End-to-end asymmetric link capacity estimation*

Ling-Jyh Chen, Tony Sun, Guang Yang, M. Y. Sanadidi, Mario Gerla

Computer Science Department, UCLA, Los Angeles, CA 90095, USA
{cclljj,tonysun,yangg,medy,gerla}@cs.ucla.edu

Abstract. Knowledge of link capacity is important for network design, management, and utilization. With the increasing popularity of asymmetric link technologies (such as DSL, 1xRTT, and satellite links), it is desirable to have a capacity estimation technique, which can simultaneously measure forward and backward direction link capacities on an Internet path. Moreover, this estimation must often be "sender only", because of receiver limitations or lack of standards. In this study, we propose a simple, fast and accurate technique, called AsymProbe, to estimate asymmetric link capacities. AsymProbe is a "sender only", round trip procedure. It achieves asymmetric link capacity estimation by strategically altering the ratio of probe and acknowledgement packet sizes. Using simulation and testbed experiments, we validate AsymProbe with a variety of network configurations. The results show that AsymProbe can correctly estimate the asymmetric link capacities as long as an appropriate packet size ratio can be employed.

1 Introduction

Knowledge of link capacity is particularly important for network design, management and utilization. A simple and accurate scheme for capacity measurement and monitoring is becoming increasingly desirable, especially for emerging technologies and applications such as overlay, peer-to-peer (P2P), sensor, grid and mobile networks. A successful capacity estimation solution will need to encompass speed of execution, simplicity, accuracy, and extendibility beyond the limits of traditional networks, in particular the increasingly popular *asymmetric* access methods to the Internet, e.g. DSL, cable modem and satellite links. It is also often imperative to carry out the estimation in a round trip, "sender only" fashion. This is because the receiver is not powerful enough to implement the estimation algorithm. It must, however, participate in the response to probe packets.

Several capacity estimation methods exist, including CapProbe [4], which is sender only and fast. However, sender only methods so far have addressed symmetric path extimations, ie, the minimum capacity is the same in both directions. Yet, asymmetric links do exist; moreover, many applications are intrinsically asymmetric too and thus can benefit from the knowledge of such asymmetries.

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

For example, in multimedia streaming and file downloading, bulk data is transmitted only in the forward direction, consuming much more bandwidth than the control traffic in the reverse direction. In this case, the knowledge of one-way capacity on the *forward* direction link is mandatory, as it is a much better predictor of the streaming or downloading rate, than the blindly measured round-trip bottleneck capacity, accounting for the cases when the forward link has larger capacity than the backward link.

Previous approaches on capacity estimation can be divided into two categories: one-way probing (e.g. Pathrate [1]) and round-trip probing (e.g. CapProbe [4]). In [4], a thorough comparison of modern capacity estimation methods was presented, where CapProbe was especially singled out as a fast and accurate capacity estimation mechanism addressing both wired and wireless links. However, limited by its round-trip nature, CapProbe only works well on symmetric links. When operating on an asymmetric link, CapProbe measures the narrower capacity of the two directions. It cannot distinguish the respective capacities of the forward and backward links.

Even though capacity estimation for asymmetric links can be achieved by conducting single direction capacity probing (e.g. Pathrate) for the two directions separately, this estimation strategy is often considered undesirable, as it imposes unnecessary computation overhead and complexity on the receiver (eg, the mobile host). Moreover, it requires compatible software and consistent computation methods in both hosts. To simplify the process of estimating asymmetric link capacities, round trip capacity probing is still the most desirable solution. Still, existing method like CapProbe, lacked such a capacility, modifications are needed to add support for accurate capacity estimations of asymmetric links.

To this end, in this study, we propose and evaluate a round trip technique for estimating asymmetric link capacities called AsymProbe. AsymProbe is engineered based on the well proven CapProbe mechanism. Through careful selection of probe and acknowledgement packet sizes, AsymProbe can successfully provide simple, fast, and accurate capacity estimates for asymmetric links.

The rest of the paper is organized as follows. In section 2, we survey and summarize work related to this study. An in-depth description of AsymProbe follows in Section 3. In section 4, we evaluate the accuracy of AsymProbe in estimating link capacity through series of NS2 simulations. In section 5, we present results from our testbed experiments to validate the capability of AsymProbe. Section 6 concludes the paper.

2 Background and Related Work

Previous research on capacity estimation relied on either delay variations among probe packets as illustrated in pathchar [3], or dispersion among probe packets as described in Nettimer [6] and Pathrate [1]. The analysis in [1] clearly revealed that the dispersions distribution can be multi-modal without multi-channels, and that the strongest mode in the multimodal distribution of the dispersion may correspond to either (1) the capacity of the path, or (2) a “compressed”

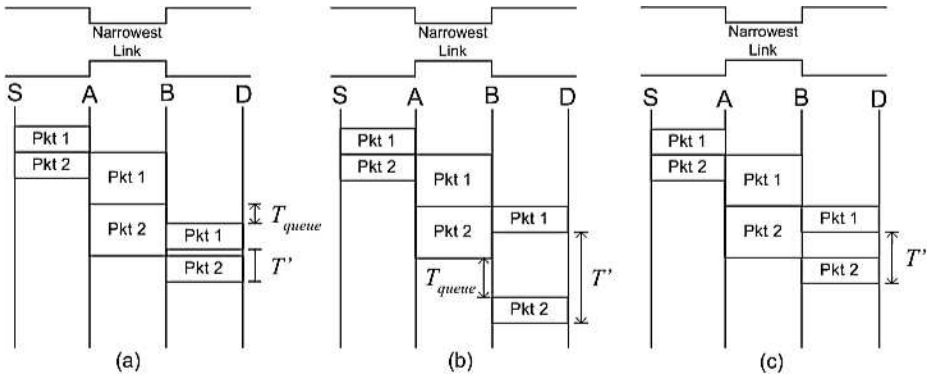

Fig. 1. (a) under-estimation caused by "expansion" (b) over-estimation caused by "compression" (c) the ideal case. (T': Measured dispersion; T_{queue}: Queueing delay)

dispersion, resulting in capacity over-estimation, or (3) to the Average Dispersion Rate (ADR), which is lower than the capacity.

Other tools such as pchar and clink [2] use variations of the same idea as pathchar. Pchar employs regression techniques to determine the slope of the minimum RTT versus the probing packet size. However, pathchar-like tools have limitations with respect to the speed of estimation process as shown in [4].

CapProbe [4] is a recently proposed capacity estimation technique shown to be both fast and accurate over a large range of scenarios. When a back-to-back packet pair is launched into a network, it is always dispersed at the bottleneck link according to the bottleneck capacity. If such dispersion is preserved until the pair arrives to destination, it identifies the bottleneck capacity (as shown in Fig. 1-c). Unfortunately, the dispersion can be either expanded or compressed, where "expansion" of dispersion leads to under-estimation and "compression" of dispersion leads to over-estimation of the capacity, as shown in Fig. 1-a,b.

To overcome this problem, CapProbe combines the use of dispersion and end-to-end delay measurements thus filtering out packet pair samples distorted by cross traffic. Whenever an incorrect value of capacity is estimated, either the first or the second packet, or both, have been delayed by cross traffic. In this case, the sum of the delays of the two packets in the packet pair, called the delay sum, includes some queuing delay. A delay sum that does not include any queuing delay introduced by cross traffic is referred to as the minimum delay sum. The dispersion of such a packet pair sample is not distorted by cross traffic and reflects the actual capacity. A valid sample can easily be identified since its delay sum is the minimum among delay sums of all packet pair samples. The capacity is then estimated by the equation:

$$C = \frac{P}{T} \tag{1}$$

where P is the sampling packet size, and T is the dispersion of the sample packet pair with the minimum delay sum.

The majority of the existing capacity estimation tools, including the ones discussed above, are inherently round-trip based. They estimate the narrowest capacity on the round-trip path. These techniques encounter severe constraints when measuring link capacities of increasingly popular asymmetric links, such as DSL, cable modem and satellite links where the forward link capacity is very different from the backward link capacity. In this study, we propose AsymProbe, a novel scheme that measures asymmetric link capacities in the round trip fashion. Details of this proposed approach will be presented in the following sections; evaluation of AsymProbe will be discussed in the simulation and experiments sections.

3 Proposed Approach: AsymProbe

In this section, we present AsymProbe, a novel capacity measuring technique that allows to measures the capacity of either the forward or backward narrow link on the path. The basic idea of AsymProbe stems from the observation that the measured dispersion in the original CapProbe can be introduced either in the forward or backward direction of an asymmetric link. When probing and acknowledgement packets are of same size, the measured dispersion is good for estimating the round-trip bottleneck capacity, since the narrowest link along the round-trip path gives the largest dispersion to the (probing or acknowledgement) packet pairs. One can then easily estimate this capacity by applying Eq. 1.

Fig. 2 depicts the packet pair interactions in an asymmetric link scenario, with link capacity C_1 on the forward direction link and capacity C_2 on the backward direction link. The probe packets are sent back-to-back with packet size P_1 on the forward direction link (from A to B); the acknowledgement packets are sent immediately upon receipt of probe packets with packet size P_2 on the backward direction link (from B to A). Suppose T_1 and T_2 represent the respective dispersions of probe packets and acknowledgement packets when they are sent back-to-back on the link; from the definition of Eq. 1, T_1 and T_2 can then be derived as $T_1 = \frac{P_1}{C_1}$ and $T_2 = \frac{P_2}{C_2}$.

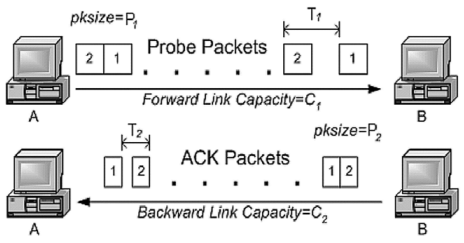

Fig. 2. Interaction of probe packets in asymmetric link scenarios

Table 1. Estimate asymmetric link capacity by varying packet sizes (ideal case without cross traffic and any queuing delays)

Probe (P_1) and ACK (P_2) Packet Size	Measured Dispersion T'	Capacity Estimation C'_1 and C'_2
$\frac{P_2}{C_2} > \frac{P_1}{C_1}$	$T' \rightarrow T_2$	$C'_1 < C_1; C'_2 = C_2$
$\frac{P_2}{C_2} < \frac{P_1}{C_1}$	$T' \rightarrow T_1$	$C'_1 = C_1; C'_2 < C_2$
$\frac{P_2}{C_2} = \frac{P_1}{C_1}$	$T' \rightarrow T_1 = T_2$	$C'_1 = C_1; C'_2 = C_2$
$P_2 = P_1$	$T' \rightarrow max(T_1, T_2)$	$C'_1 = C'_2 = min(C_1, C_2)$

The dispersion measured at the end host A, denoted as T', is the dispersion between back-to-back acknowledgement packets. Suppose $T_1 > T_2$, this means that measured dispersion T' equates to T_1. We assume that host B immediately acknowledges the probe packets without incurring additional queuing delay, else the min sum condition would be violated and the pair discarded. On the other hand, suppose $T_1 < T_2$, then T' reflects T_2 instead, i.e. the dispersion generated on the backward direction link prevails. Therefore,

$$T' = max(T_1, T_2) \tag{2}$$

By varying the packet size ratio between the probe and the ACK packets, and observing the forward link capacity estimate (C'_1, which is $\frac{P_1}{T'}$) and the backward link capacity estimation (C'_2, which is $\frac{P_2}{T'}$), the source node can obtain the correct capacity estimations for both directions of the link. For instance, suppose $C_1 > C_2$ and the initial packet size $P_1 = P_2$, it can be concluded that $T_2 > T_1$ because $\frac{P_2}{C_2} > \frac{P_1}{C_1}$. From the discussion above, the end-to-end dispersion T' measured equates to T_2. Therefore, $C'_2 = \frac{P_2}{T'} = C_2$ and $C'_1 = \frac{P_1}{T'} = C_2 < C_1$. The estimated capacity is the round-trip bottleneck link capacity on the asymmetric link (the minimum value of C_1 and C_2), which is exactly is what CapProbe estimates as presented in [4].

However, by increasing P_1 gradually, C'_2 remains equivalent to C_2, but C'_1 increases and approaches C_1 gradually. When P_1 increased to $\frac{P_2 \times C_1}{C_2}$, C'_1 converges to C_1 and C'_2 converges to C_2. Conversely, after P_1 increased to larger than $\frac{P_2 \times C_1}{C_2}$ (i.e. $T_1 > T_2$, since $\frac{P_1}{C_1} > \frac{P_2}{C_2}$), T' will reflect T_1. As a result, $C'_1 = \frac{P_1}{T'} = C_1$ and $C'_2 = \frac{P_2}{T'} < C_2$. This simple relationship between the estimated capacity (C'_1, C'_2) and the varying packet size can be harvested for the accurate estimation of asymmetric link capacities. Table 1 below details this relationship.

Based on the relationship presented in the table, the AsymProbe algorithm consists of four phases, of which the first three phases are *Probing* phases, and the last is the *Decision* phase. Two packet sizes are used in the probing phases: P_{max} and P_{min}, which are chosen carefully by taking network and system issues into account. In the first probing phase, P_1 and P_2 are both set to P_{max}. Thus we estimate the bottleneck capacity, C_{low}, of the round trip path. In phase 2 and 3, (P_1,P_2) are set first to (P_{max},P_{min}) and then to (P_{min},P_{max}) in order to

```
If (C[1][1]==C[2][1]){
    C1 = C[2][1];
    If (C[1][1] > C[3][1])  C2 = C[3][2];
    else C2 is larger than Max(C[2][2], C[3][2]);
} else If (C[1][1]==C[3][1]){
    C1 = C[3][1];
    If (C[1][1] > C[2][1])  C2 = C[2][2];
    else  C2 is larger than Max(C[2][2], C[3][2]);
} else If (C[1][2]==C[2][2]){
    C2 = C[2][2];
    If (C[1][2] > C[3][2])  C1 = C[3][1];
    else  C1 is larger than Max(C[2][1], C[3][1]);
} else If (C[1][2]==C[3][2]){
    C2 = C[3][2];
    If (C[1][2] > C[2][2])  C1 = C[2][1];
    else  C1 is larger than Max(C[2][1], C[3][1]);
}
```

Fig. 3. AsymProbe Algorithm (The *Decision* Phase)

estimate the forward and backward link capacities respectively. We use $C[i][1]$ and $C[i][2]$ to denote the estimation results of C_1' and C_2' in the i-th phase, respectively.

In the fourth phase, namely the *Decision* phase, a decision algorithm is performed to determine the estimation results of both direction links from all $C[i][1]$ and $C[i][2]$ as shown in Fig. 3. However, it should also be mentioned that the capability of AsymProbe in determining the larger capacity is mathematically bounded by the maximum ratio of packet sizes between probe and acknowledgement packets, i.e. the max of $\frac{P_1}{P_2}$ and $\frac{P_2}{P_1}$. Specifically, if the capacity estimates of all 3 phases are equal, and we know a priori that the link is asymmetric, then, the packet size ratio is not sufficient large to provide an accurate capacity estimation of the larger link. Therefore, AsymProbe is unable to estimate the actual capacity in the direction with higher speed, but will indicate that such condition has occurred and report a "lower-bound" (i.e. $\frac{P_{max}}{P_{min}} \times C_{low}$) of the capacity instead. In this case, one-way capacity estimation tools (e.g. one way version of CapProbe or Pathrate) can be applied to accurately measure the capacity in this direction - if this solution is feasible within the scope of the application. In section 5.4, we discuss another extension of AsymProbe to this problem.

4 Simulation

In this section, we present simulation results that evaluate the accuracy of capacity estimation of AsymProbe on paths with asymmetric links. AsymProbe is implemented in the NS-2 simulator [8]. Fig. 4 depicts the simulation topology that represents a commonly seen scenario nowadays with an asymmetric DSL link. All links are symmetric 100Mbps Ethernet links except the one between

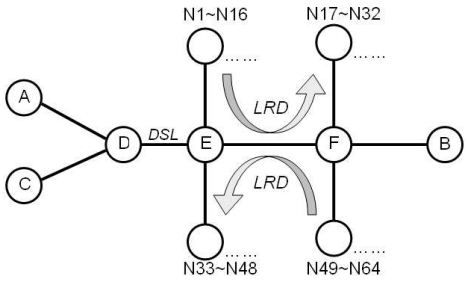

Fig. 4. Last-hop ADSL scenario. The link capacities are 100Mbps for all links, except the asymmetric DSL link between D and E ($D \rightarrow E : 128Kbps; E \rightarrow D : 1.5Mbps$)

Table 2. Simulation results of AsymProbe in last-hop DSL scenarios (Unit: Kbps)

Cross Traffic	AsymProbe from A		AsymProbe from B		CapProbe
	$A \rightarrow B$	$B \rightarrow A$	$B \rightarrow A$	$A \rightarrow B$	$A \Leftrightarrow B$
none	128	1500	1500	128	128
FTP ($B \rightarrow C$)	128	1500	1505	128.057	128
FTP ($C \rightarrow B$)	128.062	1500	1500	128	128
Poisson ($B \rightarrow C$, rate=300Kbps)	128	1500	1500	128	128
Poisson ($B \rightarrow C$, rate=750Kbps)	128	1500	1500	128	128
Poisson ($B \rightarrow C$, rate=1500Kbps)	128	1500	1500	128	128
Poisson ($C \rightarrow B$, rate=25.6Kbps)	128	1500	1500	128	128
Poisson ($C \rightarrow B$, rate=64Kbps)	128.006	1500	1500	127.936	128
Poisson ($C \rightarrow B$, rate=128Kbps)	127.988	1500	1483.143	128.039	128

node D and E, which is an asymmetric DSL link with 1.5Mbps downlink capacity (from E to D) and 128Kbps uplink capacity (from D to E). Nodes to the left of node D (namely A and C) belong to a home networks, while nodes to the right of node E are on the Internet.

The AsymProbe estimation is performed on the path between node A and B. In addition to the AsymProbe flow, various types of cross traffic were generated on the DSL link to test AsymProbe robustness. The cross traffic types used were FTP and Poisson based UDP traffic of different rates. For the Internet segment, long range dependent (LRD) traffic is created between node E and F in both directions. The LRD traffic is composed of 16 Pareto flows with alpha = 1.9 [7], and the overall rate of LRD traffic is 60Mbps in each direction.

The maximum and minimum AsymProbe packet sizes, P_{max} and P_{min}, are set to 1500 bytes and 100 bytes respectively. For the various cross traffic configurations described in Table. 2, AsymProbe is independently initiated from both A and B; results obtained from AsymProbe are then compared against CapProbe as summarized in Table 2.

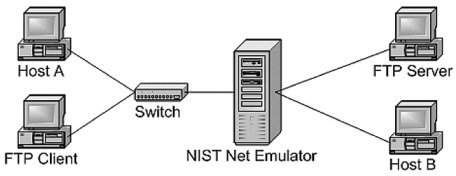

Fig. 5. Testbed for NIST Net experiments

From the results shown in Table 2, AsymProbe is able to estimate the correct link capacity in both directions for all test cases; whereas CapProbe can only estimate the bottleneck link capacity of the round-trip path. Moreover, simulation results also show that AsymProbe works when placed on either the end-client (node A) or the Internet server (node B). The results are consistent in both cases.

It is also worth mentioning that since the link capacity ratio of the simulated scenario is 1.5Mbps/128Kbps, it is smaller than the packet size ratio $\frac{P_{max}}{P_{min}}$, AsymProbe is thus able to measure the correct link capacities. However, if we decrease the packet size ratio (e.g. increasing P_{min} in order to avoid the fine time resolution problem as described in [5]) and obtain a packet size ratio that is larger than the link capacity ratio, AsymProbe will only estimate the correct capacity of the narrower link and output the other direction link as a lower bound estimation, defined as $\frac{P_{max}}{P_{min}} \times C_{low}$. In such case, one-way link capacity estimation tools can be launched.

5 Experiments

In this section, we present testbed and Internet experimental results to further evaluate AsymProbe. We first perform a set of experiments on a "controlled" testbed to calibrate and verify the correctness of the AsymProbe scheme and its Linux implementation. We then move to Internet measurements for an evaluation in the diverse and realistic scenario.

5.1 Testbed Experiments

The testbed experiments are performed in the configuration shown in Fig. 5. The NISTNet emulator [9] is used to set up the asymmetric bottleneck link of various capacities. A backlogged file transfer session is generated from the FTP server to the client as cross traffic. This FTP connection shares the bottleneck link with an AsymProbe connection that traverses from host A to B.

For reasons we will discuss shortly, we choose 1500 bytes and 500 bytes as the maximum and minimum packet sizes in this set of experiments, respectively. Thus we have $\frac{P_{max}}{P_{min}} = \frac{1500}{500} = 3$. We present the experiment results in Table 3 and Table 4, in which we may see that when the forward/backward (Table 3)

Table 3. NIST Net results on High/Low asymmetric links (Unit: Mbps)

Link Capacity		AsymProbe Estimation		CapProbe Estimation
F	B	F	B	
1	1	1.063	1.064	0.981
		1.064	1.065	0.981
		1.063	1.063	0.985
2	1	2.010	1.063	0.979
		2.015	1.062	0.979
		2.010	1.064	0.979
3	1	2.997	1.065	0.985
		3.012	1.065	0.983
		3.015	1.055	0.981
4	1	$\geq$3.611	1.064	0.979
		$\geq$3.609	1.061	0.981
		$\geq$3.611	1.062	0.979

F: Forward Link; B: Backward Link

Table 4. NIST Net results on Low/High asymmetric links (Unit: Mbps)

Link Capacity		AsymProbe Estimation		CapProbe Estimation
F	B	F	B	
1	1	1.063	1.065	0.981
		1.065	1.064	0.981
		1.065	1.065	0.979
1	2	1.064	2.015	0.979
		1.062	2.010	0.975
		1.006	1.989	0.981
1	3	1.062	2.997	0.983
		1.063	3.018	0.985
		1.056	2.997	0.981
1	4	1.059	$\geq$3.610	0.981
		1.065	$\geq$3.610	0.979
		1.065	$\geq$3.611	0.979

F: Forward Link; B: Backward Link

or backward/forward (Table 4) capacity ratio is below 3, AsymProbe measures both forward and backward capacities very accurately. When the ratio increases beyond 3, only a lower bound can be obtained in the direction with the larger capacity.

5.2 Internet Experiments

In addition to the controlled testbed experiments, we also perform a set of Internet measurements to evaluate AsymProbe in a more diverse and realistic scenario. In this set of experiments, again we have $\frac{P_{max}}{P_{min}} = \frac{1500}{500} = 3$. However, the asymmetric links we have found, provided by DSL [1] and Cable [2] companies, all have a higher down-link/up-link capacity ratio than 3. As presented in Table 5, AsymProbe captures the up-link capacities accurately, while only obtaining lower bounds for the down-links.

5.3 Discussion

From the simulation and experiment results above, AsymProbe is capable of estimating asymmetric link capacities, as long as the capacity ratio of the forward and backward links is within the range of the packet size ratio of the employed probe and acknowledgement packets. In order to increase the estimation range of AsymProbe, the packet size ratio should be as large as possible. However, this ratio is bound by implementation.

[1] DSL 1 is provided by Verizon: *http://www.verizon.com*; DSL 2 is provided by Hinet: *http://www.hinet.net*

[2] Cable Modem is provided by Comcast: *http://www.comcast.com*

Table 5. Internet results on asymmetric link

Link	Claimed Capacity		Estimated Capacity		
	Down	Up	#	Down	Up
DSL 1	1.5 Mbps	128 Kbps	1	$\geq$ 379 Kbps	132 Kbps
			2	$\geq$ 382 Kbps	132 Kbps
			3	$\geq$ 380 Kbps	132 Kbps
DSL 2	3 Mbps	512 Kbps	1	$\geq$ 1.49 Mbps	565 Kbps
			2	$\geq$ 1.53 Mbps	567 Kbps
			3	$\geq$ 1.51 Mbps	558 Kbps
Cable Modem	3 Mbps	256 Kbps	1	$\geq$ 721 Kbps	247 Kbps
			2	$\geq$ 730 Kbps	255 Kbps
			3	$\geq$ 723 Kbps	248 Kbps

Specifically, the maximum size of the employed packets must be bounded by the Maximum Transmission Unit (MTU), which is the largest size of an IP datagram allowed to transmit on the path without fragmentation. The size of MTU may vary greatly in different system configurations. However, practically it is set to 1500 bytes in most networks. Packets larger than MTU will be segmented into smaller fragments for transmission and then reassembled on the receiving host. Therefore, using packets larger than MTU is not appropriate for CapProbe-based capacity estimation techniques, since the dispersion measurement no longer reflects the bottleneck capacity.

On the other hand, the minimum size of AsymProbe packets is also bounded in accordance with the supported time resolution on the estimating host. This is due to the fact that a packet pair with a smaller packet size will result in a smaller inter-packet dispersion, which in turn requires a finer time resolution to be measured accurately. Assume the capacity of the narrow link is C and the probing packet size is P, the dispersion time (and also the clock granularity needed for accurate estimation) that needs to be measured is $T = P/C$. Table 6 shows the required clock granularities that are needed for different probing packet sizes and narrow link capacities.

Table 6. Required time resolution for accurate estimation

Packet Size	Narrow Link Capacity			
	1 Gbps	100 Mbps	10 Mbps	1 Mbps
100 bytes	0.0008 ms	0.008 ms	0.08 ms	0.8 ms
500 bytes	0.004 ms	0.04 ms	0.4 ms	4 ms
1000 bytes	0.008 ms	0.08 ms	0.8 ms	8 ms
1500 bytes	0.012 ms	0.12 ms	1.2 ms	12 ms

It is clear that the time resolution of an end host relies on the hardware speed and the operating system. A system with fast processors and I/O interfaces can provide a finer time resolution. Additionally, [5] also shows that the accuracy of CapProbe-based capacity estimation is tightly related to the runtime execution mode. With kernel mode implementations, the capacity estimation is faster and more accurate than with user mode implementations. Kernel mode implementations also provide better time resolutions. Therefore, kernel mode implementations can use smaller packets for capacity estimation than the user mode implementations.

In the presented testbed experiments, the employed packet sizes are bounded with 1500 bytes as the maximum and 500 bytes as the minimum. The value of 1500 bytes is determined by the MTU on the path, whereas the value of 500 bytes is the minimum packet size which can measure the dispersion accurately with the provided machine time resolution. Thus it is only capable of estimating an asymmetric link with capacity ratio up to 1500 : 500, namely 3 : 1. For those links with even higher "asymmetric ratios" (e.g. 1.5Mbps/128Kbps DSL links or 400Kbps/64Kbps satellite links), it is necessary to increase the packet size ratio by either increasing the maximum packet size or decreasing the minimum packet size. In such cases, using a faster machine or switching from user mode to kernel mode can help.

If all of the above procedures do not work, one can resort to one way capacity estimation as mentioned in Section 3. This, however, requires full implementation on the receiver. If the receiver does not cooperate, a possible solution is to use a packet "train" probing concept as suggested by other researchers [1]. The intent is to replicate the effect of a "long" probing packet without paying the penalty of reassembly at the host. To illustrate the technique, consider for example the situation of an asymmetric satellite link with 15Mbps downlink and 128Kpbs uplink capacity. The server in the Internet must determine the downlink speed to deliver the proper content to a mobile user. The downlink capacity estimation can be achieved by transmitting a train of 10 consecutive 1500 byte packets, with a Probe leader and Probe trailer. As before, the two Probe packets each trigger a 100 byte packet probe response from the receiver. The train dispersion in the forward link is preserved in the ACK dispersion measured by the sender after the round trip and provides the desired estimate. Basically, this scheme is an extension of AsymProbe, where the source experiments with trains of increasing length until success. As pointed out in [1], the longer the train, the less dominant the mode corresponding to the forward narrow capacity. Consequently, the less frequent the train samples where no delay/interference occurred along the path and thus the less accurate the measurements. The measurement however, provides a conservative (lower bound) estimate of the narrow forward capacity, which can be progressively improved as more and more samples are collected. By the way, the "average" capacity measurement (as opposed to min sum measurement) was shown to converge for large train length N to a value between the narrow link and the residual link bandwidth. As expected, the lower the utilization, the faster the min sum measurement convergence [1].

6 Conclusions

In this paper, we studied asymmetric link capacity estimation and proposed an extension of CapProbe, namely AsymProbe, to estimate asymmetric link capacities. By strategically altering the ratio of probe and acknowledgement packet sizes, AsymProbe can simultaneously measure the link capacities of both forward and backward direction links. Through simulation and testbed experiments, we validated the accuracy and capabilities of our proposed approach.

The unique advantage of AsymProbe is the ability to measure capacities from the server using a round trip method that does not require the cooperation of the receiver (which may have limited processing power or may be altogether unaware of AsymProbe). Moreover, the technique is extremely fast, thus it is suitable for mobile receivers that experience rapidly varying, often asymmetric Internet connectivity. The simplicity, accuracy, and speed of AsymProbe make it ideal in real deployments where online and timely capacity estimation is required. Better service can be provided by estimating both forward/backward direction path capacities. Typical applications feature the efficient transfer of multimedia files over rapidly varying Internet paths (which may include wireless segments). Popular examples are P2P streaming and file sharing, overlay network structuring, and intelligent vertical handoff decision.

References

1. C. Dovrolis, P. Ramanathan, and D. Moore. What do packet dispersion techniques measure? In Proc. of IEEE Infocom 2001.
2. A. B. Downey. Using Pathchar to Estimate Internet Link Characteristics. In Proc. of ACM SIGCOMM 1999.
3. V. Jacobson. Pathchar: A tool to infer characteristics of Internet paths. ftp://ftp.ee.lbl.gov/pathchar
4. R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, M. Y. Sanadidi. CapProbe: A Simple and Accurate Capacity Estimation Technique. In Proc. of ACM SIGCOMM 2004.
5. R. Kapoor, L.-J. Chen, M. Y. Sanadidi, M. Gerla. Accuracy of Link Capacity Estimates using Passive and Active Approaches with CapProbe. In Proc. of ISCC 2004.
6. K. Lai and M. Baker. Measuring Bandwidth. In Proc. of IEEE INFOCOM 1999.
7. M.S. Taqqu, W. Willinger, R. Sherman. Proof of a fundamental result in self-similar traffic modeling. SIGCOMM Computer Communications Review, 27: 5-23, 1997.
8. Network Simulator (NS-2). www.mash.cs.berkeley.edu/ns/
9. NIST Net. http://snad.ncsl.nist.gov/itg/nistnet/