PBProbe: A Capacity Estimation Tool for High Speed Networks[*]

Ling-Jyh Chen[1], Tony Sun[2], Bo-Chun Wang[1], M. Y. Sanadidi[3], and Mario Gerla[3]

[1]Institute of Information Science, Academia Sinica

[2]PacketMotion Inc.

[3]Department of Computer Science, University of California at Los Angeles

Abstract

Knowledge about the bottleneck capacity of an Internet path is critical for efficient network design, management, and usage. In this paper, we propose a new technique, called PBProbe, for estimating high speed links rapidly and accurately. Although it is based on CapProbe, instead of relying solely on packet pairs, PBProbe employs the concept of "Packet Bulk" and adapts the bulk length to compensate for the well known problem with packet pair-based approaches, namely the lack of accurate timer granularity. As a result, PBProbe not only preserves the simplicity and speed of CapProbe, but also correctly estimates link capacities over a much larger range. Using analysis, we evaluate PBProbe with various

bulk lengths, network configurations, and traffic models. We then perform a set of experiments to evaluate the accuracy of PBProbe on the Internet over wired and wireless links. Finally, we perform emulation and Internet experiments to verify the accuracy and speed of PBProbe on high speed links (e.g., the Gigabit Ethernet connection). The results show that PBProbe is consistently fast and accurate in the majority of test cases.

Keyword: Analysis, High Speed Networks, Link Capacity Estimation, Simulation, Experiments.

1 Introduction

Estimating the bottleneck capacity of an Internet path is a fundamental research problem in computer networking, since knowledge of the capacity is critical for efficient network design, management and usage. In recent years, with the growing popularity of emerging technologies, such as overlay, peer-to-peer (P2P), sensor, grid, and mobile networks, it has become increasingly important to have a simple, fast and accurate tool for capacity estimation and monitoring. To accommodate the diversity of network arrangements, a capacity estimation tool should also be scalable and applicable to a variety of network configurations.

A number of techniques have been proposed for capacity estimation of generic Internet paths [3, 7, 14, 16, 18, 20, 29]. Among them, Pathrate [14] and CapProbe [18] are widely recognized as fast and accurate tools for generic network scenarios. However, CapProbe is a round-trip estimation scheme that only works well on paths with a symmetric bottleneck link, while Pathrate is based on histograms and may converge slowly if the initial dispersion measurements are not unimodal. As a result, CapProbe has difficulty estimating the capacity of asymmetric links [12], and Pathrate performs poorly on wireless links [18].

To address the above problems, specialized capacity estimation tools have been proposed for specific and emerging network scenarios. For instance, ALBP [24] and AsymProbe [12] are intended for capacity estimation on asymmetric links, and Ad-Hoc Probe [11] tries to estimate the end-to-end path capacity of wireless networks. However, recent studies have shown that, for emerging high speed network links (i.e., gigabit links), estimating high speed link capacity remains a major challenge [17, 19]; hence, a simple, fast, and accurate technique is desirable.

In this paper, we propose a capacity estimation tool, called PBProbe, for high speed network links. Although PBProbe is inspired by CapProbe, instead of relying solely on one pair of packets, it employs the concept of "Packet Bulk" to adapt the number of probe packets in each sample in accordance with the dispersion measurement. More specifically, when the bottleneck link capacity is expected to be low, PBProbe uses one pair of packets as usual (i.e., the bulk length is 1). For paths with high bottleneck capacities, PBProbe increases the bulk length and sends several packets together. This enlarges the dispersion between the first and last packets, and resolves the well known timer granularity problem. Timer granularity is the main challenge in estimating the capacity of high capacity links [17] [19]. This issue is discussed in detail later in the paper.

We study the performance of PBProbe under different Poisson cross traffic loads via analytical models; and evaluate the technique's accuracy and speed through testbed experiments, a laboratory testbed, and on high speed Internet paths. We also compare it with Pathrate. The results show that PBProbe can accurately and rapidly estimate the link capacities in all tested scenarios, thereby outperforming Pathrate in most experiments.

The remainder of this paper is organized as follows. In Section 2, we summarize related work on capacity estimation. In Section 3, we describe PBProbe. Section 4 contains an analysis of PBProbe and an evaluation of its speed and accuracy with Poisson

cross traffic, Deterministic cross traffic, and Pareto ON/OFF cross Traffic. In Section 5, we describe PBProbe experiments for general network scenarios, and demonstrate the approach's compatibility with general Internet links. In Section 6, we evaluate PBProbe on high speed links (e.g., Gigabit Ethernet) in our emulator testbed as well as on the Internet. Then, in Section 7, we summarize our conclusions.

2 Background and Overview

2.1 Related Work

Previous research on capacity estimation relied on delay variations among probe packets, as illustrated by Pathchar [16], or on dispersion among probe packets, as in Nettimer [20] and Pathrate [14]. Pathchar-like tools (such as pchar [7] and clink [3]) are limited in terms of accuracy and speed, as reported in [18, 21]. Moreover, they evaluate the capacity of a link based on the estimates of previous links along the path; thus, estimation errors accumulate and amplify with each link measured [29].

Dispersion-based techniques suffer from other problems. In particular, Dovrolis' analysis in [14] shows that the dispersion distribution can be multi-modal due to cross traffic, and that the strongest mode of such a distribution may correspond to (1) the capacity of the path; (2) a "compressed" dispersion, resulting in capacity over-estimation; or (3) the Average Dispersion Rate (ADR), which is always lower than the capacity. Another dispersion-based tool, SProbe [29], exploits the SYN and RST packets of the TCP protocol to estimate the downstream link capacity, and employs two heuristics to filter out samples affected by cross traffic. However, SProbe does not work efficiently when the network is heavily utilized [23].

Unlike the above approaches, CapProbe [18] uses both dispersion measurements and end-to-end delay measurements to filter out packet pair samples distorted by cross traffic. The method has been shown to be both fast and accurate in a variety of sce-
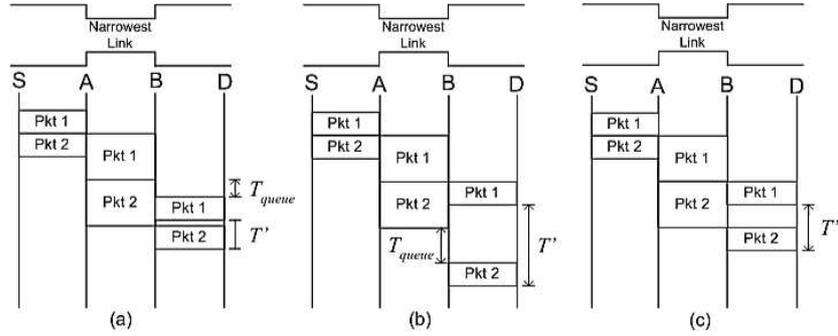
Figure 1: (a) Over-estimation caused by "compression", (b) Under-estimation caused by "expansion", (c) The ideal case.

narios. The original implementation of CapProbe uses ICMP packets as probe packets, and measures the bottleneck capacity on a round-trip basis. As a result, the capacity estimate does not reflect the higher capacity link when the path is asymmetric. Other difficulties are encountered when intermediate nodes block ICMP packets [25], or employ priority schemes to delay ICMP packet forwarding (e.g., the Solaris operating system limits the rate of ICMP responses, and is thus likely to perturb CapProbe measurements) [30].

Recent capacity estimation studies have extended the target network scenarios to more diverse environments. For instance, Lakshminarayanan et al. evaluated tools for estimating the capacity and available bandwidth of emerging broadband access networks [22], while Chen et al. extended CapProbe to estimate the *effective path capacity* in ad hoc wireless networks [11]. In addition, ABLP [24] and AsymProbe [12] have been developed for capacity estimation of the increasingly popular asymmetric links, such as DSL and satellite links.

Nonetheless, capacity estimation on high speed links remains a challenge. Though recent studies verified the accuracy of Pathrate in estimating gigabit links [28], the evaluations were conducted on an emulator-based testbed, which cannot represent realistic Internet dynamics [15]. Thus, an experimental evaluation of capacity estimation on high speed links is still required.

5

In this paper, we propose a novel packet bulk technique called PBProbe for estimating the capacity of high speed links. PBProbe is based on CapProbe, but it probes the bottleneck link capacity using UDP packets (instead of the ICMP packets used by CapProbe). We summarize the CapProbe algorithm in the next subsection.

2.2 CapProbe: An Overview

CapProbe is a packet pair-based capacity estimation technique that has proven both fast and accurate in several scenarios [18]. Conceptually, when a back-to-back packet pair is transmitted over a network, and assuming both packets arrive at the bottleneck link unperturbed (i.e., back-to-back) by cross traffic on previous links, they are always dispersed at that link according to the bottleneck capacity. Although the dispersion will accurately reflect the bottleneck capacity (as shown in Fig. 1-c), if either packet in a pair has been queued due to cross traffic, the dispersion of the sample might be expanded or compressed; "expansion" leads to under-estimation and "compression" leads to over-estimation of the capacity (as shown in Fig. 1-a,b).

CapProbe combines dispersion measurements and end-to-end delay measurements to filter out packet pair samples distorted by cross traffic. The basic idea is that if a sample has not been queued by cross traffic, it can be used to estimate the bottleneck capacity correctly. For such "good" samples, the sum of the delays of the packet pairs, called the *delay sum*, does not include cross-traffic induced queuing delay, and is indeed smaller than the delay sum of the samples distorted by cross traffic. Thus, these "good" samples can be identified easily because the delay sum will be the minimum of the delay sums of all packet pair samples. We call this delay sum the *minimum delay sum*. In this way, the capacity can be estimated by the equation:

$$C = \frac{P}{T},$$ (1)

where P is the sample packet size, and T is the dispersion of the sample packet

6

pair with the minimum delay sum.

However, since CapProbe relies on dispersion measurements for accurate capacity estimation, it is inaccurate when estimating high speed links or when operating on slow machines [19]. More specifically, according to Eq. 1, when C is extremely large, T must become small or P must become very large. Since the accuracy of the T measurement is limited by the system timer's granularity, the only feasible solution for CapProbe estimation on high speed links is to enlarge the packet size. However, if the packet size becomes larger than the Maximum Transmission Unit (MTU), the "big" packet will be segmented into several fragments before entering the network. The fragments will then be reassembled as the original packet size at the receiver [27]. The latency caused by segmentation (at the sender) and reassembly (at the receiver) expands the dispersion measurement and therefore results in under-estimation. To prevent such additional latency, we propose sending multiple MTU packets back-to-back (i.e., virtually, the probe packets are input to the network as one "big" packet). As a result, the dispersion measurement can be used to accurately estimate the capacity using the CapProbe algorithm because the segmentation and reassembly latency can be avoided. We discuss the design, analysis, and evaluation of PBProbe in the following sections.

3   The Proposed Approach: PBProbe

Like CapProbe, PBProbe estimates the link capacity by actively sending a number of probes over the network and using the *minimum delay sum* filter to identify a "good" sample[1]. However, instead of employing a packet pair, PBProbe uses a *packet bulk* of length k in each probing and measures the capacity of each direction separately. Specifically, PBProbe is comprised of two phases. In the first phase, it estimates the capacity

---

[1]Note that, like previous studies [14, 16, 18], for simplicity, we do not consider the issues of traffic shaping and multi-path diversity in this study.
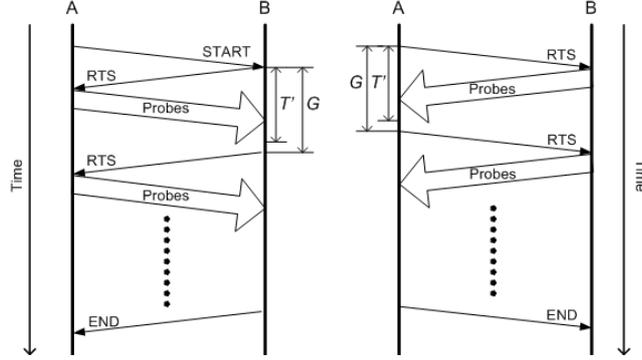
Figure 2: Illustration of PBProbe (a) Phase I: measuring forward direction link capacity; (b) Phase II: measuring backward direction link capacity.

of the *forward* link; and in the second, it estimates the capacity of the *backward* link. Fig. 2 details the PBProbe algorithm.

In the first phase (shown in Fig. 2-a), host A sends a *START* packet to host B to initiate the estimation process, after which B sends a *Request To Send (RTS)* packet to A every G time units. On receipt of the *RTS* packet, A immediately sends B a packet bulk of length k (note: bulk length = k means that $k + 1$ packets are sent back-to-back). For the i-th probing sample, suppose B sends the *RTS* packet at time $t_{send}(i)$ and receives the j-th packet (in the i-th sample) at time $t_{rcv}(i, j)$. The delay sum, S_i, and the dispersion, D_i, of the i-th packet bulk sample are given by

$$S_i = (t_{rcv}(i, 1) - t_{send}(i)) + (t_{rcv}(i, k + 1) - t_{send}(i)),\qquad(2)$$

$$D_i = t_{rcv}(i, k + 1) - t_{rcv}(i, 1).\qquad(3)$$

If none of the $k + 1$ probe packets experience cross-traffic induced queueing, the sample will reflect the correct capacity. Thus, the "good" sample (say, the m-th sample) is identified by applying the minimum delay sum filter to all probing samples (say, n samples) as follows:

$$m = arg \min_{i=1...n} S_i. \qquad (4)$$

Therefore, the capacity is estimated by using the dispersion of the m-th sample with the minimum delay sum:

$$C = \frac{kP}{D_m}, \qquad (5)$$

where P denotes the size of each probe packet. Since the packet bulk samples are only delivered in the forward direction, the estimated capacity corresponds to the bottleneck in that direction.

At the end of the first phase, B sends an *END* packet to A, and PBProbe enters the second phase, as shown in Fig. 2-b. In this phase, A first sends an *RTS* packet (every G time units) to B, which replies with a packet bulk of length k immediately on receipt of each *RTS* packet. Similar to the first phase, PBProbe measures the delay sum and dispersion for each sample, and estimates the capacity in the backward direction by using the minimum delay sum filter.

3.1   The Inter-sample Period: *G*

PBProbe assesses the link capacity by sending a packet bulk every G time units. The value of G is critical to the convergence time of PBProbe because the larger the value of G, the slower PBProbe's estimation becomes. However, G should not be too small; otherwise, PBProbe may create congestion in the network and require longer to converge. Therefore, for PBProbe, we set G as:

$$G = \frac{2D_{m'}}{U}, \qquad (6)$$

where $D_{m'}$ is the dispersion of the good sample (i.e., $D_{m'} = \frac{kP}{C}$) that has the minimum delay sum among all probing samples seen so far; and U is the maximum

network utilization allowed for PBProbe estimation. We provide a short proof below to show that, *if $G = \frac{2D_{m'}}{U}$, the network utilization constraint U can be guaranteed.*

Proof. Since $k \geq 1$, we know that

$$
\begin{aligned}
G &= \frac{2D_{m'}}{U} \\
&\geq \frac{D_{m'}}{U} + \frac{D_{m'}}{kU} \\
&= \frac{kP}{CU} + \frac{P}{CU} = \frac{(k+1)P}{CU}
\end{aligned}
\tag{7}
$$

$$
\Rightarrow \frac{(k+1)P}{G} \leq CU. \tag{8}
$$

Let R denote the data rate of the introduced packet bulk probes , i.e., $R = \frac{(k+1)P}{G}$; then, we can conclude that $R \leq CU$. In other words, the probing data rate is never larger than the load constraint U. $\qquad\square$

3.2 The Packet Bulk Length: *k*

The major difference between CapProbe and PBProbe is that the latter sends packet bulks. The purpose of using packet bulks is to overcome the system timer's granularity, and avoid the additional latency caused by segmentation and reassembly when the packet size is larger than the MTU[2]. PBProbe applies Algorithm 1 to automatically determine the proper bulk length, k, for capacity estimation.

In the beginning, k is initialized to 1, i.e., PBProbe behaves like CapProbe by using a packet-pair to probe the link capacity. However, whenever the measured dispersion is smaller than a certain threshold, say D_{thresh}, the algorithm increases the bulk length (k) ten-fold and restarts the estimation process. Clearly, the decision about the D_{thresh}

---

[2]The "path MTU", i.e., the largest packet size that can traverse this path without being fragmented, can be found by the *Path MTU Discovery* technique [26].

Algorithm 1 The algorithm for determining the appropriate bulk length, k, in PBProbe.

$k \leftarrow 1$
$count \leftarrow 0$
$D \leftarrow \infty$
repeat
  $t_1 \leftarrow time()$
  Send START packet
  Receive a packet bulk (length k) and measure D'
  if $D' < D_{thresh}$ **then**
    $k \leftarrow k \times 10$
    $count \leftarrow 0$
  else
    $D \leftarrow \min(D', D)$
    $G \leftarrow 2D/U$
    $count \leftarrow count + 1$
    $t_2 \leftarrow time()$
    Sleep(G - ($t_2 - t_1$))
  end if
until $count == n$

value depends on the system timer's granularity. In this work, we set $D_{thresh} = 1ms$ for all experiments.

3.3 The Number of Samples: *n*

CapProbe applies a sophisticated convergence test to determine whether a good sample has been obtained. To simplify the implementation, PBProbe simply estimates the link capacity using a fixed number of samples, n. Obviously, the larger the value of n, the more accurate the PBProbe estimates will be. The time required for one PBProbe capacity estimate is linearly proportional to the value of n. More specifically, the larger the value of n, the longer PBProbe will require to estimate the capacity. Based on the experiment results reported in previous CapProbe studies [13, 18], we set $n = 200$ throughout this paper.

4 Analysis

We now present a queueing model that can predict the probability of a "good" sample for a single link with three types of cross traffic distribution: Poisson, Deterministic, and Pareto ON/OFF distribution. For simplicity, we assume the probing samples of PBProbe arrive according to a Poisson distribution so that they take what amounts to "a random look" at the link. We also assume that the probing samples do not constitute a significant load on the network because they are sent infrequently. Finally, we assume the buffers are large enough so that probe packets will not be dropped due to buffer overflow. The results of our analysis are presented in the following sub-sections.

4.1 Poisson Cross Traffic

First, we analyze PBProbe with Poisson cross traffic on the bottleneck link. Suppose the arrival rate and the service rate of the Poisson cross traffic are λ and μ respectively, the service time of one single probing packet is τ, and the bottleneck link utilization is ρ. The probability that the first probe packet will find the system empty, i.e., p, is given by

$$p = 1 - \frac{\lambda}{\mu} = 1 - \rho. \tag{9}$$

Since none of the probe packets of a "good" sample experience queueing delay, the probability of no queueing delay for the remaining k probe packets (i.e., no cross traffic packets arrive in the $k\tau$ period) is $e^{-\lambda(k\tau)}$. Therefore, the probability of obtaining a "good" sample, i.e., p_0, is given by

$$p_0 = pe^{\lambda(-k\tau)} = (1 - \rho)\, e^{-k\lambda\tau}. \tag{10}$$

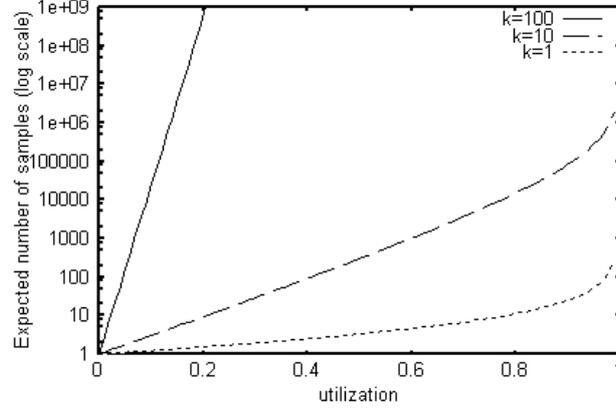The expected number of samples, $\overline{N}$, required to obtain a good sample is then derived by

Figure 3: The expected number of samples ($\overline{N}$) with different link utilization (ρ) and packet bulk length (k) under Poisson cross traffic.

$$\overline{N} = \sum_{n=1}^{\infty} np_0(1 - p_0)^{n-1} = \frac{1}{p_0} = \frac{e^{k\lambda\tau}}{1 - \rho}. \tag{11}$$

Suppose the probe packets and cross traffic packets are of equal size, then $\tau = \frac{1}{\mu} = \frac{\rho}{\lambda}$. Therefore, $\overline{N}$ can be rewritten as

$$\overline{N} = \frac{e^{k\rho}}{1 - \rho}. \tag{12}$$

The relationship between the expected number of required samples for one good sample ($\overline{N}$) and link utilization (ρ) for different packet bulk lengths (k) is shown in Fig. 3.

¿From Fig. 3, we observe that when $k = 1$ (i.e., packet-pair based CapProbe), $\overline{N}$ is around 25 when the utilization (ρ) is as high as 0.9. However, as k increases, $\overline{N}$ increases exponentially. For instance, when $\rho = 0.3^3$, $\overline{N}$ is around 30 when $k = 10$, but reaches approximately 1.5×10^{13} when $k = 100$. We find that the estimation speed of PBProbe (i.e., the number of samples required to obtain a good sample) is closely related to the packet bulk length employed. Though a large packet bulk can

---

[3]The utilization of the Abilene backbone network, which is the gigabit backbone of Internet2, is seldom higher than 30% [1].
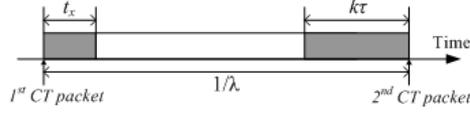
Figure 4: Illustration of the time interval between the arrival of two deterministic cross traffic packets.

improve the accuracy of dispersion measurements, a much larger number of attempts will be needed to derive a good sample, which will slow down the estimation process considerably.

4.2 Deterministic Cross Traffic

In this subsection, we analyze the probability of obtaining a good sample by using the packet bulk technique under deterministic cross traffic. Suppose the deterministic cross traffic deploys the same packet size with a rate equal to λ packets per time unit (i.e., the interval between any two contiguous cross traffic packets is $1/\lambda$ time units), and the transmission time of each cross traffic packet is t_x. To obtain a good sample of length k, the system must finish servicing the previous cross traffic and k probe packets within $1/\lambda$ time units, as shown in Fig. 4. Therefore, the probability of obtaining a good sample is given by

$$
\begin{aligned}
p_0 &= \max(0, 1 - \frac{t_x + k\tau}{\frac{1}{\lambda}}) \\
&= \max(0, 1 - k\rho - \lambda t_x).
\end{aligned}
\tag{13}
$$

When the cross traffic packets and the probe packets are the same size (i.e., $t_x = \tau$), p_0 can be rewritten as

14

$$p_0 = \max(0, 1 - k\rho - \lambda\tau)$$
$$= \max(0, 1 - k\rho - \rho). \tag{14}$$

To ensure that one can always obtain a good sample for accurate capacity estimation, p_0 must be larger than 0. Thus,

$$1 - k\rho - \rho > 0 \quad \Rightarrow \quad \rho < \frac{1}{k+1}. \tag{15}$$

Then, the expected number of samples required to obtain a good sample is given by

$$\overline{N} = \sum_{n=1}^{\infty} n p_0 (1 - p_0)^{n-1}$$
$$= \frac{1}{p_0} = \frac{1}{1 - (k+1)\rho}. \tag{16}$$

Fig. 5 depicts the relationship between $\overline{N}$ and ρ with packet bulks of different length, k. The results show that when the utilization (ρ) is smaller than $\frac{1}{k+1}$, a good sample can be obtained very rapidly (i.e., $\overline{N}$ is small); when ρ is very close to $\frac{1}{k+1}$, $\overline{N}$ suddenly increases to a very large number; and when ρ is larger than $\frac{1}{k+1}$, $\overline{N}$ is no longer very meaningful, since $p_0 = 0$ in this case and it is impossible to get any good samples. As a result, the larger k becomes, the smaller the effective range of ρ will be.

4.3 Pareto ON/OFF Cross Traffic

Pareto ON/OFF cross traffic has two states: ON and OFF. In the ON state, the cross traffic is actually deterministic; whereas in the OFF state, the cross traffic source does not transmit any packets. The distribution of ON and OFF periods follows the Pareto distribution. We assume that such periods are independently and identically distributed
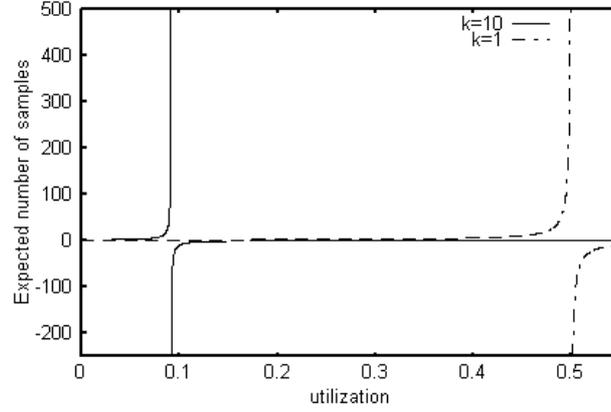
Figure 5: The expected number of samples $(\overline{N})$ with different link utilization (ρ) and packet bulk length (k) under deterministic cross traffic.

(*i.i.d.*). The probability density function (PDF), $f(t)$, of the ON/OFF periods and its mean, $\bar{t}$, are given by

$$f(t) = \frac{\alpha b^\alpha}{t^{\alpha+1}} \quad 0 < b \le t, 1 < \alpha \le 2. \tag{17}$$

and

$$\bar{t} = \frac{\alpha b}{\alpha - 1}, \tag{18}$$

where α and b are, respectively, the shape and scale parameters of the Pareto distribution. Since the mean length of the ON periods is equal to that of the OFF periods, we assume the mean arrival rate of the cross traffic is λ; hence, the mean arrival rate during ON periods will be 2λ.

Suppose t_x is the transmission time of a single cross traffic packet, and τ is the transmission time for a single probe packet. The analysis of the probability of obtaining a good sample can be divided into three cases as follows.

1. $t_x < \frac{1}{2\lambda} < t_x + k\tau$

   In this case, the inter-arrival time of the cross traffic (during an ON state) is

16

longer than the service time (t_x); thus, there will be *idle* time during an ON period. However, since the idle time is not long enough to serve k probe packets (i.e., $k\tau$ time units), the good sample can only arrive during an OFF period. According to the analysis presented in [18], the probability of obtaining a good sample can be derived from the probability that the residual life time Y will be larger than $k\tau$:

$$
\begin{aligned}
p_0 &= \frac{1}{2}P[Y \geq k\tau] \\
&= \begin{cases} \frac{1}{2\alpha}\left(\frac{b}{k\tau}\right)^{\alpha-1} & \text{if } k\tau \geq b \\ \frac{1}{2}[1 - \frac{(\alpha-1)k\tau}{\alpha b}] & \text{if } k\tau < b. \end{cases}
\end{aligned}
\tag{19}
$$

2. $\frac{1}{2\lambda} > t_x + k\tau$

   In this case, there is idle time during an ON period that is long enough to serve k probe packets. Therefore, the good sample can arrive during an ON or an OFF period. Since the cross traffic during an ON period is deterministic, the probability of obtaining a good sample during an ON period can be derived by the method presented in Section 4.2. Therefore, the probability of obtaining a good sample in this case will be the summation of the probabilities of obtaining a good sample during an ON state and an OFF state, given by

$$
p_0 = \begin{cases} P_{on} + \frac{1}{2\alpha}\left(\frac{b}{k\tau}\right)^{\alpha-1} & \text{if } k\tau \geq b \\ P_{on} + \frac{1}{2}[1 - \frac{(\alpha-1)k\tau}{\alpha b}] & \text{if } k\tau < b, \end{cases}
\tag{20}
$$

   where $P_{on} = \frac{1}{2}[1 - 2\lambda(t_x + k\tau)]$.

3. $t_x > \frac{1}{2\lambda}$

   In the third case, the inter-arrival time of the cross traffic packets (during an ON period) is shorter than the service time of a single cross traffic packet; thus, the
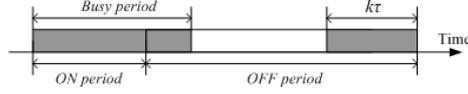
17

Figure 6: Illustration of the time interval between the arrival of two Pareto cross traffic sampls (Case 3).

system will be busy serving cross traffic packets during the entire ON periods, as well as in the first few time units of the OFF periods. Consequently, the good sample can only arrive during an OFF period when there is enough idle time to serve k probe packets. As shown in Fig. 6, the probability of obtaining a good sample will be 1 minus the proportion of the busy period (i.e., $\frac{\lambda}{\mu}$), minus the time required to service k probe packets (i.e. $\frac{k\tau}{2\bar{t}}$). Since the idle time may not be long enough to serve k probe packets either, we require that the probability can not be smaller than 0 by

$$p_0 = \max\left(0, 1 - \frac{\lambda}{\mu} - \frac{k\tau}{2\bar{t}}\right). \tag{21}$$

In summary, we have analyzed the packet bulk technique using Poisson, Deterministic, and Pareto ON/OFF cross traffic. The results show that the technique can rapidly obtain a good sample (a probing packet bulk with no queueing delay) when the utilization of the bottleneck link is low. For a heavily loaded link, the packet bulk needs a larger number of samples to accurately estimate the link capacity. In the following sections, we evaluate PBProbe further via a detailed set of experiments on an emulator-based testbed and on the Internet.

5 Evaluation on General Internet Links

In this section, we present the results of evaluating the accuracy of PBProbe on general Internet connections. We evaluate PBProbe on symmetric Internet links in subsection
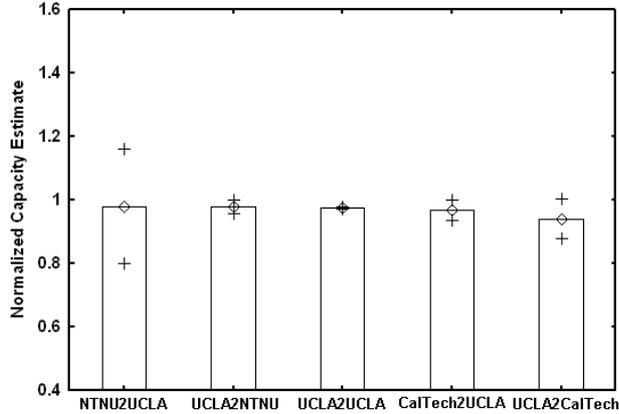
Figure 7: PBProbe capacity estimation (mean and 95% confidence intervals)on 100 Mbps Internet links.

5.1, and then present the evaluation results of PBProbe for emerging first/last-mile asymmetric access links and wireless links in subsections 5.2 and 5.3 respectively.

5.1 Symmetric Links

In the Internet experiments on the general symmetric fast Ethernet links, one of the selected paths is a local connection within the UCLA campus network (UCLA ↔ UCLA; Capacity: 100 Mbps; Round trip time: 0.2 ms). The other two paths are from UCLA to the California Institute of Technology (UCLA ↔ CalTech; Capacity: 100 Mbps; Round trip time: 8 ms) and from UCLA to National Taiwan Normal University (UCLA ↔ NTNU; Capacity: 100 Mbps; Round trip time: 180 ms).

For each network path, PBProbe was run 20 times to determine the average capacity estimate and standard deviation. The normalized capacity estimates (i.e., the ratio of estimated capacity to real capacity) and the 95% confidence intervals (i.e., ± 2 standard deviations) of the experiment results are illustrated in Fig. 7. The employed bulk length k and the average time required to make one estimate (in one direction) of each path are shown in Table 1.

19

Table 1: Path properties of the employed symmetric Internet connections.

Path	Bulk Length k	Time Spent
UCLA ↔ NTNU	10	10 sec
UCLA ↔ CalTech	10	10 sec
UCLA ↔ UCLA	10	10 sec

From the table, we observe that PBProbe adapted its bulk length, k, to 10 in all experiments. The reason is that when $k = 1$, PBProbe sends packet pairs as probing samples, which is the same as CapProbe. Hence, the dispersion measurements are smaller than the threshold value (i.e., $D_{thresh} = 1ms$). As a result, the bulk length adaptation algorithm increases k to ensure that the measured dispersion is larger than D_{thresh}, thereby overcoming the system's inadequate timer granularity.

Compared with the experiment results reported in [18], PBProbe achieves the same estimation accuracy as CapProbe and Pathrate on 100 Mbps links. As shown in Figure 7, PBProbe estimates the bottleneck capacity accurately in most cases. Note that the capacity estimate for the path from NTNU to UCLA fluctuates slightly more than the estimates for the other paths. This is because heavy cross traffic on the path reduces the probability of obtaining a good packet bulk sample.

In terms of estimation speed, PBProbe's performance is between that of CapProbe and Pathrate. It requires slightly more time than CapProbe because of packet bulk length adaptation and the fixed number of samples. Nevertheless, it is worth noting that, by employing packet bulks instead of packet pairs, PBProbe can accommodate systems with a lower timer granularity as well as higher capacity links; therefore, it is more flexible than CapProbe.

5.2 Asymmetric Links

The emerging first/last-mile Internet access links, such as DSL and Cable modem links, are usually asymmetric. Existing asymmetric estimation techniques are either very complicated (e.g. ALBP [24]) or they require fine-scale system timer granularity sup-
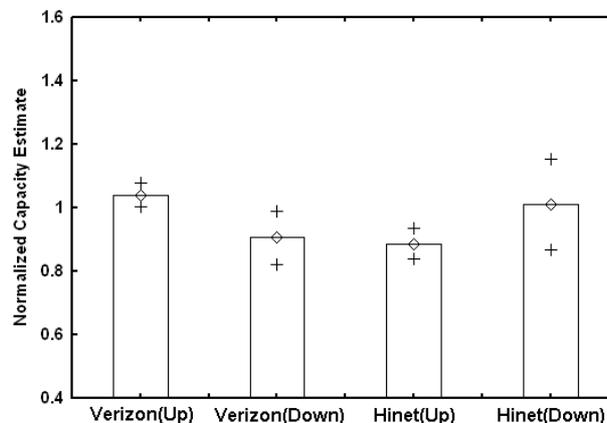
Figure 8: PBProbe experiment results (mean and 95% confidence intervals level, i.e., ± 2 standard deviations, of 20 runs) on Up and Down links of DSL connections.

port (e.g., AsymProbe [12]). In the following, we describe a set of experiments that evaluate the applicability of PBProbe to asymmetric access links.

We selected two asymmetric links from our local host (connected to the Internet via a 100 Mbps ethernet link) to two DSL hosts. One was a Verizon DSL link with 1.5M/384Kbps link capacity on the DOWN/UP links and approximately 30 ms round-trip delay time; the other was a HiNet DSL link with 3M/640Kbps link capacity on the DOWN/UP links and approximately 50 ms round-trip delay time[4]. The experiments were performed 20 times for each destination host, and k was observed to be 1 in all runs (i.e., when $k = 1$, the measured dispersion was already larger than the threshold value). In Fig. 8, we plot the normalized average capacity estimates with 95% confidence intervals for each link direction.

The results clearly demonstrate PBProbe's accuracy in estimating all asymmetric link capacities. Specifically, the average capacity estimates are all within 90% accuracy compared to the real physical link capacity. Given a 95% confidence level, all the capacity estimates remain within 80% accuracy.

---

[4]The DSL connections were provided by Verizon (http://www.verizon.net) and HiNet (http://www.hinet.net).
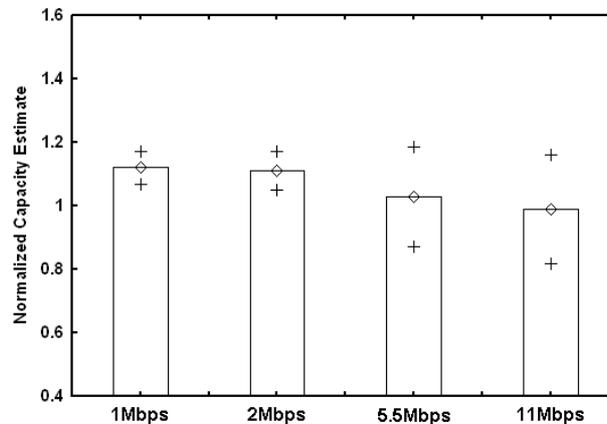
Figure 9: PBProbe experiment results (mean and 95% confidence intervals level, i.e., ± 2 standard deviation, of 20 runs) on last mile IEEE 802.11b links with different transmission modes.

5.3 Wireless Experiments

Next, we evaluate PBProbe on a one-hop wireless link because such links are becoming more prevalent in first/last mile scenarios. The wireless link used is based on IEEE 802.11b, and the transmission rate is configured as 1, 2, 5.5, and 11 Mbps respectively. For each transmission rate, we repeated the PBProbe experiment 20 times to collect the average and standard deviation results. Fig. 9 shows the experiment results[5]. From the figure, we observe that PBProbe also yields accurate capacity estimates for wireless links. The normalized capacity estimates are very close to 1 in all transmission modes, and the standard deviations are also small.

6 Evaluation on High Speed Links

We now evaluate the performance of PBProbe on high speed links (i.e., over 100 Mbps). First, we perform emulation experiments to validate the speed and accuracy of PBProbe in a controlled environment using different bulk lengths, and then conduct

---

[5]The capacity estimates were normalized to the effective capacity as reported at http://www.uninett.no/wlan/throughput.html
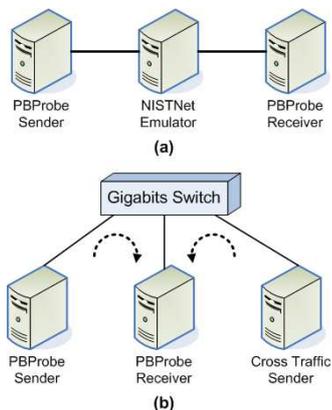
Figure 10: NIST Net emulation testbed

Internet experiments to study its performance in a more realistic environment.

6.1 Emulation Experiments

The emulator-based experiments were run on our laboratory testbed with the configurations illustrated in Fig. 10. The testbed comprised three Linux-based desktops (Fedora Core 3 with kernel version 2.6.9-1.667), each of which had an Intel Pentium 4 2GHz CPU, 2GB RAM, and two Intel PRO/1000 GT 32-bit PCI Gigabit Ethernet adapters. In the first scenario (Fig. 10-a), three test machines were connected in series with 1 Gbps links. The NISTNet emulator [6] was installed on the middle machine to create a bottleneck link on the gigabits path. No cross traffic was added to this path. In the second scenario (Fig. 10-b), three test machines were connected to a one gigabit switch with a 1 Gbps link. A Poisson traffic generator [8] was installed on one of the three machines, and PBProbe was installed on the other two machines. The Poisson traffic generator was configured to generate different levels of cross traffic in order to validate the accuracy and speed of PBProbe under different link utilization levels.

6.1.1 No Cross Traffic

First, we evaluated the accuracy of PBProbe with different k settings on high speed links by disabling the bulk length (k) adaptation algorithm. Fig. 10-a illustrates the testbed configuration in which no cross traffic was present during the experiments. We varied the bottleneck link capacity, and ran the experiment 20 times for each capacity setting. The results of $k = 10$ and 100 are shown in Fig. 11.

Since there was no cross traffic in these experiments, packets in a packet bulk were expected to traverse the path back-to-back without being disturbed. Therefore, ideally, the measured dispersion of each packet bulk should represent the undistorted dispersion corresponding to bottleneck link capacity. Moreover, based on these perfect dispersions, the capacity estimate should be accurate and consistent for every probing sample.

However, since PBProbe requires accurate dispersion measurements, the capacity estimates tend to become inaccurate when the timer granularity is inadequate. Table 2 shows the required timer granularity of PBProbe on links with different capacities and different bulk lengths. Obviously, when the bottleneck link capacity is high or the bulk length is small, a high timer granularity is required. For instance, with link capacity = 100 Mbps and bulk length = 1 (i.e., packet pairs), only a powerful processor can satisfy the required resolution of 0.12 ms. As the capacity increases to 1 Gbps, it becomes very difficult to achieve the required 0.012 ms resolution, and the measurement accuracy will be degraded as a consequence. Indeed, this trend can be observed in Figs. 11-a and 11-b, where, for small k, the capacity estimates become increasingly inaccurate and inconsistent when the required timer granularity increases (or equivalently, when the bottleneck capacity is enlarged).

However, the results also reveal that, when measuring high capacity links, a large k can successfully minimize the inaccuracy and inconsistency of capacity estimates caused by poor timer granularity. For instance, when the bottleneck link capacity is
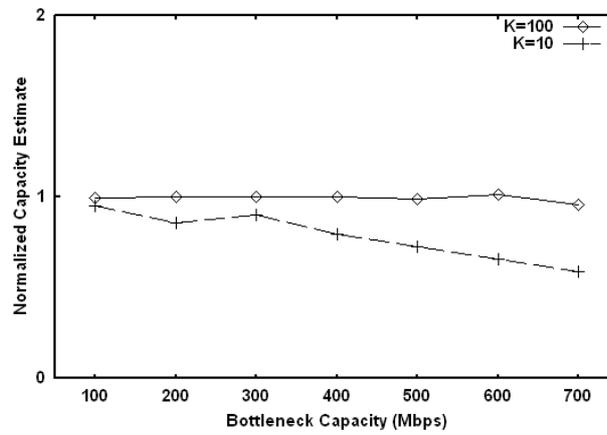
(a)
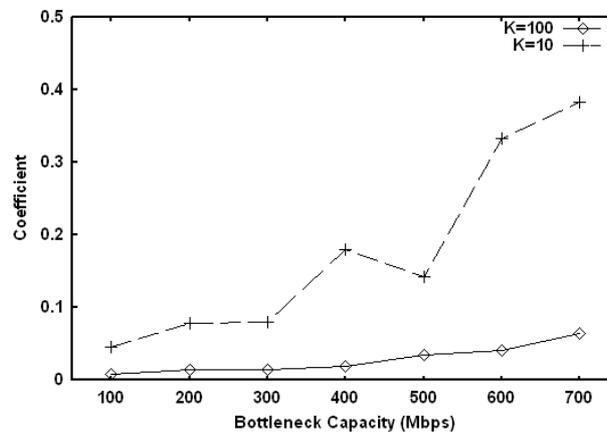
(b)

Figure 11: The results of PBProbe estimation using the NISTNet emulator: (a) normalized capacity estimates; and (b) coefficient of variation of capacity estimates with various bottleneck capacity settings.

Table 2: Required system timer granularity for accurate PBProbe estimation (assuming the probe packet size is 1500 bytes).

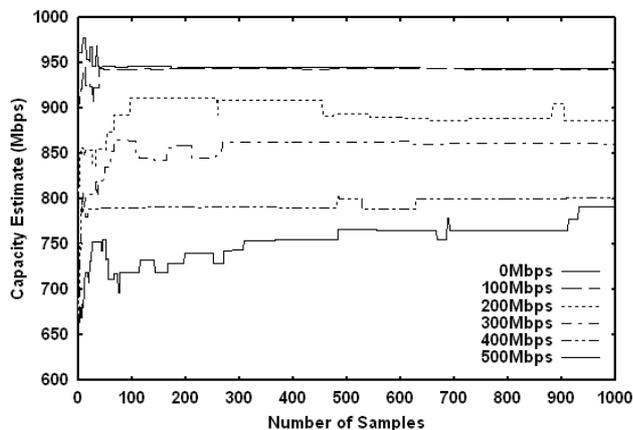|	Bottleneck Link Capacity		
k	1Gbps	100Mbps	10Mbps
1	0.012ms	0.12ms	1.2ms
10	0.12ms	1.2ms	12ms
100	1.2ms	12ms	120ms

Figure 12: Capacity estimates of PBProbe ($k = 100$) under different Poisson cross traffic.

600 Mbps, PBProbe with bulk length $k = 100$ can accurately estimate the capacity (as illustrated by the very small coefficient of variation in capacity estimates). In contrast, when $k = 10$, PBProbe can only measure around 60% of the link capacity, and the coefficient of variation is much larger. Based on the experiment results, and the analysis in Table 2, we conclude that our testbed hosts can only provide a timer granularity of approximately 1 ms. Therefore, in the following experiments, we fix $T_{thresh} = 1ms$ in PBProbe's k adaptation algorithm.

6.1.2 Poisson Cross Traffic

The objective of the second set of experiments was to investigate the impact of different levels of cross traffic on the accuracy and speed of PBProbe. Fig. 10-b illustrates a testbed topology in which a Poisson traffic generator inputs traffic to the probing path of PBProbe. We varied the generator's traffic rate from 0 to 500 Mbps, and performed 20 trials of PBProbe for each cross traffic rate with the bulk length k fixed at 100. The relationship between the number of employed samples and the average capacity estimates is plotted in Fig. 12.

The results show that PBProbe estimated 950 Mbps capacity very rapidly (in less

26

than 50 samples) when the Poisson cross traffic was very light (i.e., the Poisson cross traffic varied from 0 to 100 Mbps). As the cross traffic increased, the accuracy of PBProbe estimates decreased. Specifically, PBProbe achieved around 90% accuracy when the link utilization was 0.2, and 85% accuracy when it was 0.3. However, when the link utilization was greater than 40%, PBProbe only achieved around 80% accuracy after 1,000 samples. The results are consistent with the analytical results shown in Fig. 3, where the required number of samples increases exponentially with a link's utilization.

We find that, if the bulk length is large, PBProbe can estimate a link's capacity very rapidly when the link utilization is low, but it requires a large number of probing samples when the link is heavily utilized. Fortunately, most current Internet gigabit backbones are moderately loaded. As reported in [1, 2, 4, 5, 9][6], the utilization of Internet backbones is mostly below 15%. Therefore, in practice, PBProbe is still adequate for capacity estimation on high speed Internet links. To validate the feasibility and capability of PBProbe in more realistic scenarios, we describe experiments performed on high speed Internet paths in the next subsection.

6.2   Internet Experiments

We conducted Internet experiments to evaluate PBProbe in realistic environments. Five Internet hosts and five Internet gigabit paths (within California: UCLA and CalTech; cross country: UCLA - PSC, PSC - GaTech, GaTech - UCLA; and international: NTNU - UCLA) were selected for the experiments. The selected Internet hosts are listed in Table 3, and the topology and path properties of the selected paths are shown in Fig. 13.

Figs. 14 and 15 illustrate the experiment results (i.e., the normalized mean and the coefficient of variation of capacity estimates in 20 runs). From Fig. 14, it is clear that

---

[6]More information about Internet utilization can be found at http://netmon.grnet.gr/weathermap/ and http://people.ee.ethz.ch/ oetiker/webtools/mrtg/users.html

Table 3: Description of the participating hosts in the gigabit Internet experiments.

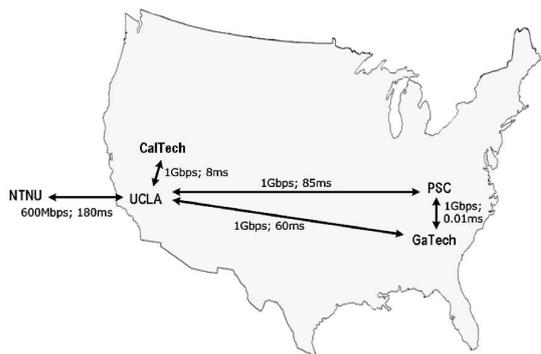| Abbrev.	Full Name	Location
CalTech	California Institute of Technology	California, USA
GaTech	Georgia Institute of Technology	Georgia, USA
NTNU	National Taiwan Normal University	Taipei, Taiwan
PSC	Pittsburgh Supercomputing Center	Pennsylvania, USA
UCLA	University of California at Los Angeles	California, USA

Figure 13: Topology and path properties (bottleneck capacity and round trip delay) of selected gigabit Internet paths.
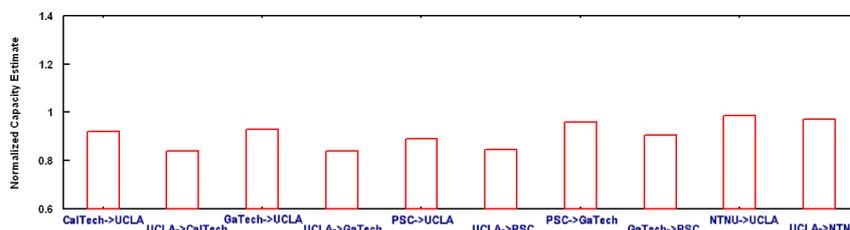

Figure 14: PBProbe experiment results (mean of 20 runs) on high speed links.

the normalized capacity estimates are mostly within 90% accuracy range, except for the three outgoing links from UCLA to CalTech, GaTech, and PSC. This was because the utilization of the outgoing links of the UCLA backbone was around 30%, which was much higher than the utilization of the incoming link (around 15%) [2]; therefore, in this case, PBProbe required a large number of samples to correctly estimate the capacity. Since we set $n = 200$ in all experiments, PBProbe could only estimate around 80% of the link capacity.
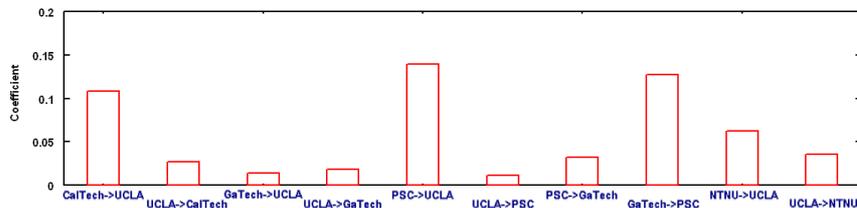
28

Figure 15: PBProbe experiment results (coefficient of variation of 20 runs) on high speed links.

Fig. 15 also shows that the coefficient of variation of PBProbe estimates is below 0.15 for all tested links, i.e., the capacity estimates (20 runs) of each high-speed link are very stable. Comparison with the results shown in Fig. 11 shows that PBProbe can adapt its bulk length to the most appropriate value (i.e., $k = 100$) so that it can consistently and accurately estimate the capacity of high speed links.

6.3 Comparison of PBProbe and Pathrate

We have evaluated PBProbe in a variety of network scenarios. Here, we compare the performance of PBProbe and Pathrate in terms of accuracy, speed, and bandwidth consumption (i.e., overhead). The experiments were performed on the set of high speed Internet links illustrated in Fig. 13, and the results of the capacity estimates and time required are detailed in Table 4.

The results in Table 4 show that PBProbe can measure at least 85% of the bottleneck capacity of all tested links, but Pathrate is more accurate and measures at least 90% of the bottleneck capacity of all links. However, for links with long delay and/or high utilization, Pathrate required more than $1,000$ seconds to estimate the capacity. The reason is that if the distribution of the measured dispersion is not unimodal after the first phase, Pathrate will start the second phase to probe the network using different packet train lengths and packet sizes. Once Pathrate enters the second phase, it takes a long time to determine the correct link capacity. As a result, the technique converges rapidly if the dispersion distribution is unimodal in the first phase; otherwise, it is much

29

Table 4: Comparison of PBProbe and Pathrate on Internet links. (Capacity: Mbps; Time: seconds)

	PBProbe		Pathrate	
	Capacity	Time	Capacity	Time
CalTech → UCLA (1Gbps)	919.6	14	933.2	17
UCLA → CalTech (1Gbps)	839.4	14	945.3	1146
GaTech → UCLA (1Gbps)	928.1	14	932.9	18
UCLA → GaTech (1Gbps)	840.3	14	968.1	1223
PSC → GaTech (1Gbps)	959.9	13	995.4	1122
GaTech → PSC (1Gbps)	905.9	13	947.0	17
PSC → UCLA (1Gbps)	889.6	14	935.6	20
UCLA → PSC (1Gbps)	845.2	15	905.6	20
NTNU → UCLA (600Mbps)	580.6	20	575.6	1641
UCLA → NTNU (600Mbps)	588.4	21	573.4	1641

Table 5: Comparison of PBProbe and Pathrate overhead on Internet gigabit links.

	GaTech → UCLA		UCLA → GaTech	
	PBProbe	Pathrate	PBProbe	Pathrate
spent time	14 sec	18 sec	14 sec	1223 sec
total packets	20,213	2,414	20,213	27,630
total bytes	30,319,500	3,543,752	30,319,500	39,707,740
BW consumption	2.166Mbps	1.575Mbps	2.166Mbps	0.260Mbps

slower than PBProbe.

We also compared the packet overhead incurred by PBProbe and Pathrate on high speed Internet links. Table 5 shows the comparison on one of the high speed links, the UCLA - GaTech link. From the experiment results, we observe that PBProbe is more expensive in terms of computation time than Pathrate, if the latter only uses one phase to estimate the capacity of high speed links. In the case where Pathrate is required to enter the second phase, PBProbe and Pathrate produce comparable amounts of packet overhead. However, since Pathrate slows down substantially after entering the second phase, the bandwidth consumption (i.e., bits per second) is much smaller than that of PBProbe. It is worth noting that, even though the bandwidth consumption of PBProbe (approximately 2 Mbps) seems relatively high, in reality, it is only 0.2% of the bottleneck capacity (i.e., 1 Gbps); thus, it does not intrude on other traffic flows in the network.

The experiment results suggest that there are trade-offs between PBProbe and Pathrate for high-speed path capacity estimation. On the one hand, PBProbe yields very good estimation results rapidly (less than 20 seconds in most cases). Given more time, it will progressively improve the estimates, since better samples can be obtained. On the other hand, Pathrate tends to produce accurate results, but the required time may vary from approximately 20 seconds to 20 minutes. Therefore, it may not be ideal for scenarios where the bottleneck capacity must be estimated in a very short time.

It should also be noted that the packet overhead of PBProbe is proportional to the employed bulk length k. While measuring a high speed link, PBProbe increases its bulk length and in turn increases the packet overhead to compensate for the limited support of the system's timer granularity. Nonetheless, thanks to the U parameter employed, the bandwidth consumption of PBProbe is restricted by the upper bound of the utilization. Hence, PBProbe can carefully control the trade-off between the bandwidth consumption and the time needed to satisfy the requirements of different applications.

7   Conclusion

We have studied a classic problem of link capacity estimation, and proposed a technique, called PBProbe, for estimating the bottleneck capacity of emerging high speed links. PBProbe is based on the CapProbe algorithm, but it uses a "packet bulk" to adapt the number of packets in each probing according to different network characteristics. As a result, it preserves the simplicity, speed, and accuracy of CapProbe, and compensates for the poor system timer granularity problem on high speed links. Using extensive analysis, emulation, and Internet experiments, we evaluated the accuracy, speed, and overhead of PBProbe in various network configurations. The results show that PBProbe can correctly and rapidly estimate the bottleneck capacity in almost all test cases. Compared to other capacity estimation techniques, PBProbe is ideal for real

deployments that require online and timely capacity estimation. The proposed technique can improve various applications, such as peer-to-peer streaming and file sharing, overlay network structuring, pricing and QoS enhancements, as well as network monitoring.

8   Acknowledgments

References

[1] Abilene network traffic. http://loadrunner.uits.iu.edu/weathermaps/abilene/.

[2] Cenic network statistics. http://cricket.cenic.org/grapher.cgi.

[3] Clink: a   tool   for   estimating   internet   link   characteristics. http://allendowney.com/research/clink/.

[4] Garr-b network weather map. http://www.noc.garr.it/mappe/backbone.shtml.

[5] Grnet: Network weathermap. http://netmon.grnet.gr/map.shtml.

[6] Nistnet: network emulation package. http://www.antd.nist.gov/itg/nistnet/.

[7] pchar: A tool for measuring internet path characteristics. http://www.kitchenlab.org/www/bmah/Software/pchar/.

[8] Poisson traffic generator. http://www.spin.rice.edu/Software/poisson_gen/.

[9] Twaren weather map. http://mrtg.twaren.net/mrtg/wmap/.

[10] L.-J. Chen, T. Sun, L. Lao, G. Yang, and M. Y. S. M. Gerla. Estimating link capacity in high speed networks. In *IFIP Networking*, pages 98–109, 2006.

[11] L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla. Adhoc probe: Path capacity probing in ad hoc networks. In *International Conference on Wireless Internet*, pages 156–163, 2005.

[12] L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla. End-to-end asymmetric link capacity estimation. In *IFIP Networking*, pages 780–791, 2005.

[13] L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla. Monitoring access link capacity using tfrc probe. *Computer Communications*, 29(10):1605–1613, June 2006.

[14] C. Dovrolis, P. Ramanathan, and D. Moore. What do packet dispersion techniques measure? In *IEEE Infocom*, pages 905–914, 2001.

[15] S. Floyd and V. Paxson. Difficulties in simulating the internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, August 2001.

[16] V. Jacobson. Pathchar: A tool to infer characteristics of internet paths. ftp://ftp.ee.lbl.gov/pathchar/.

[17] G. Jin and B. Tierney. System capability effect on algorithms for network bandwidth measurement. In *ACM SIGCOMM Conference on Internet Measurement*, pages 27–38, 2003.

[18] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi. Capprobe: A simple and accurate capacity estimation technique. In *ACM SIGCOMM*, pages 67–78, 2004.

[19] R. Kapoor, L.-J. Chen, M. Y. Sanadidi, and M. Gerla. Accuracy of link capacity estimates using passive and active approacheswith capprobe. In *IEEE Symposium on Computers and Communications*, volume 2, pages 1085–1090, 2004.

[20] K. Lai and M. Baker. Measuring bandwidth. In *IEEE Infocom*, pages 235–245, 1999.

[21] K. Lai and M. Baker. Measuring link bandwidths using a deterministic model of packet delay. In *ACM SIGCOMM*, pages 283–294, 2000.

[22] K. Lakshminarayanan, V. N. Padmanabhan, and J. Padhye. Bandwidth estimation in broadband access networks. In *ACM SIGCOMM Conference on Internet Measurement*, pages 314–321, 2004.

[23] S.-J. Lee, P. Sharma, S. Banerjee, S. Basu, and RodrigoFonseca. Measuring bandwidth between planetlab nodes. In *International Workshop on Passive and Active Network Measurement*, pages 292–305, 2005.

[24] Y. Lin, H. Wu, S. Cheng, W. Wang, and C. Wang. Measuring asymmetric link bandwidths in internet using a multi-packet delaymodel. In *IEEE International Conference on Communications*, volume 3, pages 1601–1605, 2003.

[25] M. J. Luckie, A. J. McGregor, and H.-W. Braun. Towards improving packet probing techniques. In *ACM SIGCOMM Workshop on Internet Measurement*, pages 145–150, 2001.

[26] J. Mogul and S. Deering. Path mtu discovery. IETF RFC 1191, November 1990.

[27] J. Postel. Internet protocol. IETF RFC 791, September 1981.

[28] R. Prasad, M. Jain, and C. Dovrolis. Evaluating pathrate and pathload with realistic cross-traffic, 2003 Bandwidth Estimation Workshop.

[29] S. Saroiu, P. K. Gummadi, and S. D. Gribble. Sprobe: A fast technique for measuring bottleneck bandwidth in uncooperativeenvironments. http://sprobe.cs.washington.edu/.

[30] S. Savage. Sting: a tcp-based network measurement tool. In *USENIX Symposium on Internet Technologies and Systems*, volume 2, pages 7–7, 1999.