TCP Probe: A TCP with built-in Path Capacity Estimation[1]

Anders Persson, Cesar A. C. Marcondes[2], Ling-Jyh Chen, Li Lao, M. Y. Sanadidi, and Mario Gerla
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90095, USA
Email: {anders,cesar,cclljj, llao, medy,gerla}@cs.ucla.edu

Abstract – **Knowing the capacity of an Internet path is important for efficient network utilization, pricing, and management. Using the capacity information, one can provide better TCP congestion control, multimedia streaming, P2P peer selection, and overlay structuring. Capacity estimation has been extensively studied. Though current approaches have been able to provide fast and accurate capacity estimates, they are mostly "active" in nature (ie, they utilize extra probing packets) and thus tend to be intrusive. In this paper, we propose TCP Probe, a "passive" capacity estimation extension to TCP, to accurately estimate bottleneck link capacity of an Internet path. The TCP Probe extension is packet-pair based, and it is applicable to all TCP variants. Using simulation and Internet measurements, we show that TCP Probe is able to correctly measure bottleneck capacity of a path. Moreover, we present a simple application of TCP Probe where TCP is forced to enter slow start phase when a drastic capacity change from LOW to HIGH is detected. The results show that the capacity estimate provided by TCP Probe enables it to take better advantages of the capacity increase than the original TCP. In summary, TCP Probe is simple, passive, and accurate, and it is applicable to a broad variety of TCP variants.**

Keywords – **TCP Probe, passive capacity estimation, Internet measurements**

1 Introduction

Knowing the capacity of an Internet path is important for network utilization, pricing, and management. Using the path capacity information, a multimedia server can determine appropriate streaming rates, while ISPs can monitor characteristics of their links. Moreover, emerging Internet applications, such as P2P and overlay networks, can use this information to more effectively utilize the network resources.

Capacity of an Internet path refers to the minimum physical link capacity (i.e. the bottleneck link capacity) among all links on the path. It is different from available bandwidth, which is the minimum of the unused bandwidth on the links in a path. While the available bandwidth is time varying, the capacity is a fixed quantity for a path.

Capacity estimation has been extensively studied, and various capacity estimation techniques have been proposed [2][4][7][8]. The techniques can be categorized into two types, active and passive. Active techniques estimate link capacity by sending out-of-band packets into the network; whereas passive techniques perform capacity estimation by taking advantages of the ongoing data packets without additional overhead to the network. Compared with passive techniques, active techniques usually estimate capacity faster and more accurately. However, they tend to be more intrusive since the out-of-band packets are more likely to congest the network and thus disturb the ongoing foreground applications.

Pathrate [4] and CapProbe [8] are the two most representative active techniques. The main difference between these two techniques is that Pathrate employs packet-train concept to estimate capacity, whereas CapProbe relies on packet-pair. In [8], CapProbe was shown to be faster, simpler, and less intrusive then Pathrate, while keeping the same accuracy as the latter. The simplicity feature enables CapProbe to be easily combined with other data transmission protocols to estimate capacity passively. For instance, TFRC Probe [2] has been proposed to passively estimate link capacity by embedding CapProbe algorithm within TFRC [5], which is a unicast streaming protocol based on UDP. Specifically, TFRC Probe sends some TFRC data packets back-to-back, and it performs CapProbe algorithm to estimate capacity by measuring the delays and dispersions of these back-to-back packets (i.e. probing samples).

Yet, a passive capacity estimation scheme within TCP is still lacking. Some TCP variants (e.g. TCP Westwood [6]) have attempted to approximately estimate the link capacity in order to enhance congestion control. For instance, one of the bandwidth measures TCP Westwood approximates is the link capacity. TCPW obtains such an estimate, called BE from

---

[1] This material is based upon work supported by the National Science Foundation under Grant No. CNS-0435515.

[2] Cesar Marcondes is supported by CAPES Grant/Brazil.

returning ACKs, as shown in Figure 1. However, as we will show later on, BE is not an accurate estimate of bottleneck link capacity in many cases[3]. As a result, an accurate, passive capacity estimation within TCP is still not available.

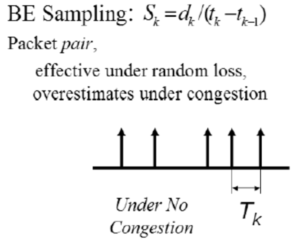

Figure 1: TCP Westwood Bandwidth Estimate (BE)

In this paper, we propose TCP Probe, as a TCP extension, to enable passive capacity estimation within TCP. TCP Probe is CapProbe [8] based, and it overcomes the Delayed Acknowledgements problem by cleverly inverting the order of the packets in the pair before transmission. Through simulation and Internet measurements, we show that TCP Probe is able to accurately measure the link capacity. Moreover, TCP Probe is simple and applicable to any existing TCP variants.

The rest of the paper is organized as follows. In section II, we present our proposed technique, TCP Probe. In section III, we evaluate the accuracy of TCP Probe in estimating link capacity through NS2 simulations and Internet experiments. In section IV, we present an application of TCP Probe by applying "fast rate adaptation" in vertical handoff scenarios. Section V concludes the paper.

2 TCP Probe

We propose TCP Probe as a passive capacity estimation extension embedded within TCP and based on the recently introduced CapProbe technique [8]. The key design principle of such an extension is simplicity, i.e. changes to the existing TCP protocol must be minimal and preferably sender-side only. Moreover, the extension needs to be applicable to any existing TCP variant. We call this extension TCP Probe hereafter.

The basic principle of CapProbe is to send a pair of packets back to back from the source to destination. The pair is then immediately returned to the source. The source measures the delay of each packet as well as the interpacket interval (often referred to as "dispersion"). This experiment is run for many packet pair samples. The source monitors the sum of the packet pair delays. When the sum is minimum (over a sufficiently large number of samples) the corresponding pair yields the path capacity estimate C, where C = packet size/delay interval [8].

In order to mimic CapProbe, TCP Probe must send a portion of the data packets back-to-back as packet pair samples. Fortunately TCP automatically does send some back-to-back data packets, mostly in Slow Start and also sometimes in Congestion Avoidance, upon receipt of an ACK packet. These instances are often sufficient to produce an adequate number of back-to-back samples. If more frequent samples are required, a duplicate packet can be injected by the source after a data packet at the expense of extra traffic. The packets in the pair are ACKed separately. Once the two ACKs of the packet pair are received by the TCP sender, the dispersion information can be used to estimate the path capacity using the previously mentioned CapProbe approach.

However, in order to confine the modifications required in TCP Probe to the sender-side only, and require no change at the TCP receiver, two problems must be solved. One of them is the widely deployed "Delayed ACK" (DelACK) [1], and the other is the potentially differing size of TCP data and ACK packets.

DelACK has been popularly deployed on most of the Internet hosts. A TCP receiver with DelACK installed will acknowledge the received data on every other data packet. Therefore if two TCP data packets *i* and *i+1* are sent back-to-back from the sender, either packet *i* or *i+1* will be acknowledged, as shown in Figure 2-a.

The problem caused by DelACK can be solved by what we call the "inverted packet-pair" technique. When TCP needs to send back-to-back data packets with sequence numbers *i* and *i+1*, it swaps the sending order, i.e., packet `*i+1*' is sent before packet `*i*'. This swapped sending order will generate back-to-back ACKs on sequence numbers *i* and *i+2*. The receiver is, thus, forced to send an individual ACK for each data packet, as shown in Figure 2-b. Note that this enhancement is applicable to all TCP variants.

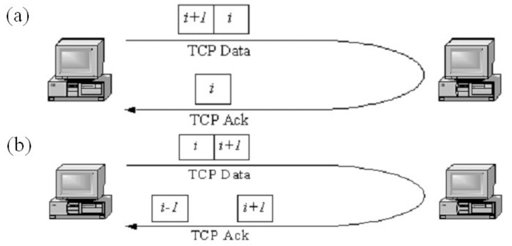

Figure 2: (a) back-to-back TCP packets generate only one ACK because of DelACK; (b) inverted back-to-back TCP Probe packets are separately acknowledged.

The second problem stems from the fact that data and ACK packet sizes are different. Clearly, the TCP receiver cannot enlarge the ACK packet size to make it equal to data packet size. Thus, the difference of TCP data and ACK packet sizes does not comply with the original

---

[3] Note that the inaccuracy of BE has been shown to be immaterial to TCPW performance gain, as BE is not used by TCPW versions such as ABSE, under any conditions in which BE is likely to be inaccurate

CapProbe algorithm assumption, where the packet pair sizes in the forward and backward direction are equal. As a result, the CapProbe equation $C=P/T$ may not hold in our case.

A related problem was addressed in [3], where Chen et al extended CapProbe to estimate asymmetric link capacity by varying the probing packet sizes on the two directions of the path. The proposed method is called AsymProbe. Specifically, suppose the bottleneck link capacities of the forward and backward direction links are C_1 and C_2, and the probing packet sizes on the forward and backward links are P_1 and P_2, AsymProbe measures the forward link capacity (C_1) when $\frac{P_1}{P_2} > \frac{C_1}{C_2}$ and measures the backward link capacity when $\frac{P_1}{P_2} < \frac{C_1}{C_2}$.

Since the size of TCP data and pure ACK packets are bounded (1500 bytes and 40 bytes, including IP header, respectively), TCP Probe thus measures the forward link capacity when $\frac{C_1}{C_2} < \frac{1500}{40} = 37.5$ or the backward link capacity when $\frac{C_1}{C_2} \geq 37.5$. To the best of our knowledge, most of current Internet links are either symmetric or moderately asymmetric with a link capacity ratio smaller than 37.5 (e.g. the down/up link capacities are 1.5Mbps/512Kbps for DSL and 3Mbps/1Mbps for Cable[4]) Therefore, TCP Probe is able to correctly estimate forward link capacity on most Internet connections.

In the rest of the paper, we will present a set of Internet measurements, supported by some simulations, to evaluate the performance of the proposed passive estimation technique, namely TCP Probe, in terms of accuracy and speed.

3 Evaluation

The evaluation of TCP Probe is divided into two parts; (1) simulation and (2) Internet Measurements. We rely on the simulation results to give us some initial insight into the behavior of TCP Probe, and then Internet measurements in order to verify the observed behavior. The performance variables of interest are: (1) Accuracy, (2) Stability and (3) Convergence Speed.

3.1 Simulation

In order to test for accuracy and stability of the two different schemes we evaluated both over a link with varying load to see if the cross-traffic resulted in a change in capacity estimation or increased the variance in the estimate samples. The topology used to perform the simulation can be seen in Figure 3. The capacity estimates were collected by TCP Probe and BE of TCP Westwood, and the TCP Probe extension was added into TCP Newreno.

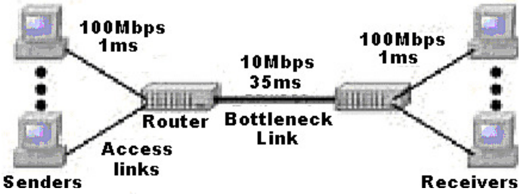

Figure 3: Network topology used in simulation

Due to BE's sensitivity of compression and expansion of packets we expect that an increased load will quickly degrade its performance, both with respect to accuracy and stability. That is, as the load of the link increases, there will be a larger chance of encountering buffered packets on the path, and thus a large chance for compression, resulting in samples reporting higher than expected capacity. Similarly, the chance of having packets being intercepted by other traffic is also likely to increase, resulting in expansion which will lower the sample capacity. Thus, we expect BE to fluctuate between high and low samples, neither of which are accurate.

On the other hand, since TCP Probe uses the CapProbe technique of filtering out packets that have suffered either compression or expansion, we expect that it should be able to maintain a more stable and accurate capacity estimate as the load increases. However, since TCP Probe relies on local minimums to determine good and bad samples, there is an obvious tradeoff between accuracy and speed; increasing the number of samples needed before determining the capacity will result in better accuracy, but a longer convergence time. It should be emphasized that only one good sample is needed for this technique to get a good capacity estimate, which is why we expect it to be more resilient in the presence of cross traffic.

Under no cross traffic, both BE and TCP Probe will correctly estimate the narrow link capacity. However, the more interesting scenario occurs when multiple sender and receiver pairs are added to the simulation. Figure 4 shows the behavior of BE and TCP Probe when there are five TCP NewReno flows sharing the narrow link. The results show that the behavior is as expected, that is, BE has fluctuations and reports an inaccurate capacity while TCP Probe is able to avoid bad samples and is thus able to accurately determine the capacity.

---

[4] The Down/Up link capacities vary by ISPs and areas.

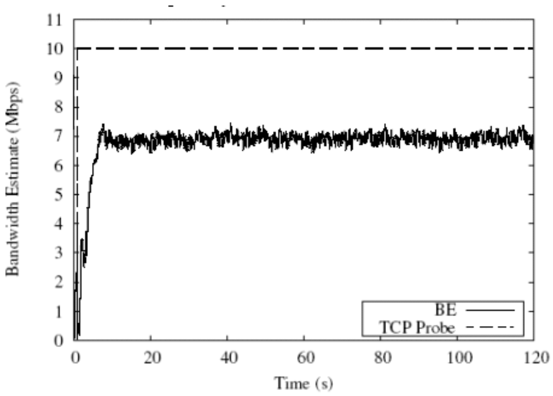

Figure 4: Simulation results showing the behavior of TCP Probe and BE in presence of 5 TCP NewReno flows

3.2 Internet Measurements

In this subsection, we perform Internet experiments to verify the above observations in a more realistic testbed. The selected links are from UCLA to a few Internet hosts that varied with respect to narrow link capacity and load, as shown in Table 1.

Table 1: Links used for Internet measurements

Destination	Narrow Link Capacity
Los Angeles (Cable)	3 Mbps
Tianjin University	45 Mbps
University of Alabama	100 Mbps

For each of the destinations we first observe the behavior of TCP Probe and BE to see how it changes over these different paths. The results are shown in Figure 5, Figure 6, and Figure 7. Of the paths that we tested, the International link leading to China (Figure 6) was the most utilized one, and this can also be observed in the behavior of BE, as it is fluctuating quite heavily over this link. It can also be seen from the results that TCP Probe is able to accurately report the capacity in all cases; however, there is an initial period in which TCP Probe is not obtaining any capacity samples. This initial "void" is due to the fact that TCP Probe requires that multiple packets must be in the TCP source transmit queue in order for a packet pair sample to be successfully transmitted. We defer the study of reducing the “void” period to future work.

In addition to the single run results as shown above, we further performed 20 runs of each link. Figure 8 shows the mean and 95% confidence level (i.e. ± 2 standard deviations) results. From the results, TCP Probe, in average, estimates 85% of the bottleneck capacity on the Cable and China link, and 70% on the Alabama link. For the longer and higher loading paths (e.g. China and Alabama links), TCP Probe also shows larger variance in its estimates. This is because, in such links, the possibility of getting one “good sample”, which has no queuing delay on both packets in the sampling pair, is lower. Such problem can be resolved as long as the number of samples increases (i.e. run TCP longer) and the cross traffic is not persistent at very high rates.

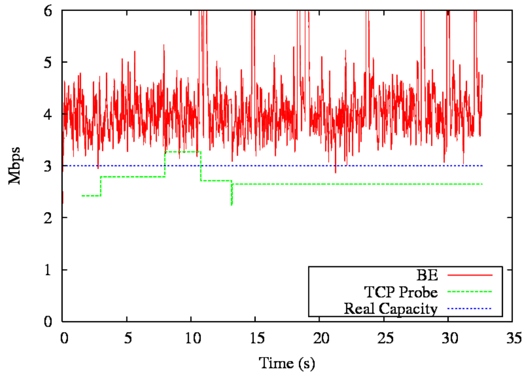

Figure 5: Measurement results of Cable link (*C=3Mbps*)

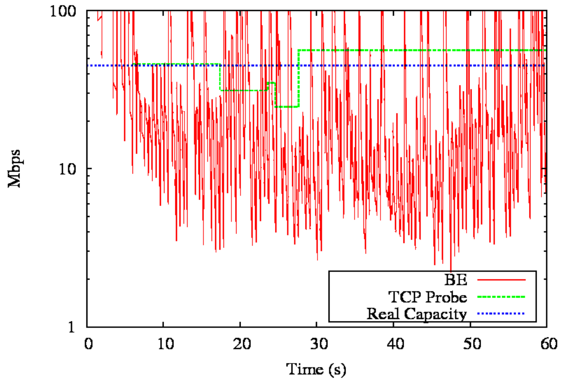

Figure 6: Measurement results of Tianjin University link (*C=45Mbps*)

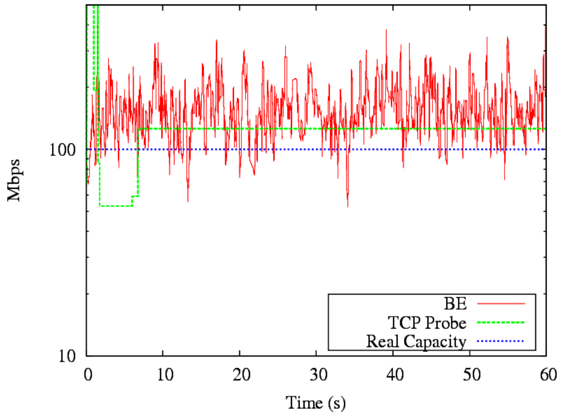

Figure 7: Measurement results of University of Alabama link (*C=100Mbps*)

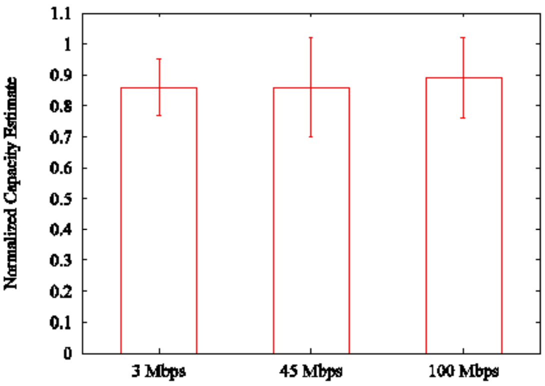

Figure 8: Mean and 95% confidence level of TCP Probe results in Internet measurements. (20 runs)

4 Application

In this section, we present an application of TCP Probe in the vertical handoff scenario. A vertical handoff involves two different network interfaces [9], which usually represent different technologies and thus result in a drastic change in link capacity. For instance, a vertical handoff from 1xRTT to 802.11b will result in a capacity change from 150Kbps to around 5Mbps[5]. By iteratively estimating link capacity using probing samples, TCP Probe is able to provide passive and "online" monitoring of link capacity. When operating in the emerging mobile and wireless network scenarios, where the access link capacity is varying due to vertical handoffs or dynamic wireless channel allocation, TCP Probe is capable of detecting the capacity changes correctly and timely. A "fast rate adaptation" technique can be carried out while a link capacity change from LOW to HIGH is detected. The "fast rate adaptation" technique will force TCP to enter *Slow Start* to exponentially increase its congestion window size to probe the network, instead of linearly probing in Congestion Avoidance phase.

The "fast rate adaptation" and TCP Probe algorithm is implemented in NS2 simulator, and a simple scenario, as shown in Figure 9, is used to evaluate the performance of TCP Probe. In the simulation, TCP Probe runs from node 1 to node 6, and four 5Mbps Pareto flows are running from node 7 to 10, 8 to 9, 11 to 14, and 12 to 13. A capacity change occurs on the link between node 1 and 2, from 10Mbps to 100Mbps (assuming this is a symmetric link), at around 80th second. The congestion window size trace and capacity estimation result is displayed in Figure 10.

From the simulation results, it is evident that TCP Probe is able to accurately and timely monitor the link capacity. Moreover, while the "fast rate adaptation" technique is employed, TCP Probe is able to better take advantages of the capacity increase than the original TCP.

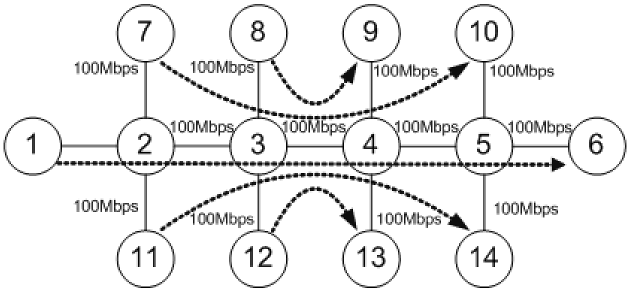

Figure 9: Simulation scenario of the vertical handoff scenario. The vertical handoff is occurred on the link between node 1 and 2, with the capacity change from 10Mbps to 100Mbps.

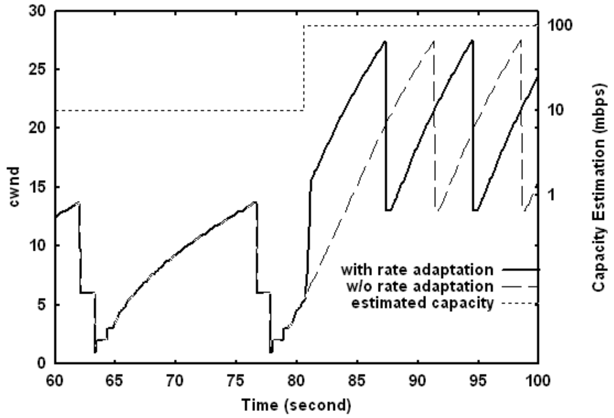

Figure 10: Simulation results of TCP Probe (Newreno) w/wo fast rate adaptation.

5 Conclusion

In this paper, we proposed TCP Probe as a passive capacity estimation extension to TCP. TCP Probe is an extension of CapProbe that is passive and does not require any additional probing traffic. , TCP Probe is applicable to all TCP variants. Using simulation and Internet measurements, we showed TCP Probe is able to accurately measure the bottleneck link capacity. The future work is to evaluate the speed and accuracy of TCP Probe with more diverse network configurations. Moreover, another work is now underway to utilize the passive capacity estimation in providing better peer selection and overlay structuring for P2P and overlay networks.

---

[5] 5Mbps is the "effective capacity" of 11Mbps transmission mode in 802.11b. The effective capacity stands for the maximum achievable data throughput excluding RTS/CTS and MAC/Physical layer O/H.

6 Acknowledgment

We are grateful to the following people for their help in carrying out TCP Probe measurements: Xiaoyan Hong (The University of Alabama) and Yantai Shu (Tianjin University, China).

7 References

[1] R. Braden, "Requirements for internet hosts communication layers," *IETF RFC 1122*, Oct. 1989.

[2] L.-J. Chen, T. Sun, D. Xu, M. Y. Sanadidi, and M. Gerla, "Access link capacity monitoring with TFRC probe," in *The 2nd Workshop on End-to-End Monitoring Techniques and Services*, 2004.

[3] L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla, "End-to-end asymmetric link capacity estimation," *IFIP Networking 2005, Waterloo, Ontario, Canada, 2005*.

[4] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?" in *IEEE Infocom'01*, 2001.

[5] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *ACM SIGCOMM*, 2000.

[6] M. Gerla, B. K. F. Ng, M. Y. Sanadidi, M. Valla, and R. Wang, "TCP westwood with adaptive bandwidth estimation to improve efficiency/friendliness tradeoffs," *Computer Communications*, vol. 27, no. 1, pp. 41–58, 2004.

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

[8] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi, "CapProbe: a simple and accurate capacity estimation technique," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 67–78, 2004.

[9] Mark Stemm, and Randy H. Katz. "*Vertical Handoffs in Wireless Overlay Networks*," ACM MONET, 1998.