UCLA Computer Science Department Technical Report CSD-TR No. 040001

CapProbe: A Simple and Accurate Technique to Measure Path Capacity

Rohit Kapoor, Ling-Jyh Chen, M. Y. Sanadidi, Mario Gerla
University of California, Los Angeles, CA 90095, USA
{rohitk, cclljj, medy, gerla} @cs.ucla.edu

Abstract: Measuring the capacity of the narrow link in an Internet path in an unobtrusive and end-to-end manner is a fundamental issue and has received a lot of attention in previous literature. Most attempts have been based on packet dispersion measurements, but they have been so far not very accurate. Both 'Packet Pairs' and 'Packet Trains' have been shown to be inadequate. In this work, we extend the packet dispersion paradigm as a result of a fundamental *queuing delay* observation. Our technique, called CapProbe, makes use of not only the *dispersion* between packets of the Packet Pair, but also the *actual delays* of the packets. Using various simulation experiments, we show that our technique works across a very broad range of cross-traffic scenarios. In fact, the only case where our technique does not work is when the cross-traffic is predominantly non-reactive (like UDP). Even in this case, the scheme fails only when the UDP traffic saturates the bottleneck link, a very unlikely event in the current Internet dominated by TCP applications.

1. INTRODUCTION

A fundamental problem in the area of computer network measurements is determining the capacity of an Internet path in a non-disruptive and accurate manner. The capacity of a path is defined as the minimum *physical* bandwidth among all links traversed by a path. Such a link has been called the *narrow link*. Note that such capacity is clearly different from the *available bandwidth*, which is the minimum of the unused capacities among the links of a path. The capacity is fixed for a path, whereas the available bandwidth is time varying. Our focus in this paper is on measuring the *capacity* of a path.

Accurate knowledge of the capacity of a path is valuable to service providers and users alike. While Internet service providers can use capacity estimates to detect either congested or unutilized links, applications running on users' terminals can use capacity estimates to enhance their performance. A clear example of this is in the case of TCP Westwood [1] [11] [12], where various bandwidth measures are combined to obtain tighter congestion control.

While cooperation from Internet routers in measuring the capacity could certainly lead to very accurate estimates (an example being MRTG [2]), this would require access to the behavior of routers. An end-to-end approach, on the other hand, is expected to be less accurate, but is often the only feasible methodology for a path that may cross several ISP networks. In this paper, we present CapProbe, an end-to-end capacity estimation approach that is simple, not based on heuristics, and extremely accurate across a range of system parameters.

Previous work on estimating the capacity of a path has mostly been based on using dispersions of packet pairs or trains. In [3], Carter and Crovella proposed *bprobe*, in which filtering methods are applied on packet pair measurements, consisting of different packet sizes. In [4], Lai used packet pair measurements, but filtered them through a kernel density estimator. Lai also filtered packet pair measurements, whose *potential bandwidth*, derived from the minimum separation of the packet pair at the sender, was less than the measured bandwidth. Such measurement samples clearly correspond to time-compressed packets. The underlying assumption of all these techniques is that the distribution of measurements obtained from packet pair samples is unimodal, i.e., the sample with the maximum frequency of occurrence corresponds to the capacity.

Paxson showed in [5] that this distribution can in often be multimodal. He identified multi-channel links as a failure case of packet pairs and presented the Packet Bunch Modes (PBM) technique to overcome this limitation. The PBM methodology consists of sending packet trains of different lengths in response to a distribution with multiple modes, treating multiple modes as corresponding to multi-channel links. Dovrolis et al [6] elaborated further on the occurrence of multiple modes. They showed that the strongest mode in the multimodal distribution may correspond to the capacity, or to an under- or over-estimate of the capacity. Under-estimation occurs when the network is heavily congested, while over-estimation occurs, to various degrees, when the narrow link is followed by links of higher capacity, referred to as *Post-Narrow Links*. They also observed that a packet train consisting of N packets is most useful for estimating capacity when N = 2, corresponding to a packet pair, since interference from cross-traffic is likely to increase as N increases. In fact, when N is sufficiently large, the distribution becomes unimodal. The resulting mode corresponds to the so-called Asymptotic Dispersion Rate (ADR), for which no physical interpretation was given. Finally, they presented a capacity estimation methodology, which first sends packet pairs. If this yields a multimodal distribution, then probing with packet trains with an increasing value of N is initiated. For some value of N, the distribution becomes unimodal, and the capacity is selected as the next highest mode after this mode in the multimodal distribution that was obtained from packet pairs.

A different technique, not based on dispersion of packet pairs, but rather on the variation of the round-trip delay as the packet size increases, was used by Jacobson in pathchar [7]. This technique, based on the generation of ICMP replies from routers, is known to have scalability problems. The overhead associated with this method can be quite high since it tries to estimate the capacity of every link on the path in order to estimate the path's capacity. Packet tailgating is another technique proposed by Lai [8]. This technique is divided into two phases: the Sigma phase, which measures the characteristics of the entire path, and the Tailgating phase, which measures the characteristics of each link individually. However, the reported measurements are still often inaccurate, possibly due to accumulation of errors in capacity estimates as measurements proceed along the path.

Our capacity estimation technique, CapProbe, is based on a simple and fundamental observation*: A packet pair measurement corresponding to either an over-estimated or an under-estimated capacity suffers cross-traffic induced queuing at some link.* Exploiting this observation, we develop a technique combining dispersion and delay measures to filter out packet pair samples that were "distorted" by cross-traffic. Simulation and measurement results show that CapProbe is accurate over almost all environments, except for intensive non-responsive (UDP) traffic that nearly saturates the narrow link. In this last case however, all other dispersion-based methods fail as well.

We describe this idea in more detail in Section 2 and show how it can be used with existing packet pair techniques to estimate capacity accurately in a simple, quick and non-disruptive manner. Section 2 also describes scenarios under which under- and over-estimation of capacity can occur. Section 3 presents simulation and measurement experiments to study the accuracy of our technique. Section 4 concludes the work and outlines future studies in this area.

2. The Capacity Estimation Problem

Relying on Packet Pair dispersion to estimate path capacity can actually lead to either under-estimation or over-estimation of capacity. We now discuss the scenarios which lead to such distortion.

2.1 Over-Estimation of Capacity

Over-estimation occurs when the narrow link is not the last one on the path, i.e., when so-called *post narrow* links are present. The presence of these links can reduce the packet pair dispersion created by the narrow link if the first packet of the pair queues at a post-narrow link, while the second does not (or experiences queuing for a shorter time than the first packet). In this case, the dispersion between the packet pair is smaller than that created by the narrow link, leading to an over-estimation of capacity. Note that the queueing of the first packet in this case is caused by interference from cross-traffic. This behavior, termed "compression" by previous researchers, is more pronounced when the probe packets are smaller than cross-traffic packets and as the cross-traffic rates increase.

Fig 1 shows how capacity can be over-estimated. At the post-narrow link, the dispersion of the two packets is reduced since the first packet is queued, but the second one is not.

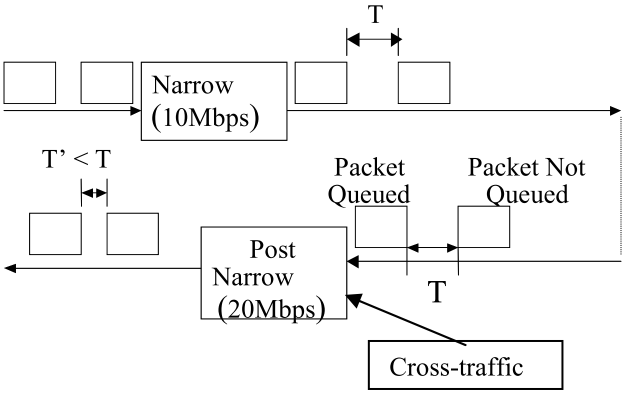


Fig 1: Over-estimation of Capacity

The key observation here is that when capacity over-estimation happens, the *first packet* of the packet pair will have queued at a post-narrow link *due to interference from cross-traffic*.

2.2 Under-Estimation of Capacity

Under-estimation occurs when cross-traffic packets are served (transmitted) in between packets of the packet pair samples. This increases the dispersion of the packet pair and leads to a lower capacity estimation. Fig 2 shows how under-estimation of capacity can occur. This under-estimation of capacity is more pronounced as the cross-traffic rate increases, and when the underlying expansion of the dispersion is not counter-balanced by compression after the narrow link.

The key observation here is that when under-estimation occurs, the *second packet* of the packet pair will have queued *due to interference from cross-traffic*. Note that the second packet can also experience queuing delay due to the first packet. Such delay is different from that induced due to cross-traffic and does not distort the dispersion.

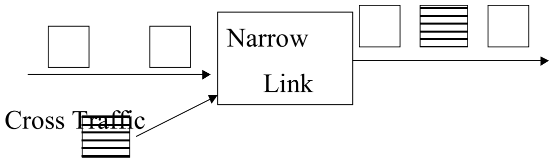


Fig 2: Under-estimation of Capacity

2.3 CapProbe

We present a new capacity estimation technique, which we call CapProbe that is based on the association of increased queuing delay (resulting from cross-traffic) with capacity estimation errors as discussed above. CapProbe combines dispersion as well as delay measurements of packet pair probes. As we now show, using both dispersion and delay together, one can go beyond existing packet pair techniques in providing accurate capacity estimates.

When packet dispersion under-estimates or over-estimates capacity, at least one packet of the packet pair involves cross-traffic induced queueing. Thus, whenever an incorrect value of capacity is estimated, the *sum of the delays* of the packet pair packets includes cross-traffic induced queuing delay.

On the other hand, when the correct value of capacity is estimated, it is *not necessary* for either packet of the packet pair to experience cross-traffic induced queueing delay. Of course, even when dispersion estimates capacity correctly, both packets of the packet pair *could* be delayed due to cross-traffic *by the same amount*. In this case, dispersion will measure capacity accurately, but delays of packets will include cross-traffic induced queuing delay.

If we now make the assumption that at least one sample of the packet pairs goes through without cross-traffic interference, we get a sample that *measures the correct capacity* and *does not experience cross-traffic queuing*. The sum of delays of the two packet pair packets for this sample will not involve any cross-traffic queuing delay. This sum will, thus, be the minimum value for the sum of the delays of the two packets among all packet pair samples.

Our technique, CapProbe, calculates the sum of delays of the two packets for all samples of the packet pair. The dispersion measured from the sample corresponding to the minimum over all "delay sums" reflects the narrow link capacity.

To illustrate, consider the set of packet pair samples, i = 0, 1, 2… Let d_i represent the 'delay sum', i.e., the sum of the delays of the first and second packets of the packet pair i. Let L be the size of the packet pair packets.

The dispersion τ_i for a packet pair sample is defined as the difference between the delays of the two packets of the packet pair sample. We determine the minimum d_i for subset of samples, having equal value of sample dispersion τ_k. Let the minimum delay obtained be d_k. Thus,

$$d_k = \min \{d_i, \text{ for all i such that } \tau_i = \tau_k\}$$

In Fig 3, we plot d_k versus bandwidth estimates, $B_k = L/ \tau_k$. The plot shows that the bandwidth estimate equal to the narrow link capacity of 4Mbps is obtained at the minimum value of d_k.

Note that for some bandwidth values, the minimum delay is not shown. No packet pair samples corresponding to these bandwidth values were obtained.

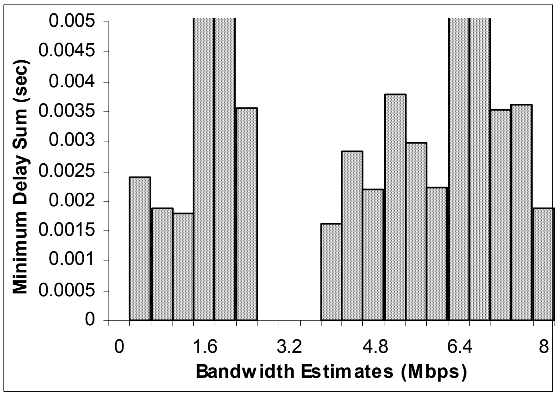


Fig 3: Minimum delay sums corresponding to different bandwidth estimates

CapProbe is based on the assumption that at least one packet pair sample with the appropriate minimum d_k is received at the destination. In a network such as the Internet in which the traffic intensity varies due to reactive TCP flows, there is very high likelihood of obtaining one or more of the desired samples. In fact, as we show in our simulation and testbed experiments later, we found very few cases that are deprived of such samples. The only cases in which these samples may not be obtained correspond to a highly congested (almost 100% congested), UDP-predominant (i.e., non-reactive) network, in which all packet pair measurements techniques will fail anyway.

3. Results

In this section, we show results of simulations and measurement experiments to test our scheme. The network topology used in our simulations consists of a six-hop path with capacities {10, 7.5, 5.5, 4, 6 and 8} Mbps. The narrow link on the path is 4Mbps. NS-2 [10] was used to perform the simulations.

The source of the packet pair is at the 10Mbps link and the destination is at the 8Mbps link. The capacity is measured at the destination. Note that though we are performing this measurement one-way, the same technique could be applied "round-trip". In such a mode, the source sends the packet pair as well as receives the reply and performs the measurements with a straightforward extension of the method used here.

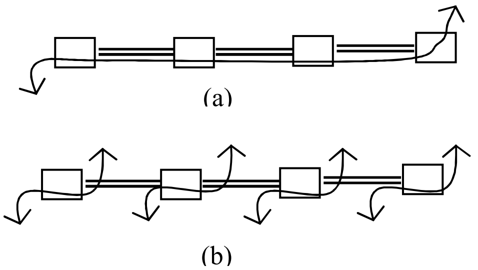


Fig 4: (a) Persistent traffic (b) Non-persistent traffic

The cross-traffic on the path can be either persistent (Fig 4 (a)) or non-persistent (Fig 4 (b)). The different traffic types we used for the cross-traffic were TCP, UDP and Pareto. The Pareto traffic consisted of 16 Pareto sources with parameter alpha = 1.9 (this is known to result in Long Range Dependent traffic [9]). In each set of experiments, we increased the rate of the cross-traffic from 1Mbps to 4Mbps, which is the capacity of the narrow link. Cross-traffic links were used to limit the amount of cross-traffic on our test path. The packet pairs were UDP packets, sent at intervals of 100m sec. Unless stated otherwise, the packet size for the packet pair was 200 bytes. The simulation time was 30 sec. The size of cross-traffic packets was 500 bytes.

We study below the packet pair delay sum statistics. In particular, the minima of such delays obtained for various bandwidth estimates. We also present corresponding bandwidth estimate distributions for same scenarios of path and cross-traffic flows. The results reveal the accuracy of CapProbe capacity estimates.

3.1 Persistent Cross-Traffic

3.1.1 Responsive (TCP) Cross-Traffic

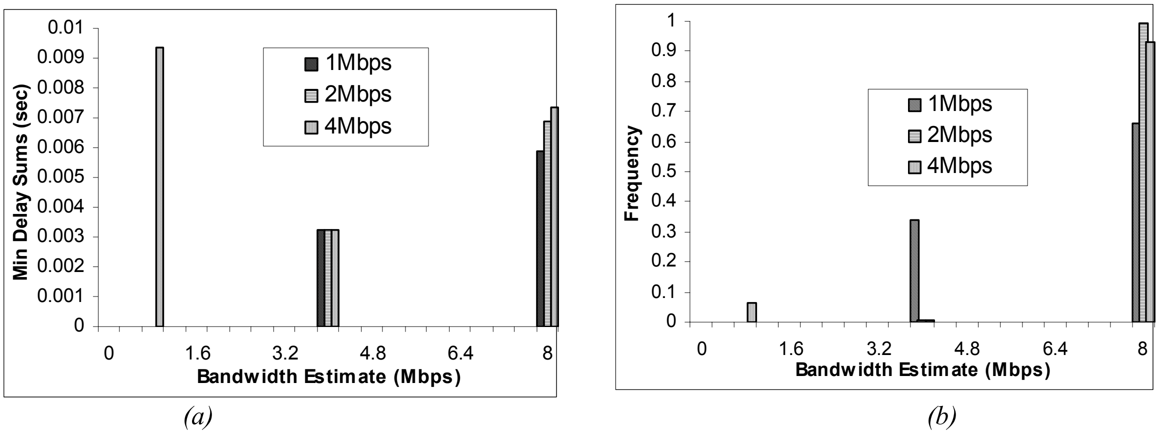


Fig 5: (a) Minimum delay sums and (b) frequency of occurrence when cross-traffic is TCP

Fig 5 (a) shows the minimum packet pair delay sums. The index in the figures shows maximum cross-traffic rates[1]. Fig 5 (b) shows the frequency distribution of bandwidth estimates from packet dispersion.

We make the following observations from the graphs:

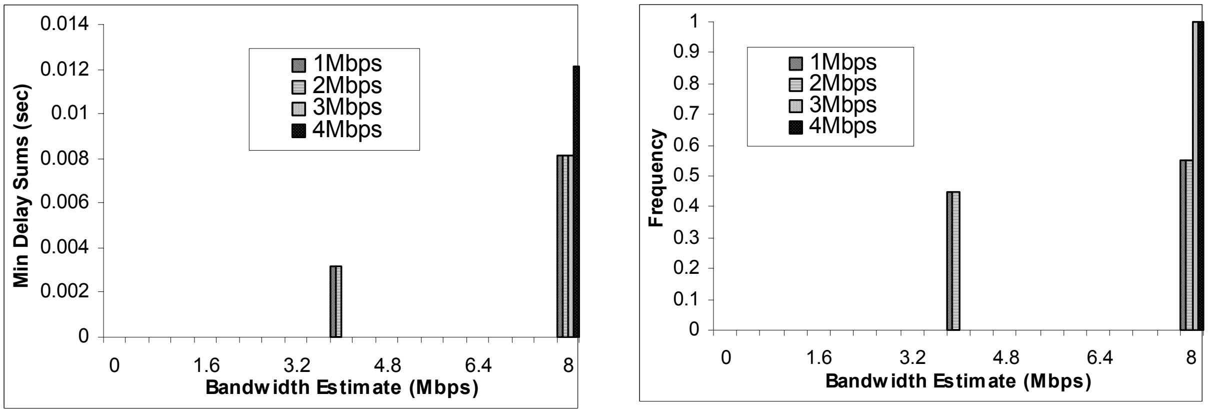
- The value of minimum packet pair delay sums is smallest at the point corresponding to the narrow link capacity, which is 4Mbps. We note here that when cross-traffic rate is 4Mbps, i.e., equal to the narrow link capacity, CapProbe still works. Thus, when cross-traffic is TCP, our technique measures the right capacity even for highly congested links.
- Looking at Fig 5(b), the strongest mode always occurs at 8Mbps. This mode is the Post-Narrow Capacity Mode (PNCM) introduced in [6] and represents compression impact.

[1] TCP rates fluctuate, but will be limited by cross-link speeds that vary between 1Mbps and 4Mbps.

3.1.2 Non-Responsive CBR Cross-Traffic

We now show simulation results when the cross-traffic is CBR using UDP as transport layer protocol. Fig 6 (a) shows the minimum packet pair delay sums. Fig 6 (b) shows the frequency distribution of bandwidth estimates from packet dispersion.

The UDP cross-traffic results are different from TCP results since UDP is not reactive to congestion.



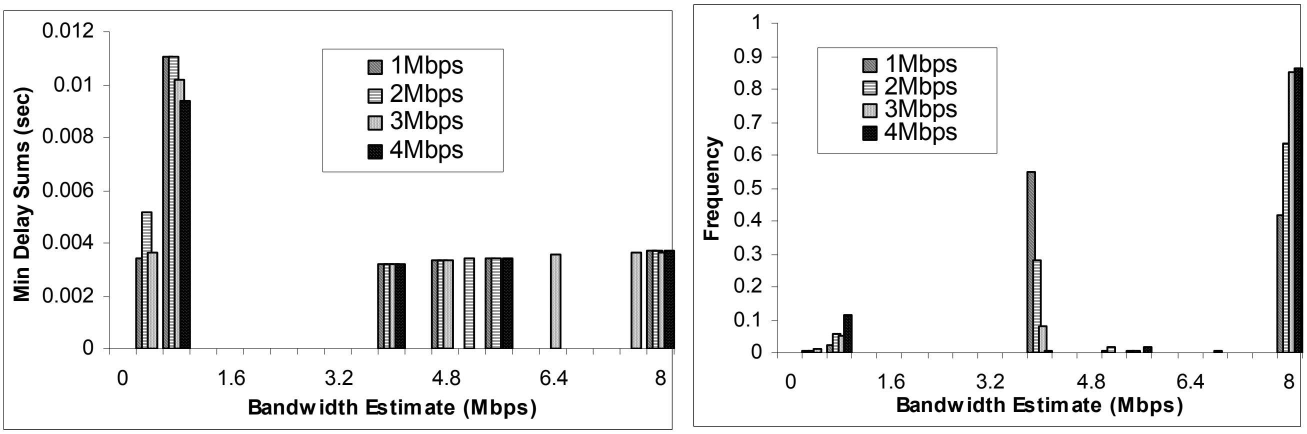
(a) *(b)*

Fig 6: (a) Minimum delay sums and (b) frequency of occurrence when cross-traffic is UDP

- ➢ From Fig 6 (a), CapProbe predicts the correct capacity till the UDP cross-traffic is 2Mbps, i.e., till a load of 50% on the narrow link. For higher cross-traffic rates, no samples corresponding to the correct capacity are obtained. In fact, all samples for cross-traffic of 3Mbps or 4Mbps have a packet dispersion corresponding to an estimated bandwidth of 8Mbps. This is basically the PNCM mode described in [6].
- ➢ It should be noted here that high-rate UDP is a worst-case for our technique (and in general, for all packet pair-based techniques). Such cross-traffic is also not very realistic. Our basis for experimenting with such cross-traffic is to identify cases in which our technique fails.
- ➢ From Fig 6 (b), the strongest mode corresponds to PNCM at 8Mbps, again resulting from compression.

3.1.3 Pareto Traffic

We now show the results when the cross-traffic is Long Range Dependent, i.e., consists of 16 Pareto sources, that are using UDP as the transport protocol. Fig 7 (a) shows the minimum packet pair delay sums. Fig 7 (b) shows the corresponding frequency distribution of bandwidth estimates.



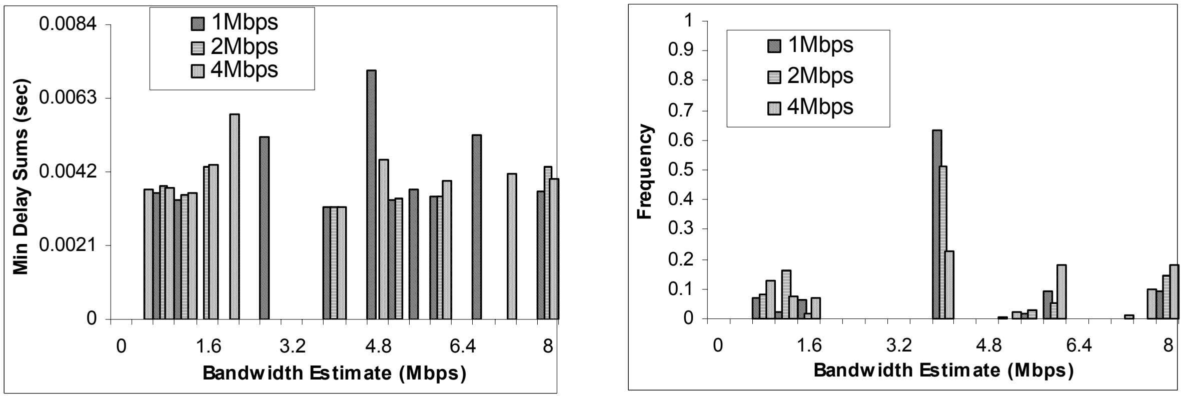
(a) *(b)*

Fig 7: (a) Minimum delay sums and (b) frequency of occurrence when cross-traffic is Pareto

In the case of Pareto traffic, our method predicts the right capacity upto 4 Mbps cross-traffic, which is 100% congestion on the narrow link. The strongest mode again fails to estimate the capacity in most cases.

3.2 Non-Persistent Cross-Traffic

3.2.1 Responsive (TCP) Cross-Traffic



(a) *(b)*

Fig 8: (a) Minimum delay sums and (b) frequency of occurrence when cross-traffic is TCP

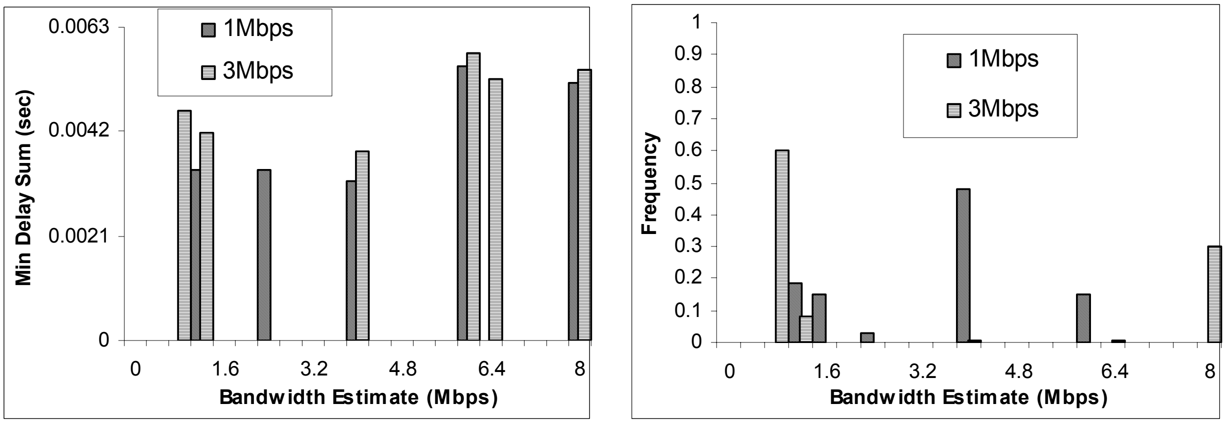
Fig 8 (a) shows the minimum packet pair delay sums. Fig 8 (b) shows the frequency distribution of bandwidth estimates from packet dispersion. From the plots, both the CapProbe and the strongest mode yield the correct capacity for all cross-traffic values. As stated earlier, these results are for a packet pair packet size of 200 bytes.

We now show results when size of packet pair packets is increased to 500 bytes. Fig 9 (a) shows the minimum packet pair delay sums. Fig 9 (b) shows the frequency distribution of bandwidth estimates from packet dispersion.

➢ From Fig 9 (b), when the size of packet pair packets is similar to that of cross-traffic packets, results show the emergence of the ADR (Asymptotic Dispersion Rate), which has been noted in [6]. The strongest mode is the ADR when cross-traffic is 3Mbps.

➢ From Fig 9 (a), the CapProbe scheme still estimates the right capacity for different cross-traffic values. Thus, even when the strongest mode is the ADR, our CapProbe scheme is able to estimate the correct capacity.

➢ Note that the smallest value of the minimum packet delay sums has a value of 0.003214 sec when cross-traffic rate is 1Mbps. The narrow link capacity of 4Mbps is correctly estimated in this case. When cross-traffic rate is 3Mbps, the smallest value of the minimum packet delay sums is a little higher at 0.003796 sec, but still the correct capacity of 4Mbps is estimated. This minimum delay sums value includes a small amount of cross-traffic induced queuing delay. Thus, the requirement of obtaining a single packet pair sample involving no cross-traffic queuing delay is *sufficient, but not necessary*. While CapProbe is certain to estimate the correct capacity when a packet pair sample involving no cross-traffic is obtained, it is likely to work even in cases where the packet pair sample involves very little cross-traffic induced queuing delay, as for the 3Mbps cross-traffic case.



(*a*) (*b*)

Fig 9: (a) Minimum delay sums and (b) frequency of occurrence when cross-traffic is TCP

Non-Responsive Pareto/CBR Cross-Traffic

We do not present the results of these simulations. These simulations also resulted in CapProbe estimating the correct capacity in most cases. Apart from the case of very intensive UDP cross-traffic, the CapProbe technique estimated the narrow link capacity accurately.

3.3 Measurement Experiments

We also performed Internet experiments to validate our scheme. Fig 10 shows the testbed configuration. We sent ICMP PING packets from a source machine in our lab to www.yahoo.com. These packets were forced to go through a machine running DummyNet, on which the narrow link was created. The narrow link capacity on DummyNet was varied from 500Kbps to 20Mbps. Beyond the latter capacity, we were not sure if DummyNet would still be the narrow link.

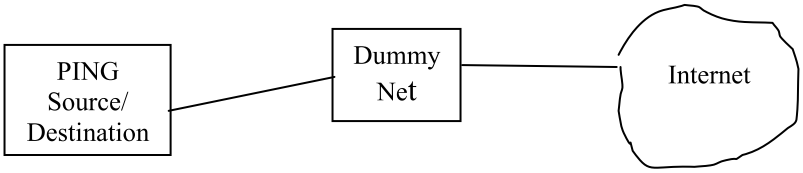


Fig 10: Testbed for measurements

Table 1 shows the results obtained from varying the narrow link through DummyNet. Capacity estimate results are presented for the CapProbe technique. The capacity indicated by the strongest mode for the bandwidth distribution is also shown. Also shown are the differences (in percentage) between the estimated capacity and the actual narrow link capacity. The results show a 50-80% reduction in estimation errors when using CapProbe compared to estimates indicated by the strongest mode.

DummyNet Link	CapProbe (kbps)	Strongest Mode (kbps)
500Kbps	483.383 (- 3.32 %)	469.07 (- 6.19 %)
1Mbps	941.619 (- 5.84 %)	860.47 (- 13.95 %)
5Mbps	4576.659 (- 8.47 %)	4109.85 (- 17.80 %)
10Mbps	10416.666 (+ 4.17 %)	7691.32 (- 23.09 %)
20Mbps	21038.67 (+ 5.19 %)	17059.03 (- 14.70 %)

Table 1: Results of testbed experiments

4. CONCLUSION

In this work, we presented a new technique we call CapProbe to estimate the capacity of an Internet path. This technique combines packet pair dispersion and delay measurements resulting in a simple and accurate end-to-end method to estimate capacity.

Simulations show that the scheme works over a wide range of path and cross-traffic conditions. In fact, the only case where it fails to determine the right capacity is when cross-traffic is predominantly CBR and nearly "saturates" the narrow link. In the more prevalent environment of congestion-controlled reactive flows, our scheme always works. Compared to a vanilla Packet Pair scheme that selects the strongest mode as the capacity, our scheme is significantly and consistently more accurate. In fact, the capacity indicated by the strongest mode is incorrect in a rather large number of cases, being equal to either the ADR (Asymptotic Dispersion Rate) or the PNCM (Post Narrow Compression Mode).

The strength of our scheme is based on its simplicity (e.g., relative to other schemes such as pathchar, pathrate, packet tailgating) and robustness. In fact, for an accurate estimate, it suffices that a single Packet Pair sample involving no cross-traffic induced queuing delay is obtained. Obtaining such a sample in today's Internet, dominated by reactive (TCP) flows, a practical certainty. Only extremely intense non-reactive cross-traffic can prevent the outcome of one such sample.

For the future, we are planning extensive Internet experiments to test our scheme in "actual" traffic loads on Internet paths. We are also looking at various applications of CapProbe. One potential application is its use within TCP for enhanced congestion control. Another application is estimating the wireless link capacity of a mobile user. Such estimation can be used by multimedia streaming servers to determine appropriate streaming rates to clients.

References

[1] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, *"TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links"*, In Proceedings of ACM Mobicom 2001, pp 287-297, Rome, Italy, July 16-21 2001.

[2] T. Oetiker, *"MRTG: Multi Router Traffic Grapher"*, http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html

[3] R. Carter and M. Crovella, *"Measuring Bottleneck Link Speed in Packet-Switched Networks",* Performance Evaluation, Vol. 27-8, pp. 297-318, Oct. 1996.

[4] K. Lai and M. Baker, *"Measuring Bandwidth"*, In Proceedings of IEEE INFOCOM '99, New York, NY, March 1999.

[5] V. Paxson, *"Measurements and Dynamics of End-to-End Internet Dynamics"*, Ph.D. thesis, Computer Science Division, Univ. Calif. Berkeley, April 1997.

[6] C. Dovrolis, P. Ramanathan, and D. Moore, *"What do packet dispersion techniques measure?"*, in Proceedings of IEEE Infocom'01, Anchorage, Alaska, April 22--26 2001.

[7] V. Jacobson, *"Pathchar: A tool to infer characteristics of Internet paths"*, ftp://ftp.ee.lbl.gov/pathchar/

[8] K. Lai, M. Baker, *"Measuring Link Bandwidth Using a Deterministic Model of Packet Delay"*, SIGCOMM 2000.

[9] M. S. Taqqu, W. Willinger, and R. Sherman, *"Proof of a fundamental result in self-similar traffic modeling"*, ACM/SIGCOMM Computer Communications Review, 27:5--23, 1997.

[10] Network Simulator NS-2, www.isi.edu/nsnam/ns

[11] R. Wang, M. Valla, M.Y. Sanadidi, B. K. F. Ng, M. Gerla, *"Efficiency/Friendliness Tradeoffs in TCP Westwood"*, Seventh IEEE Symposium on Computers and Communications, Taormina, Italy, 1- 4 July, 2002.

[12] R. Wang, M. Valla, M. Y. Sanadidi, and M. Gerla, *"Adaptive Bandwidth Share Estimation in TCP Westwood"*, In Proc. IEEE Globecom 2002, Taipei, Taiwan, R.O.C., November 17-21, 2002.