidth Probing Techniques," *IEEE Journal on Selected Areas in Communications*, vol. 21, No. 6, August 2003.
- [9] V. Jacobson, "Congestion Avoidance and Control," *ACM SIGCOMM*, 1988.
- [10] V. Jacobson, "pathchar - A Tool to Infer Characteristics of Internet Paths," <ftp://ftp.ee.lbl.gov/pathchar/>.
- [11] M. Jain and C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput," *ACM SIGCOMM*, 2002.
- [12] M. Jain and C. Dovrolis, "Pathload: A Measurement Tool for End-to-End Available Bandwidth," *Passive and Active Measurements (PAM) Workshop*, 2002.
- [13] R. Kapoor, L. Chen, L. Lao, M. Sanadidi, and M. Gerla, "CapProbe: a Simple and Accurate Capacity Estimation Technique," *ACM SIGCOMM*, August 2004.
- [14] S. Keshav, "A Control-Theoretic Approach to Flow Control," *ACM SIGCOMM*, 1991.
- [15] K. Lai and M. Baker, "Measuring Bandwidth," *IEEE INFOCOM*, 1999.
- [16] K. Lai and M. Baker, "Measuring Link Bandwidths Using a Deterministic Model of Packet Delay," *ACM SIGCOMM*, August 2000.
- [17] X. Liu, K. Ravindran, B. Liu, and D. Loguinov, "Single-Hop Probing Asymptotics in Available Bandwidth Estimation: A Sample-Path Analysis," *ACM IMC*, October 2004.
- [18] B.A. Mah, "pchar: A Tool for Measuring Internet Path Characteristics," <http://www.employees.org/~bmah/Software/pchar/>, June 2001.
- [19] B. Melander, M. Björkman, P. Gunningberg, "A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks," *IEEE GLOBECOM*, November 2000.
- [20] A. Pasztor and D. Veitch, "Active Probing using Packet Quartets," *ACM IMW*, 2002.
- [21] A. Pasztor and D. Veitch, "The Packet Size Dependence of Packet Pair Like Methods," *IEEE/IFIP IWQoS*, 2002.
- [22] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics," *Ph.D. dissertation*, Computer Science Department, University of California at Berkeley, 1997.
- [23] S. Ratnasamy and S. McCanne, "Inference of Multicast Routing Trees and Bottleneck Bandwidths using End-to-End Measurements," *IEEE INFOCOM*, March 1999.
- [24] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient Available Bandwidth Estimation for Network Paths," *Passive and Active Measurements (PAM) Workshop*, 2003.
- [25] D. Sisalem and H. Schulzrinne, "The Loss-delay Based Adjustment Algorithm: A TCP-friendly adaptation scheme," *IEEE Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 1998.
- [26] J. Strauss, D. Katabi, and F. Kaashoek, "A Measurement Study of Available Bandwidth Estimation Tools," *ACM IMC*, 2003.
- [27] R. Wade, M. Kara, and P.M. Dew, "Study of a Transport Protocol Employing Bottleneck Probing and Token Bucket Flow Control," *IEEE ISCC*, July 2000.
- [28] R.W. Wolff, "Stochastic Modeling and the Theory of Queues," *Prentice-Hall*, 1989.