TRIO: Measuring Asymmetric Capacity with Three Minimum Round-trip Times

Edmond W. W. Chan[†‡], Ang Chen[§], Xiapu Luo[§], Ricky K. P. Mok[§], Weichao Li[§], and Rocky K. C. Chang[§]
Corporate Research Department[†]
Huawei Research, China
edmond.chan@huawei.com
Department of Computing[§]
The Hong Kong Polytechnic University
{csachen|csxluo|cskpmok|csweicli|csrchang}@comp.polyu.edu.hk

ABSTRACT

Measuring network path capacity is an important capability to many Internet applications. But despite over ten years of effort, the capacity measurement problem is far from being completely solved. This paper addresses the problem of measuring network paths of asymmetric capacity without requiring the remote node's control or overwhelming the bottleneck link. We first show through analysis and measurement that the current packet-dispersion methods, due to the packet size limitations, can only measure up to a certain degree of capacity asymmetry. Second, we propose TRIO that removes the limitation by using round-trip times (RTTs). TRIO cleverly exploits two types of probes to obtain three minimum RTTs to compute both forward and reverse capacities, and another minimum RTT for measurement validation. We validate TRIO's accuracy and versatility on a testbed and the Internet, and develop a system to measure path capacity from the server or user side.

Categories and Subject Descriptors: C.2.3 [Computer-Communication Networks]: Network Operations; C.4 [Performance of Systems]: Measurement Techniques

General Terms: Measurement, Performance

Keywords: Network capacity, Bottleneck bandwidth, Non-cooperative, Packet-pair dispersion, Packet delay

1. INTRODUCTION

Many network applications can benefit from the knowledge of *path capacity*—the transmission rate of the slowest link in an end-to-end network path. For example, network capacity is often one of the metrics required in the diagnostic services offered by companies, such as SamKnows and OOKLA. Due to the proliferations of ADSL, DOCSIS cable networks, VSAT, and others, the path capacity is generally asymmetric with the downstream data rate (C_{dn}) different from the upstream's (C_{up}). For the xDSL technologies, their C_{up}/C_{dn} are determined by many factors, such as the wire quality, transmission distance, and different broadband offerings (e.g., C_{up}/C_{dn} of 1/18 [5] and 0.5/20 [8] in Mbits/s), and the data rates span a wide range [33].

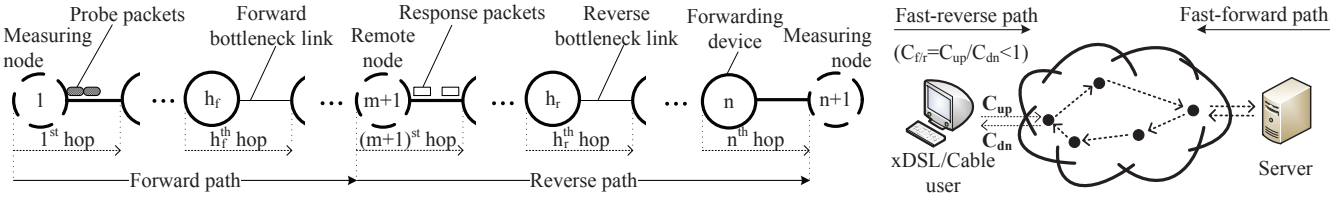
Measuring asymmetric capacity is a challenging problem. To measure both *forward capacity* (from a measuring node to the path's remote node) and *reverse capacity*, a possible solution is to perform two one-way measurements on the forward and reverse directions. However, the one-way measurement tools [10, 16] usually require controlling both nodes of a path, thus making this approach impractical for measurement with arbitrary remote nodes. On the other hand, only few tools—DSLprobe [14] and SProbe [32] based on packet-dispersion methods, and the flooding-based method [15]—can be used for measuring asymmetric capacity without the remote node's control, but their utility is limited by the restrictions on packet size.

All existing tools for measuring asymmetric capacity (AsymProbe [13], DSLprobe, SProbe, and the flooding-based method) generally require setting probe packets much larger (smaller) than response packets to measure the forward (reverse) capacity. Such requirement introduces two serious limitations. First, they cannot measure all degrees of capacity asymmetry, because the packet size is upper bounded by the path MTU. Second, they generally cannot support all measurement scenarios, because they may not be able to elicit response packets of the required size from the remote node. For example, DSLprobe elicits only small TCP RSTs (but not large response packets) from remote residential broadband users. Moreover, compared with the packet-dispersion method, the flooding-based method performs the measurement by sending high-rate packet trains to saturate the bottleneck link, and the packet rate limits the maximum capacity it can measure.

In this paper, we make three main contributions:

[‡]This work was completed when the author was with The Hong Kong Polytechnic University.

ACM CoNEXT 2011, December 6–9 2011, Tokyo, Japan.



(a) An n-hop round-trip path. (b) An xDSL/cable network scenario.

Figure 1: The asymmetric capacity measurement model used in the paper.

1. Probe generalization We generalized the packet-dispersion methods used by AsymProbe, DSLprobe, and SProbe into *round-trip probes* (RTPs) and *two-way probes* (TWPs), and conducted a detailed analysis for the properties and limitations of the two generalized probes under different degrees of packet-size asymmetry. Our analysis shows that the RTP's dispersions underestimate the path capacities in case of insufficient degrees of packet-size asymmetry, and the TWP's can measure only the reverse capacity. To the best of our knowledge, this paper is the first to generalize and analyze the current packet-dispersion methods.

2. Method for measuring asymmetric capacity We designed TRIO for measuring asymmetric capacity without requiring the remote node's control or overwhelming the bottleneck link. TRIO exploits *round-trip times* (RTTs) of RTPs and TWPs to estimate packet dispersions indirectly. By doing so, TRIO still computes forward and reverse capacities based on packet dispersions but can measure the capacities for any degree of asymmetry without using asymmetric packet sizes. To reduce measurement traffic, TRIO cleverly exploits RTPs and TWPs to obtain three minimum RTTs (minRTTs) for estimating both forward and reverse capacities at the same time. A fourth minRTT can be additionally gleaned from RTPs to measure both faster-path and slower-path capacities for measurement validation.

3. Implementation and evaluation of TRIO We prototyped TRIO in a measuring system that can measure asymmetric capacities from the (web) server side or (residential) user side without installing additional software at the remote node. Our testbed and Internet experiments evaluated TRIO over a wide range of network environments and compared its accuracy and versatility with AsymProbe, DSLprobe, and SProbe.

In §2, we first present a measurement model used in this paper. In §3, we analyze the properties and limitations of using the packet-dispersion method to measure asymmetric capacity. In §4, we introduce TRIO and its implementation. We evaluate TRIO's accuracy and compare TRIO with the existing tools based on testbed and Internet experiments in §5. We discuss limitations and possible extensions of TRIO in §6. After highlighting previous works in §7, we conclude this paper in §8.

2. CAPACITY MEASUREMENT MODEL

2.1 Capacity metrics

This paper considers an n-hop round-trip path between a measuring node and a remote node. Figure 1(a) depicts the path by laying it out straightly; therefore, the measuring node is replicated at the right end, and the remote node is located in the "middle" of the path. The forward path consists of the first m hops, and the reverse path the remaining $n-m$ hops, where $1 \le m < n$. The h^{th} $(h = 1, \ldots, n)$ hop comprises the node and its outgoing link with a transmission rate of $C^{(h)}$ in bits/s. The figure also shows a forward bottleneck hop (i.e., the h_f^{th} hop) with capacity C_f and a reverse bottleneck hop (i.e., the h_r^{th} hop) with capacity C_r. Def. 1 defines four path capacity metrics.

DEFINITION 1. *(End-to-end capacity metrics for an n-hop round-trip path)*

1. *Forward capacity:* $C_f \equiv C^{(h_f)} = \min_{1 \le h \le m} C^{(h)}$,
2. *Reverse capacity:* $C_r \equiv C^{(h_r)} = \min_{m+1 \le h \le n} C^{(h)}$,
3. *Faster-path capacity:* $C_B = \max\{C_f, C_r\}$,
4. *Slower-path capacity:* $C_b = \min\{C_f, C_r\}$.

The slower-path capacity is often referred to as *round-trip capacity*. When $C_f \neq C_r$, the round-trip path is a *capacity-asymmetric path*.

This paper considers the problem of measuring both C_f and C_r of a capacity-asymmetric path by the measuring node. The capacity-asymmetric path can be further classified into *fast-reverse* (FR) path if $C_{f/r} < 1$ and *fast-forward* (FF) path if $C_{f/r} > 1$, where $C_{f/r} = C_f/C_r$. The degree of capacity asymmetry obviously decreases with $C_{f/r}$ for an FR path but increases with $C_{f/r}$ for an FF path. For example, we consider the scenario in Figure 1(b) in which the xDSL/cable with $C_{up}/C_{dn} < 1$ is the bottleneck for both forward and reverse paths. Therefore, if the measurement is performed on the xDSL/cable side, its path is an FR path (i.e., $C_{f/r} = C_{up}/C_{dn}$). However, if the measurement is performed on the server side, its path is an FF path.

2.2 Measurement model

The measuring node measures the path capacity by injecting a sequence of probes, each of which comprises

a group of one or more probe packets, to the remote node. Each probe in turn elicits one or more response packets from the remote node. As will be seen next, the capacity is measured based on the response packets' dispersion or RTTs. Besides the measurement traffic, the path also admits cross traffic that enters and exits from arbitrary hops of the path. In our analysis in §3 and §4, unless stated otherwise, we adopt the following assumptions, which are commonly found in previous works (e.g., [24, 18, 12]).

1. Both the forward and reverse paths are static and unique during the measurement.
2. Each node is a store-and-forward device using a FCFS queueing discipline.
3. All the probe and response packets are received successfully. Combining with (1)-(2) also implies that the probe (or response) packets arrive at the remote (or measuring) node in the original order.
4. The processing delay is negligible when compared with the transmission delay and propagation delay at each hop.
5. The probes are sufficiently spaced out that the first packet in a probe is never delayed by the preceding probe, and similarly for the response packets.

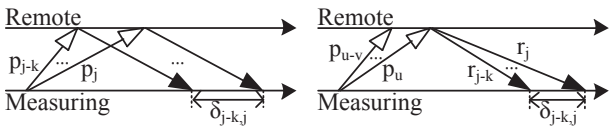
3. GENERALIZED PROBES

To motivate the design of TRIO, in this section we analyze the properties and limitations of the current packet-dispersion methods used by SProbe [32], AsymProbe [13], and DSLprobe [14] for asymmetric capacity measurement. It is important to note that they (and TRIO) are special instances of two generalized probing methods: *round-trip probes* (k-RTP) and *two-way probes*[1] ((v,k)-TWP), where k and v are configurable parameters. Table 1 shows that all three methods use k-RTP for forward-path measurement. For the reverse-path measurement, SProbe uses (0,1)-TWP, and the other two use k-RTP.

3.1 k-round-trip probe

3.1.1 Packet dispersion and capacity estimates

Figure 2(a) illustrates the k-RTP, where $k \geq 0$. To measure the capacity, the measuring node dispatches $k+1$ back-to-back probe packets $\{p_{j-k}, \ldots, p_j\}$, each of which elicits a single response packet from the remote node. To reduce the notations, we also use p_j to refer to the response packet elicited by p_j. The capacity is computed based on the response packets' dispersion, denoted by $\delta_{j-k,j}$, which is the time elapsed between receiving the first response packet and the last. For $k = 1$, the probe is usually known as a packet pair, and the dispersion as *packet-pair dispersion* (PPD). For $k > 1$, the probe is a packet train, and the dispersion is known as *packet-train dispersion* (PTD). We also let S_f and S_r be the respective sizes of the probe and response packets in bits, and $S_{f/r} = S_f/S_r$ be the degree of packet-size asymmetry.



(a) k-RTP. (b) (v,k)-TWP.

Figure 2: Two generalized probing methods for measuring asymmetric capacity.

Lemma 1 gives the packet dispersion obtained by the k-RTP method when the probe and response packets are not interfered by cross traffic. Note that the dispersion is either the dispersion generated by the h_f^{th} hop or that by the h_r^{th} hop. To measure the forward capacity, it is therefore necessary to set $S_{f/r} > 1$ in order to obtain the forward dispersion. Similarly, it is necessary to set $S_{f/r} < 1$ for measuring the reverse capacity. Furthermore, Prop. 1 uses the result in Lemma 1 to obtain the conditions for gaining the correct dispersions for forward and reverse capacity estimations.

LEMMA 1. *Consider that none of the probe and response packets in a k-RTP suffers from queueing delay induced by cross traffic throughout the path. The unbiased packet dispersion is given by*

$$\delta_{j-k,j} = \max_{h=1,\ldots,n}\{kX^{(h)}\} = k \times \max\{X^{(h_f)}, X^{(h_r)}\}, \quad (1)$$

where $X^{(h)} = S/C^{(h)}$ is the time of transmitting a packet of size S at the h^{th} hop. $X^{(h_f)}(= S_f/C_f)$ and $X^{(h_r)}(= S_r/C_r)$ are the forward PPD and reverse PPD, respectively.

PROOF. By extending the result in [13] for the PPD, we obtain Eqn. (1). □

PROPOSITION 1. *Given the same assumptions in Lemma 1,*

1. *If the packet sizes satisfy $S_{f/r} \geq C_{f/r}$, the forward capacity can be obtained by $C_f = (kS_f)/\delta_{j-k,j}$.*
2. *If the packet sizes satisfy $S_{f/r} \leq C_{f/r}$, the reverse capacity can be obtained by $C_r = (kS_r)/\delta_{j-k,j}$.*

PROOF. We obtain the results by substituting $X^{(h_f)} = S_f/C_f$ and $X^{(h_r)} = S_r/C_r$ in the right hand side of Eqn. (1). □

3.1.2 Measurement limitations

Unfortunately, practical path MTUs limit the degree of packet-size asymmetry to the extent that the conditions in Prop. 1 may not hold. Let S_{max} and S_{min} be the maximally and minimally permitted packet sizes.

[1] As will be clear shortly, "two-way" means that both the measuring and remote nodes dispatch back-to-back packets to the forward and reverse paths, respectively.

Table 1: The methods (1–3) and capabilities (4–6) of the existing tools and TRIO for measuring asymmetric capacity. The ✓* means that the tool works for only some measurement scenarios.

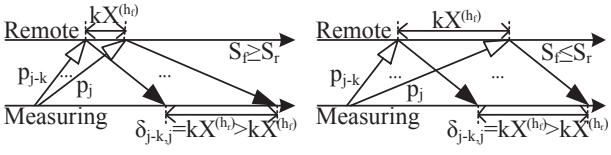
	AsymProbe [13]	DSLprobe [14]	SProbe [32]	TRIO
1. Probes for measuring C_f	1-RTP ($S_f > S_r$)	k-RTP ($S_f/S_r = S_{max}/S_{min}$)	1-RTP ($S_f/S_r = S_{max}/S_{min}$)	0-RTP & $(1,0)$-TWP
2. Probes for measuring C_r	1-RTP ($S_f < S_r$)	k-RTP ($S_f = S_r = S_{min}$)	$(0,1)$-TWP (S_r = MSS)	$(1,1)$-TWP
3. Measurement data	Packet dispersion	Packet dispersion	Packet dispersion	Packet delay
4. FF path (C_f,C_r)	(✓*,✓)	(✓*,✓)	(✓*,✓)	(✓,✓)
5. FR path (C_f,C_r)	(✓,✓*)	(×,×)	(✓,✓)	(✓,✓)
6. Non-cooperativeness	No	Yes	Yes	Yes

Thus, $S_{min}/S_{max} \leq S_{f/r} \leq S_{max}/S_{min}$. Furthermore, using $S_{f/r} \geq 1$ for C_f and $S_{f/r} \leq 1$ for C_r, we enumerate in Table 2 six possible scenarios of using k-RTP to measure the capacities of FF and FR paths. It is clear that (1)-(4) give the correct estimates, because they satisfy the conditions in Prop. 1, but (I) and (II) do not due to the insufficient degree of packet-size asymmetry.

Table 2: The six scenarios of using k-RTP to measure C_f and C_r for FF and FR paths.

	FF path ($C_{f/r} > 1$)	FR path ($C_{f/r} < 1$)
Measuring C_f ($S_{f/r} \geq 1$)	$S_{f/r} \geq C_{f/r} > 1$ (1) $C_{f/r} > S_{f/r} \geq 1$ (I)	$S_{f/r} \geq 1 > C_{f/r}$ (3)
Measuring C_r ($S_{f/r} \leq 1$)	$S_{f/r} \leq 1 < C_{f/r}$ (2)	$S_{f/r} \leq C_{f/r} < 1$ (4) $C_{f/r} < S_{f/r} \leq 1$ (II)

For both cases (I) and (II), the measuring node underestimates the faster-path capacity due to the packet dispersion resulted from the slower-path bottleneck. To see the reason, consider Figure 3(a) which shows a failure scenario for measuring forward capacity. Since $C_{f/r} > S_{f/r}$ (case (I)), the measuring node actually obtains the reverse (instead of forward) dispersion which will result in an underestimated forward capacity. Figure 3(b), on the other hand, shows a failure scenario for measuring reverse capacity, where $C_{f/r} < S_{f/r}$ (case (II)). The measuring node actually obtains the forward (instead of reverse) dispersion, and the incorrect dispersion, similar to case (I), will underestimate the reverse capacity.



(a) $C_{f/r} > S_{f/r} \geq 1$ (I). (b) $C_{f/r} < S_{f/r} \leq 1$ (II).

Figure 3: Examples for (I) and (II) in Table 2.

To be more concrete, we consider $S_{max}/S_{min} = 1518/64$ (the maximum and minimum Ethernet frame sizes in bytes) for capacity measurement. According to Prop. 1, we set $S_{f/r} = 1518/64 = 23.72$ for measuring forward capacity and $S_{f/r} = 64/1518 = 0.042$ for reverse capacity. However, some ADSL services (e.g., [8]) provide data rates of $C_{up}/C_{dn} = 0.5/20$. Therefore, the case (I) error is expected when measuring paths to a remote user using the ADSL service, because $S_{f/r} < C_{f/r} = C_{dn}/C_{up} = 40$. Similarly, the case (II) error will occur for reverse capacity measurement when the measuring node is the ADSL user, because $S_{f/r} > C_{f/r} = C_{up}/C_{dn} = 0.025$.

Using S_f or S_r larger than path MTU introduces serious problems to the k-RTP measurement, because the probe or response packets will be fragmented along the path. When the fragmentation occurs before a bottleneck, the additional fragments' IP headers will increase the packet dispersion obtained by the measuring node. Therefore, the measuring node should obtain the resultant packet size for the k-RTP measurement, but knowing the new S_f generally requires capturing all probe fragments from the remote node. The packet dispersion can also be biased by the post-bottleneck fragmentation, because the fragments may queue one another at a post-bottleneck link due to the size increase.

3.1.3 Testbed evaluation

We evaluated the impact of insufficient degree of packet-size asymmetry in a testbed depicted in Figure 4. It was configured with a 12-hop round-trip path ($n = 12$), consisting of a probe sender as the measuring node, a web server running Apache v2.2.3 as the remote node, three cross-traffic clients $\mathbb{X}_1 - \mathbb{X}_3$, and five forwarding devices: two Linux 2.6.26 routers $\mathbb{R}_1 - \mathbb{R}_2$ and three 100-Mbits/s Ethernet switches $\mathbb{S}_1 - \mathbb{S}_3$.

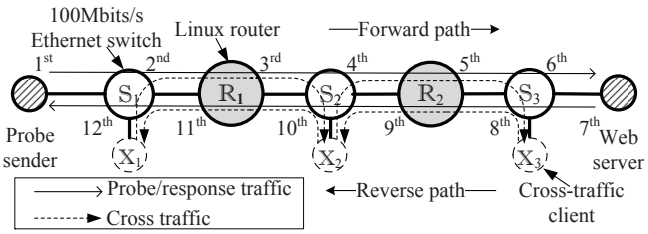


Figure 4: The testbed topology.

Each cross-traffic client generated forward (or reverse) cross traffic to the cross-traffic client on the right (or

left) to emulate a fixed loading rate ρ of 20% on the corresponding path segment. Similar to [16], the inter-arrival times of the packets follow the Pareto distribution with a shape parameter of $\alpha = 1.9$ (i.e., infinite variance), and the IP packet size is uniformly distributed over $[40, 1500]$ bytes. The probe sender dispatched a sequence of Poisson-modulated 1-RTPs with a mean probing rate of 2 Hz. It was equipped with a DAG card [3] to obtain RTTs in microsecond resolution (which is limited by `libpcap` [9]).

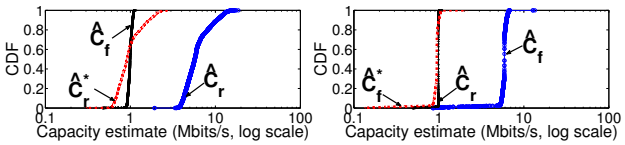
FR path We emulated an FR path by running Click v1.8 [1] at $\mathbb{R}_1$. As a result, $\mathbb{R}_1$'s link to $\mathbb{S}_2$ was the forward bottleneck link with $C_f = C^{(3)} = 1$ Mbit/s, and $\mathbb{R}_1$'s link to $\mathbb{S}_1$ the reverse bottleneck link with $C_r = C^{(11)} = 24$ Mbits/s. The Click router was also configured to set the RTT between the probe sender and web server to 300ms. We used IP packet sizes $S_{max}/S_{min} = 1500/240$ (in bytes) to obtain 3500 estimates for the forward and reverse capacities without using any cross-traffic filtering technique.

Figure 5(a) plots the CDFs of the capacity measurement results. It shows that the 1-RTP measurement for forward capacity (case (3)) is very accurate: 99% of the estimates (denoted by $\hat{C}_f$) fall within [0.87,1.10] Mbits/s. However, the 1-RTP measurement significantly underestimates the reverse capacity (case (II)) with 99% of the estimates (denoted by $\hat{C}_r$) falling within [3.63,14.18] Mbits/s. Moreover, we notice a large variation in $\hat{C}_r$, which could be reduced by applying a filtering technique (e.g., [20]). We have also included $\hat{C}_r^* = S_{min}/\delta_{j-1,j}$ which estimates reverse capacity using S_{min} (instead of S_{max}). Therefore, $\hat{C}_r^*$'s CDF is just a left shift of $\hat{C}_r$'s, and it actually measures C_f. As a result, $\hat{C}_f$'s and $\hat{C}_r^*$'s CDFs are close to each other.

FF path We ran another set of experiments by designating $\mathbb{R}_2$ to emulate an FF path. We first restored $\mathbb{R}_1$'s link capacities ($C^{(3)}$ and $C^{(11)}$) to 100 Mbits/s and ran Click at $\mathbb{R}_2$ to emulate $C_f = C^{(5)} = 24$ Mbits/s and $C_r = C^{(9)} = 1$ Mbit/s. Other parameters remained unchanged. Figure 5(b) shows that the 1-RTP measurement for reverse capacity (case (2)) is very accurate: 99% of $\hat{C}_r$ fall in [0.73,1.01] Mbits/s. However, the forward capacity (case (I)) is significantly underestimated with 99% of $\hat{C}_f$ in [1.18,6.75] Mbits/s. We have also included $\hat{C}_f^* = S_{min}/\delta_{j-1,j}$ which estimates forward capacity using S_{min}. Similar to the FR path, $\hat{C}_r$'s and $\hat{C}_f^*$'s CDFs are close to each other, because the dispersion is actually determined by the reverse bottleneck.

3.2 (v, k)-two-way probe

Figure 2(b) shows a (v, k)-TWP (where $v, k \geq 0$) which comprises a sequence of $v+1$ back-to-back probe packets $\{p_{u-v}, \ldots, p_u\}$. The probe packets are customized to induce from the remote node a sequence



(a) $C_{f/r} = 1/24$ (FR path). (b) $C_{f/r} = 24/1$ (FF path).

Figure 5: CDFs of the capacity estimates obtained from 1-RTPs with $S_{max}/S_{min} = 1500/240$ (in bytes) for two capacity-asymmetric paths.

of $k+1$ back-to-back response packets $\{r_{j-k}, \ldots, r_j\}$ upon p_u's arrival. However, the remote node will ignore other preceding probe packets. Therefore, we can regard $\{r_{j-k}, \ldots, r_j\}$ as a "curtailed" k-RTP dispatched by the remote node to the measuring node. Assuming that the sequence of response packets do not experience queueing delay caused by cross traffic on the reverse path, it is straightforward to derive from Eqn. (1) the unbiased packet dispersion of $\{r_{j-k}, \ldots, r_j\}$ (where $k > 0$) observed by the measuring node:

$$\delta_{j-k,j} = \max_{h=m+1,\ldots,n} \{kX^{(h)}\} = kS_r/C_r. \qquad (2)$$

The measuring node can therefore estimate the reverse capacity by $kS_r/\delta_{j-k,j}$.

Although (v, k)-TWPs are only for measuring the reverse capacity, the TWP measurement has two attractive properties. First, the reverse capacity estimation is independent of the probe packets preceding p_u. As will be shown in the next section, this property is exploited by TRIO to use the same TWP to measure both forward and reverse capacities. Second, Eqn. (2) shows that the packet dispersion of $\{r_{j-k}, \ldots, r_j\}$ is independent of the forward dispersion. Thus, the reverse capacity estimation is immune from the case (II) error.

To evaluate the quality of the TWP measurement, we configured the testbed in Figure 4 with $C_{f/r} = 1/24$ (in Mbits/s) using $\mathbb{R}_1$ and measured reverse capacity using $(0, 1)$-TWPs and $(1, 1)$-TWPs with $S_f \in \{240, 1500\}$ bytes (to induce different forward PPDs) and $S_r = 1500$ bytes. Other configuration settings were unchanged. Similar to before, we did not apply any cross-traffic filtering technique to the analysis. As recalled, the previous 1-RTP measurement fails to obtain correct dispersion for this FR path due to the case (II) error.

Figure 6 shows that more than 60% of the TWPs obtain $\hat{C}_r$ with less than 2% of error, regardless of the number of probe packets and probe packet size. The TWP measurement also does not overestimate the reverse capacity, because the cross traffic did not exist after the reverse bottleneck that would otherwise compress the PPD [16]. The underestimation, on the other hand, was the result of expanding the PPD by the cross traffic present at or before the reverse bottleneck.

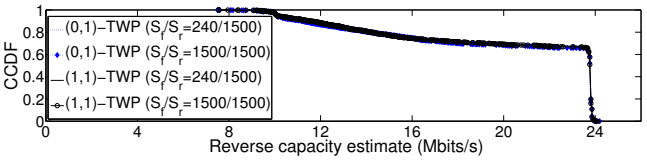


Figure 6: CCDF of the capacity estimates obtained by $(0,1)$-TWPs and $(1,1)$-TWPs with $S_f \in \{240, 1500\}$ bytes, $S_r = 1500$ bytes, and $\rho = 20\%$.

3.3 AsymProbe, DSLprobe, and SProbe

Table 1 summarizes the probing methods, packet sizes, and measurement data (under (1)-(3)) used by AsymProbe, DSLprobe, SProbe, and TRIO, and their measurement capabilities (under (4)-(6)). AsymProbe, DSLprobe, and SProbe all use the k-RTPs to measure forward capacity by setting $S_f > S_r$. Based on the packet dispersion, they estimate the capacity by $kS_f/\delta_{j-k,j}$. However, due to the case (I) error, none of them can measure the forward capacity of FF paths accurately for all measurement scenarios. Therefore, we highlight this by $\checkmark^*$ for C_f in row 4 of the table. AsymProbe and DSLprobe also use the k-RTPs with $S_f \leq S_r$ for measuring reverse capacity by $kS_r/\delta_{j-k,j}$. Therefore, the case (II) error will affect their results for FR paths, as indicated by $\checkmark^*$ for C_r in row 5 of the table. Finally, based on the choices of S_f and S_r, it is not difficult to see that DSLprobe is designed only for FF paths.

Table 1 also shows that SProbe is the only tool that exploits the (v,k)-TWP (with $v = 0$ and $k = 1$) to measure reverse capacity. SProbe dispatches an HTTP `GET` request to induce a pair of S_r-byte TCP data packets from a web server for the measurement, where S_r depends on the negotiated TCP maximum segment size (MSS). However, our empirical evaluation of SProbe (to be presented in §5) shows that its reverse-path measurement is often inaccurate, because the response packets are not dispatched consecutively.

4. TRIO

Table 1 shows that TRIO is designed for both FF and FR paths. Figure 7 illustrates a TRIO's probe which consists of a 1-RTP and a $(1,1)$-TWP. For the 1-RTP, two back-to-back probe packets (p_0^R and p_1^R) are dispatched, and each elicits a response packet. For the $(1,1)$-TWP, two back-to-back probe packets are also dispatched (p_0^T and p_1^T), but only p_1^T elicits two response packets (r_0^T and r_1^T). The only requirement on the packet size is that all probe packets share the same size and that all response packets share the same size. To ease the discussion, we simply assume that $S_f = S_r = S$. The results below can be easily adapted for $S_f \neq S_r$.

Unlike the existing methods, TRIO does not measure the packet dispersion directly. Instead, it measures three RTTs—d_{R0}, d_{T0}, and d_{T1}—from the RTP and TWP as shown in Figure 7. Specifically, it uses d_{R0} and d_{T0} to estimate forward capacity, and d_{T0} and d_{T1} reverse capacity. An important advantage of admitting RTT as the basic unit for capacity measurement is removing the case (I) and case (II) errors. As a result, TRIO can measure both FF and FR paths with any degree of capacity asymmetry. Another advantage, as will be shown shortly, is the ability of filtering RTT samples that are biased by cross traffic.

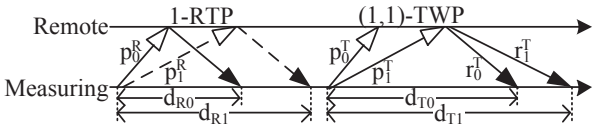


Figure 7: A TRIO's probe consisting of a 1-RTP and a $(1,1)$-TWP.

There are two important points worth noting about TRIO's measurement methods. First, as shown in Table 1, using 0-RTP and $(1,0)$-TWP in fact suffices for the sole purpose of measuring forward capacity, because they can provide d_{R0} and d_{T0}, respectively. However, TRIO uses $(1,1)$-TWP instead to measure reverse capacity at the same time. Second, TRIO uses 1-RTP (instead of 0-RTP in Table 1) to additionally obtain d_{R1} which, as we will see in §4.4, is used for measurement validation. Since d_{R1} is optional for the capacity measurement, we use dotted lines for the corresponding probe and response packets in Figure 7.

4.1 Preliminaries

This section provides preliminaries for the analysis in the next section. We consider a pair of back-to-back packets $\{p_{j-1}, p_j\}$ with identical packet size S dispatched from the measuring node to traverse an n-hop network path. Let d_j be the *packet delay* for p_j (which is for the second probe packet and its elicited response packet) to complete the n-hop traversal. Then,

$$d_j = \sum_{h=1}^{n} (X^{(h)} + T^{(h)} + w_j^{(h)}) = D^{(n)} + \sum_{h=1}^{n} w_j^{(h)}, \quad (3)$$

where $D^{(n)} = \sum_{h=1}^{n} (X^{(h)} + T^{(h)})$ is the constant forwarding delay comprising at each hop a constant transmission delay $X^{(h)}$ and a constant delay of $T^{(h)}$ for propagating the packet to the next hop. The component $w_j^{(h)}$ is the queueing delay introduced by the packets at the "head" of the queue upon p_j's arrival at the h^{th} hop. Lemma 2 gives the second packet's total queueing delay, and the proof is given in the Appendix.

LEMMA 2. *Consider that a pair of back-to-back packets $\{p_{j-1}, p_j\}$ are dispatched from the measuring node to traverse an n-hop path ($n \geq 1$), and they do not en-*

counter any cross-traffic-induced queueing delay on the path. Then, p_j's total queueing delay is given by

$$\sum_{h=1}^{n} w_j^{(h)} = \max_{h=1,\ldots,n}\{X^{(h)}\}. \quad (4)$$

4.2 Measuring asymmetric capacity

To measure the forward and reverse capacities, TRIO first obtains the minimums of d_{R0}, d_{R1}, d_{T0}, and d_{T1} from a sequence of 1-RTPs and $(1,1)$-TWPs. A probe packet's minRTT is the RTT experienced by the probe packet and the elicited response packet, neither of which encounters cross-traffic-induced queueing delay on the path [12]. However, the minRTT could still include the queueing delay induced by the preceding packets belonging to the same probe. By sending a sufficiently long sequence of probes, we assume (similarly as [17, 25]) that the minimum observable values of d_{R0}, d_{R1}, d_{T0}, and d_{T1} from the sequence converge to their corresponding minRTTs, which are given in Lemma 3.

LEMMA 3. *A sequence of adequately spaced 1-RTPs and $(1,1)$-TWPs with $S_f = S_r = S$ are dispatched on an n-hop round-trip path with C_f and C_r. Then, the minimums of d_{R0}, d_{R1}, d_{T0}, and d_{T1} are given by*

$$\min\{d_{R0}\} = D^{(n)}, \quad (5)$$
$$\min\{d_{R1}\} = D^{(n)} + \max\{S/C_f, S/C_r\}, \quad (6)$$
$$\min\{d_{T0}\} = D^{(n)} + S/C_f, \quad (7)$$
$$\min\{d_{T1}\} = D^{(n)} + S/C_f + S/C_r. \quad (8)$$

PROOF. According to Eqn. (3), the four minRTTs contain the same $D^{(n)}$ (due to fixed packet size and unique network path) but different amounts of queueing delay ($\sum_{h=1}^{n} w_j^{(h)}$).

For Eqns. (5)-(6) Since p_0^R is never queued behind any packet on the path, $\sum_{h=1}^{n} w_{R0}^{(h)} = 0$. For the minimum of d_{R1}, since p_1^R is sent back to back after p_0^R to traverse all the hops of the path, using Lemma 2,

$$\sum_{h=1}^{n} w_{R1}^{(h)} = \max_{h=1,\ldots,n}\{X^{(h)}\} = \max\{S/C_f, S/C_r\}.$$

For Eqns. (7)-(8) Since $\{r_0^T, r_1^T\}$ are both elicited by p_1^T, their forward-path queueing delays (introduced by p_0^T) are the same (i.e., $\sum_{h=1}^{m} w_{T0}^{(h)} = \sum_{h=1}^{m} w_{T1}^{(h)}$). Using Lemma 2 for $h = 1, \ldots, m$,

$$\sum_{h=1}^{m} w_{T0}^{(h)} = \sum_{h=1}^{m} w_{T1}^{(h)} = \max_{h=1,\ldots,m}\{X^{(h)}\} = S/C_f^{(n)}.$$

Their reverse-path queueing delays are, however, different. For the minimum of d_{T0}, since r_0^T is never queued behind any packet on the reverse path, we have $\sum_{h=m+1}^{n} w_{T0}^{(h)} = 0$. On the other hand, since r_1^T traverses the reverse path after r_0^T, we apply Lemma 2 for $h = m+1, \ldots, n$ to yield

$$\sum_{h=m+1}^{n} w_{T1}^{(h)} = \max_{h=m+1,\ldots,n}\{X^{(h)}\} = S/C_r^{(n)}.$$

We therefore obtain Eqns. (5)–(8) by substituting the respective queueing delays in Eqn. (3). □

With the minRTTs, it is straightforward to obtain the four capacities in Prop. 2 by subtracting the respective minimum delays in Lemma 3.

PROPOSITION 2. *Given the minimums of d_{R0}, d_{R1}, d_{T0}, and d_{T1}, the four capacities can be computed as*

$$C_f = S/(\min\{d_{T0}\} - \min\{d_{R0}\}),$$
$$C_r = S/(\min\{d_{T1}\} - \min\{d_{T0}\}),$$
$$C_B = S/(\min\{d_{T1}\} - \min\{d_{R1}\}),$$
$$C_b = S/(\min\{d_{R1}\} - \min\{d_{R0}\}).$$

4.3 Implementation

We implemented TRIO using HTTP/OneProbe's probing technique [28] to provide both *client-side* and *server-side* measurements for asymmetric capacity. The client-side measurement allows a user to measure asymmetric capacity of the network path to any remote web server. Each probe packet sent by TRIO is a TCP data packet that carries a legitimate HTTP `GET` request; each response packet elicited from a web server is also a TCP data packet that contains the requested HTTP data. Therefore, each 1-RTP is the same as an HTTP/OneProbe's probe. To implement $(1,1)$-TWP, we modified the HTTP/OneProbe's probe by inserting a zero receive window (`rwnd`) in the first probe packet and a $(2 \times \text{MSS})$-byte `rwnd` in the second probe packet.

The server-side measurement, on the other hand, enables a user to perform capacity measurement without installing TRIO. To this end, TRIO serves as a web server to listen for incoming HTTP requests from remote web clients. Upon receiving an HTTP request, TRIO replies with an HTTP response that instructs the client's browser to launch a Flash object downloaded from TRIO. The Flash object enables the browser to establish a separate TCP connection with TRIO and respond to TRIO's probes. An alternative approach is to deploy a Java applet in the client's browser, but the Java implementation limits the sending rate [23] and thus the maximum capacity measured by TRIO.

4.4 Self-diagnosis tests

To improve the measurement accuracy, TRIO performs three types of self-diagnoses.

Packet loss and reordering Using HTTP/OneProbe [28], TRIO can detect loss and reordering of individual probe and response packets that could significantly affect the measurement accuracy [31]. TRIO therefore

removes all packet pairs that do not elicit the expected TCP response packets to ensure that all RTT samples used for the capacity measurement come from lossless and order-preserved probe and response packets.

Incorrect minRTT estimates Incorrect estimates of the minimums of d_{R0}, d_{R1}, d_{T0}, and d_{T1} will seriously affect the measurement accuracy. TRIO validates the minRTT estimates based on the following inequality:

$$\min\{d_{R0}\} < \min\{d_{T0}\} \leq \min\{d_{R1}\} < \min\{d_{T1}\}, \quad (9)$$

which can be easily observed from Lemma 3.

Incorrect capacity estimates Given the four capacity estimates ($\hat{C}_f$, $\hat{C}_r$, $\hat{C}_B$, and $\hat{C}_b$), TRIO can further validate their accuracy by performing internal validation based on Eqns. (10)-(11). If either equation cannot be fulfilled (to a certain precision) after sending a predefined number of probes, it is likely that the minRTT estimates have yet to converge.

$$\hat{C}_b = \min\{\hat{C}_f, \hat{C}_r\}, \quad (10)$$

$$\hat{C}_B = \max\{\hat{C}_f, \hat{C}_r\}. \quad (11)$$

To sum up, if Eqn. (9), Eqn. (10), or Eqn. (11) does not hold, TRIO invalidates the current capacity estimates and keeps sending more probes until the three equations are fulfilled.

5. EVALUATION

We evaluated TRIO empirically and compared it with AsymProbe, DSLprobe, and SProbe, whenever possible, both in a testbed and real Internet paths. Compared with the evaluation studies reported in the existing literature [32, 13, 14], our evaluation considers more general scenarios, including both FF and FR paths, and different degrees of capacity asymmetry.

5.1 Testbed evaluation

We expanded the testbed in Figure 4 by inserting a Linux router $\mathbb{R}_3$ and a 100-Mbits/s Ethernet switch $\mathbb{S}_4$ between $\mathbb{S}_3$ and the web server and attaching a cross-traffic client $\mathbb{X}_4$ to $\mathbb{S}_4$. We designated $\mathbb{R}_1$ to emulate eight different cases of capacity asymmetry and a fixed RTT of 300ms, and imposed a loading rate ρ of 20% on the corresponding path segments. Other configuration settings remained unchanged. As shown in Table 3, the first four cases correspond to FF paths, whereas the next four cases FR paths.

The probe sender ran AsymProbe, DSLprobe, and SProbe with most of their default configuration settings unchanged for 30 times. To obtain a fair comparison, we set $S_{max}/S_{min} = 1500/40$ (the maximum and minimum IP packet sizes in bytes) for all of them to achieve the greatest packet-size asymmetry. We also repeated the experiments with TRIO that dispatched an interleaved sequence of Poisson-modulated 1-RTPs and $(1,1)$-TWPs with $S = S_{max}$ and a mean probing rate of 2 Hz for 300 seconds. Moreover, we discounted DSLprobe's reverse capacity estimates by a factor of 2.65 [14] for adjusting the layer-2 overhead due to the ADSL link. To compute layer-two (Ethernet) capacity, we then applied a factor of 1518/1500 to its forward capacity estimates and a factor of 64/40 to its reverse capacity estimates. Similarly, we applied the factor of 1518/1500 to the forward capacity and reverse capacity estimates obtained by AsymProbe, SProbe, and TRIO.

Table 3 shows the evaluation results in terms of the means and 95% confidence intervals of $\hat{C}_f$ and $\hat{C}_r$. We boldfaced those results with a relative difference—$|E[\hat{C}_f] - C_f|/C_f$ or $|E[\hat{C}_r] - C_r|/C_r$—greater than 0.1. The overall results are consistent with the analytical results discussed in the last two sections: (1) TRIO's capacity estimates are accurate for all scenarios, (2) AsymProbe's capacity estimates are accurate, except for high degrees of capacity asymmetry, (3) DSLprobe's capacity estimates are accurate for some cases under FF paths, and (4) SProbe's capacity estimates are accurate only for forward capacity for some cases under FF and FR paths.

AsymProbe Since AsymProbe uses 1-RTP for both forward-path and reverse-path measurements, it is expected to encounter the case (I) error for the FF path with $C_{f/r} = 20/0.512$ and the case (II) error for the FR path with $C_{f/r} = 0.512/20$ based on Table 2.

DSLprobe For a similar reason, DSLprobe should encounter the case (I) error for the FF path with $C_{f/r} = 20/0.512$, but it turned out that DSLprobe did not report the forward capacity estimate. A study of the DSLprobe's source code [4] reveals that a sanity check withholds the forward capacity estimation when the observed forward dispersion is either less than 1.3 times (i.e., $1500/20 < 40/0.512 \times 1.3$ in this case) or greater than 15 times of the observed reverse dispersion. For the FR paths, the reverse capacity estimates are inaccurate, because DSLprobe uses $S_f = S_r$ for the reverse-path measurement, and the forward capacity estimates were all rejected by the sanity check.

SProbe Similarly, SProbe underestimates the forward capacity for the FF path with $C_{f/r} = 20/0.512$, because of the case (I) error. It is also surprising to see that its forward capacity estimates' accuracy decreases with C_f. This observation indicates that its forward-path measurement is sensitive to the cross-traffic interference when the forward PPD is small. Moreover, SProbe unexpectedly underestimates the reverse capacity for all scenarios. Inspecting the server-side raw packet traces reveals that the HTTP `GET` request failed to elicit back-to-back response packets, which was probably due to the server's limited TCP congestion window (`cwnd`). On the other hand, using HTTP/OneProbe's probing technique [28], TRIO ensures that the server has a sufficient `cwnd` to dispatch back-to-back response packets.

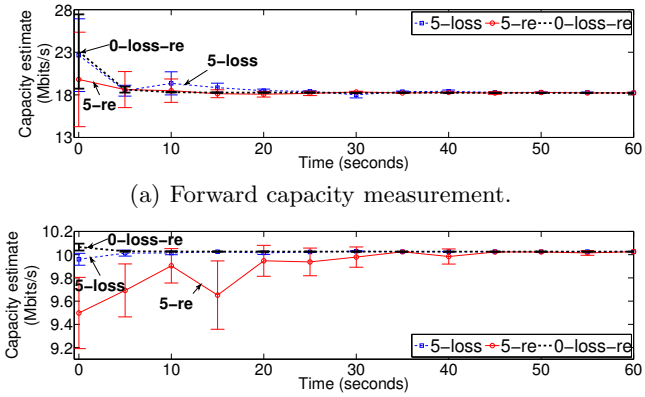
Table 3: Capacity estimates (in Mbits/s) obtained by AsymProbe, DSLprobe, SProbe, and TRIO. The symbol '-' means that the corresponding tool could not output the result.

	C_f	AsymProbe	DSLprobe	SProbe	TRIO	C_r	AsymProbe	DSLprobe	SProbe	TRIO
FF	6	5.58 ± 0.10	5.87 ± 0.07	6.50 ± 1.29	6.07 ± 0.01	0.64	0.64 ± 0.00	0.68 ± 0.00	$\mathbf{0.08 \pm 0.00}$	0.64 ± 0.00
	8	8.24 ± 0.15	7.89 ± 0.05	$\mathbf{11.72 \pm 2.24}$	8.08 ± 0.02	0.8	0.80 ± 0.00	0.85 ± 0.00	$\mathbf{0.10 \pm 0.00}$	0.80 ± 0.00
	30	31.07 ± 0.17	$\mathbf{36.80 \pm 7.37}$	$\mathbf{109.70 \pm 64.54}$	30.63 ± 0.09	10	10.07 ± 0.04	10.81 ± 0.09	$\mathbf{1.22 \pm 0.03}$	10.04 ± 0.01
	20	$\mathbf{12.56 \pm 0.20}$	-	$\mathbf{12.84 \pm 0.02}$	20.97 ± 0.21	0.512	0.51 ± 0.00	0.55 ± 0.00	$\mathbf{0.06 \pm 0.00}$	0.51 ± 0.00
FR	0.64	0.63 ± 0.04	-	0.63 ± 0.01	0.64 ± 0.00	6	5.95 ± 0.20	$\mathbf{0.87 \pm 0.01}$	$\mathbf{0.74 \pm 0.00}$	6.01 ± 0.02
	0.8	0.79 ± 0.01	-	0.86 ± 0.08	0.80 ± 0.00	8	8.04 ± 0.06	$\mathbf{1.08 \pm 0.02}$	$\mathbf{0.94 \pm 0.06}$	8.02 ± 0.00
	10	10.51 ± 0.40	-	$\mathbf{11.35 \pm 2.50}$	10.09 ± 0.01	30	31.10 ± 0.22	31.91 ± 6.96	$\mathbf{3.66 \pm 0.10}$	30.14 ± 0.05
	0.512	0.51 ± 0.01	-	0.51 ± 0.01	0.51 ± 0.00	20	$\mathbf{12.10 \pm 0.17}$	$\mathbf{0.65 \pm 0.01}$	$\mathbf{2.41 \pm 0.10}$	20.08 ± 0.02

Other evaluations for TRIO We repeated the evaluation of TRIO by changing (i) S to 240 bytes, (ii) ρ to 40%, and (iii) the designated router to $\mathbb{R}_3$ (next to the web server) for emulating the forward and reverse bottlenecks, whereas other parameter settings remained unchanged. Our results (not shown in the paper) show that under all scenarios (i)–(iii) TRIO still obtains fairly accurate forward and reverse capacity estimates for the eight capacity-asymmetric paths with less than 8% and 3% errors, respectively.

Moreover, Figure 8 shows the means and 95% confidence intervals of the capacity estimates obtained by TRIO under two adverse path conditions—a packet-dropping probability of 5% (labeled by 5-loss) and reordering every fifth packet (labeled by 5-re)—which were emulated by $\mathbb{R}_1$ for both forward and reverse paths. For comparison purpose, we also emulated a perfect path condition without loss or reordering (labeled by 0-loss-re). For each path condition, we set the degree of capacity asymmetry to $C_{f/r} = 18/1$ (in Mbits/s, emulated by $\mathbb{R}_1$) and ρ to 20%. We ran TRIO to send an interleaved sequence of Poisson-modulated 1-RTPs and $(1,1)$-TWPs with $S = 1500$ bytes and a mean probing rate of 2 Hz for 60 seconds, and repeated the experiment for 50 times.

Figure 8 shows that TRIO's estimates converge to the true values in less than 20 seconds under the perfect path condition. Although the emulated packet loss and reordering affect many probes and RTT samples, TRIO can remove them from the capacity estimation and obtain accurate estimates in less than 50 seconds.



(a) Forward capacity measurement.

(b) Reverse capacity measurement.

Figure 8: Time series of TRIO's capacity estimates under three path conditions with $C_{f/r} = 18/1$ (in Mbits/s).

5.2 Internet evaluation with emulated links

We also evaluated TRIO and the packet-dispersion methods on Internet paths with real cross traffic. To establish the ground truth, we implemented the residential broadband network model [7] using a Click router at our campus network to emulate a set of forward and reverse capacities falling in $[0.1, 20]$ Mbits/s. A measuring node was connected to the Click router via the NetMagic platform [27] to obtain packet timestamps (for measuring RTTs and PPDs) with a resolution of 8ns. We found that we were no longer able to ensure that the bottleneck link was still located in our campus network when the emulated capacities were greater than 10 Mbits/s. Here we report the results for two emulated capacity-asymmetric links: link 1 (upstream: 5 Mbits/s, downstream: 0.25 Mbits/s) and link 2 (upstream: 0.25 Mbits/s, downstream: 5 Mbits/s), which correspond to FF and FR paths, respectively.

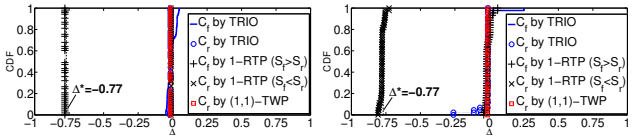
For each link, we deployed TRIO to measure the asymmetric capacities of the paths to 50 Debian mirror sites reported in [2] on 1 June 2010. These sites were located in 50 different countries; therefore, the paths' characteristics are expected to be very diverse. However, the AsymProbe's deployment failed, because its current implementation requires the remote node's control. DSLprobe and SProbe, on the other hand, could not trigger valid responses from the mirror sites with their current implementations.

To achieve the evaluation, we implemented (i) RTP measurement with $S_f > S_r$ for forward capacity (used by AsymProbe, DSLprobe, and SProbe), (ii) RTP measurement with $S_f < S_r$ for reverse capacity (AsymProbe), and (iii) TWP measurement for reverse capacity (SProbe) using HTTP/OneProbe's probing technique; and refer to the implementation as PDProbe. As a packet-pair technique, PDProbe dispatches 1-RTPs

for (i) and (ii), and $(1,1)$-TWPs for (iii). To improve the measurement accuracy, we incorporated the minimum delay sum method [20] in PDProbe to remove PPD samples distorted by cross traffic.

For each path measurement, TRIO and PDProbe were configured to send legitimate HTTP requests to fetch the same web object from each mirror site. TRIO sent an interleaved sequence of Poisson-modulated 1-RTPs and $(1,1)$-TWPs with $S = 1024$ bytes at an average rate of 2 Hz for 180 seconds. Similarly, PDProbe sent a sequence of 180 Poisson-modulated probes at an average rate of 2 Hz for each packet-dispersion measurement. Moreover, the packet sizes were $S_{max}/S_{min} = 1380/300$ (in bytes) for (i) and (ii), and $S_r = 1380$ for (iii). By using 1-RTPs with $S_f > S_r$ and $S_f < S_r$, we expect that PDProbe will underestimate the faster-path capacities—link 1's upstream capacity (case (I) in Table 2) and link 2's downstream capacity (case (II)), respectively—for all the paths.

Figure 9 plots for each link a CDF for $\Delta(\hat{c}, C) = (\hat{c} - C)/C$, the relative difference between forward (or reverse) capacity estimates $\hat{c}$ and the actual forward (or reverse) capacity C which is assumed to be the emulated link's upstream (or downstream) capacity. For both links, TRIO obtains fairly accurate forward and reverse estimates: more than 95% of the capacity estimates obtained by TRIO deviate less than 5% from the actual capacity, and similarly for the $(1,1)$-TWP's reverse estimates. On the other hand, the two types of 1-RTP measurements obtain accurate slower-path estimates for all the paths, but, as discussed earlier, they significantly underestimate the faster-path capacity due to the limited packet-size asymmetry. We also plot in each figure $\Delta^* = \Delta(\frac{S_{max}}{S_{min}/C_b}, C_B)$ which measures the expected difference between the inaccurate measurement of C_B due to the slower-path PPD (i.e., $\delta_{j-1,j} = S_{min}/C_b$) and C_B. As a result, the CDF for the faster-path estimates is close to Δ^* in each case.



(a) Link 1 ($C_{f/r} = 5/0.25$). (b) Link 2 ($C_{f/r} = 0.25/5$).

Figure 9: CDFs of the relative differences between the configured capacity and the capacity estimates.

5.3 Internet measurement with web and residential services

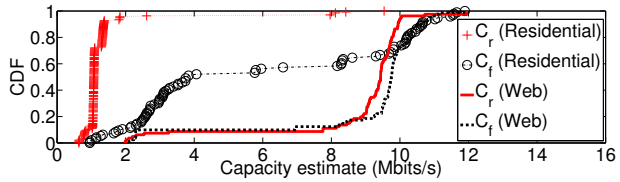
We conducted TRIO's client-side measurement from another measuring node at our campus network to measure paths to 12 public Internet web servers, and server-side measurement from the same node to measure paths to 19 residential broadband users in June 2011. The web servers were located in Austria, Canada, Denmark, Taiwan, UK, and US, whereas the residential users connected through eight ISPs in China, Hong Kong, and US. The measuring node was well-connected to a 100-Mbits/s Ethernet link and obtained RTT samples using the DAG card.

For the client-side measurement, we conducted the path measurement to the web servers in a round-robin fashion in order to avoid congestion. For each server, we launched TRIO to dispatch an interleaved sequence of 1-RTPs and (1,1)-TWPs with $S = 1240$ bytes and a mean probing rate of 2 Hz for 30 seconds, and repeated the measurement for every half an hour. After each round, we collected the RTT samples obtained in the last four hours to compute capacity estimates for all the paths. For the server-side measurement, we conducted the path measurement with $S = 616$ bytes. The measurement started when a TCP connection from a residential user's web browser was successfully established with the measuring node. The capacity estimations continued until the browser was closed. The measurement duration for each user ranges from one hour to more than eight hours.

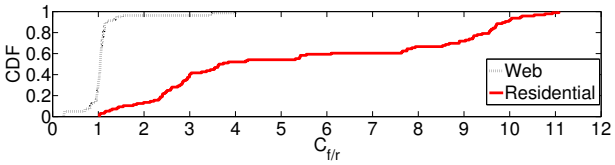
Figures 10(a)-10(b) show the CDFs of all capacity estimates for the web servers and residential users, and their corresponding forward-reverse capacity ratios. For the web servers, 75% (or 78%) estimates for forward (or reverse) capacity fall in $[9.5, 11]$ (or $[8.9, 10]$) Mbits/s, and 84% forward-reverse capacity ratios are in $[0.8, 1.2]$. Considering the diverse locations of the servers (and hence their network paths), such consistent results suggest that the bottleneck links were near our measurement node, thus shared by all the paths. We confirmed this finding with a network manager of our campus network that the measuring node's Internet connection was allocated with a symmetric capacity of 10 Mbits/s. Moreover, around 20% estimates report both capacities less than 10 Mbits/s, and this indicates that their paths involved other bottleneck links outside our campus network. We also observe some capacity estimates greater than 10 Mbits/s, which were probably due to incorrect RTT samples caused by cross traffic.

Figure 10(b) shows that the residential users exhibit a high degree of capacity asymmetry, which is typical for residential broadband links. About 93% reverse estimates (i.e., upstream capacity for the users) are in $[0.64, 1.4]$ Mbits/s, but 74% forward estimates (i.e., downstream capacity) exhibit a large variation of $[1.7, 11]$ Mbits/s. In particular, the forward estimates of four users in China and one user in US fall in $[1.7, 4]$ Mbits/s which can be further confirmed by their ISPs' advertised service plans. Moreover, the forward capacity measurements for seven users in Hong Kong are in $[9.3, 11]$ Mbits/s, but we found that their subscribed

services actually provide a downlink capacity up to 20 Mbits/s. Therefore, the measurement results for these users were probably determined by the maximum upstream capacity imposed by our campus network. We have also identified a symmetric broadband link from a China's user and have received the user's confirmation.



(a) CDF of capacity estimates.



(b) CDF of forward-reverse capacity ratios.

Figure 10: Capacity measurement results obtained by TRIO for Internet web servers (using client-side measurement) and residential broadband users (using server-side measurement).

6. DISCUSSION

A highly congested path can affect both TRIO's and AsymProbe's measurements which exploit minRTTs to filter out distorted capacity estimates. Nonetheless, only TRIO can flexibly reduce the packet size to mitigate the cross-traffic impact on the second packet in the packet pair [16, 20, 12], or increase the size when there is significant variation in capacity estimates which is possibly due to a limited time resolution supported by the measuring node [20]. Moreover, TRIO can apply the convergence test [17] and the bootstrap method [25] to detect the convergence of minRTTs.

Similar to other packet-pair techniques (e.g., AsymProbe and SProbe), TRIO may also produce incorrect estimates for multichannel bottleneck links [16, 31], because a packet pair could pass through different channels (i.e., queues) in parallel. Moreover, TRIO can only measure the peak rate for a traffic shaper based on the leaky bucket algorithm, but not its sustainable rate that happens only after a certain burst size [16]. To tackle these issues, we will explore the feasibility of extending TRIO's packet-pair probes to packet trains (i.e., k-RTP and (v, k)-TWP with $v, k > 1$).

7. RELATED WORK

Methods proposed for measuring path capacity are mostly based on active measurement (e.g., [10, 11, 29, 16, 32, 20, 13, 14]) and only a handful based on passive measurement (e.g., [24, 26]). The majority of the active methods are designed to measure one-way capacity (e.g., [10, 16]), round-trip capacity (e.g., [11, 10]), sub-path capacity [18], and per-hop link capacity (e.g., [19, 17, 6, 25, 29]). Moreover, the methods for one-way measurement are usually cooperative, whereas others are mostly non-cooperative. But the non-cooperative methods for the sub-path measurement and per-hop link capacity measurement can only characterize forward capacity. On the other hand, less than a handful of methods [32, 13, 14] target on measuring asymmetric capacity. TRIO belongs to the same category as SProbe [32] and DSLprobe [14] in terms of measuring asymmetric capacity of a round-trip path from only a single endpoint.

In general, there are two kinds of active measurement methods for tackling various capacity measurement problems. The first is using TTL-limited probe packets [19, 17, 6] or tailgating probe packets [25, 29, 18] to measure per-hop link capacity. The second is using packet pairs [21, 10, 11, 32, 20, 16, 13, 12] or packet trains [16, 15, 14] to measure path capacity. These methods obtain packet delay [19, 6, 17, 25], packet dispersion [21, 10, 11, 24, 26, 32, 30, 20, 16, 13, 14, 18], delay variation [29, 12], or packet loss rate [15] to estimate the capacity. Note that the TTL-limited and tailgating probe packets mostly obtain packet delay and delay variation, whereas the packet pair and train mostly obtain packet dispersion. In contrast, TRIO uses packet pairs to obtain delay variation for computing asymmetric capacity of a round-trip path and does not overwhelm the bottleneck link.

Minimum estimator [19, 6, 17, 25, 20, 13, 12] and mode estimator [10, 11, 24, 26, 29, 30, 16] are the most common techniques for mitigating the cross-traffic interference on path capacity and link capacity measurements. The minimum estimator determines the minimum delay from a sequence of packet delay samples. The minimum delay sum method [20] employed by AsymProbe exploits the sum of minimum packet-pair delays to remove biased PPDs. The minimum delay difference (MDDIF) method [12] estimates the unbiased PPD based on the variation of the minimum packet-pair delays. Similar to the MDDIF method, TRIO exploits the minimum estimator to mitigate the cross-traffic interference on the PPD estimation.

8. CONCLUSIONS

We presented TRIO for measuring forward and reverse capacities with three minRTTs obtained from two types of carefully crafted probes: RTP and TWP. Using the minRTTs, instead of packet dispersion, eliminates the measurement limitations suffered by the existing methods and mitigates the cross-traffic interference. We showed that integrating the RTP and TWP also enables a simultaneous measurement of the forward

and reverse capacities and their validations. As a result, TRIO is the first method that can measure any degree of capacity asymmetry. Both analytical results and empirical evaluations using our client-side and server-side implementations confirmed TRIO's capability and measurement accuracy. Future works include using TRIO to identify the location of a capacity bottleneck from multiple vantage points.

Appendix

Proof of Lemma 2 Based on the Lindley's recurrence equation for $w_j^{(h)}$ [22], we have

$$\sum_{h=1}^{n} w_j^{(h)} = \sum_{h=1}^{n} [(w_{j-1}^{(h)} + X^{(h)} - \delta_{j-1,j}^{(h-1)})^+ + q_{j-1,j}^{(h)}], \quad (12)$$

where $(x)^+ = \max\{0, x\}$, $w_{j-1}^{(h)}$ is the queueing delay of p_{j-1} at the h^{th} hop, $q_{j-1,j}^{(h)}$ is the queueing delay of p_j caused by intervening cross traffic between $\{p_{j-1}, p_j\}$ at the h^{th} hop, and $\delta_{j-1,j}^{(h)}$ is the PPD of $\{p_{j-1}, p_j\}$ at the outgoing link of the h^{th} hop and $\delta_{j-1,j}^{(0)} = 0$.

Since neither of $\{p_{j-1}, p_j\}$ experiences cross-traffic-induced queueing delay at any hops in the path, we set $w_{j-1}^{(h)} = q_{j-1,j}^{(h)} = 0,\ \forall 1 \le h \le n$, in Eqn. (12) to obtain

$$\sum_{h=1}^{n} w_j^{(h)} = \sum_{h=1}^{n} (X^{(h)} - \delta_{j-1,j}^{(h-1)})^+. \quad (13)$$

We now use mathematical induction on n for the proof.

Basis Consider $n = 1$. Since $\delta_{j-1,j}^{(0)} = 0$, Eqn. (13) becomes $w_j^{(1)} = X^{(1)} = \max_{h=1}\{X^{(h)}\}$.

Inductive step Assuming that Eqn. (4) holds for n, we show that Eqn. (4) also holds for $n+1$. By substituting Eqn. (4) into Eqn. (13) for $n+1$, we have

$$\sum_{h=1}^{n+1} w_j^{(h)} = \max_{h=1,\ldots,n}\{X^{(h)}\} + (X^{(n+1)} - \delta_{j-1,j}^{(n)})^+. \quad (14)$$

Note also that with Eqn. (6) from [30], we have

$$\delta_{j-1,j}^{(n)} = \sum_{h=1}^{n} (X_j^{(h)} - X_{j-1}^{(h)}) + \sum_{h=1}^{n} (w_j^{(h)} - w_{j-1}^{(h)}). \quad (15)$$

Since $X_j^{(h)} = X_{j-1}^{(h)} = X^{(h)},\ \forall 1 \le h \le n$, for the constant size of $\{p_{j-1}, p_j\}$, $\sum_{h=1}^{n} w_{j-1}^{(h)} = 0$, and the induction hypothesis that Eqn. (4) holds, Eqn. (15) becomes

$$\delta_{j-1,j}^{(n)} = \sum_{h=1}^{n} w_j^{(h)} = \max_{h=1,\ldots,n}\{X^{(h)}\}. \quad (16)$$

We now consider two cases from Eqn. (14): (i) $X^{(n+1)} \le \delta_{j-1,j}^{(n)}$ and (ii) $X^{(n+1)} > \delta_{j-1,j}^{(n)}$. For each case, it is not difficult to obtain Eqn. (4) from Eqns. (14)-(16).

Acknowledgement

We thank the five anonymous reviewers for their critical reviews and suggestions and Chadi Barakat, in particular, for shepherding our paper. This work is partially supported by a grant (ref. no. ITS/355/09) from the Innovation Technology Fund in Hong Kong and a grant (ref. no. H-ZL17) from the Joint Universities Computer Centre of Hong Kong.

9. REFERENCES

[1] Click Modular Router. http://read.cs .ucla.edu/click/.
[2] Debian mirror sites. http://www.debian.org/mirror/list/.
[3] Endace. http://www.endace.com/.
[4] FAB-Probe. http://www.eurecom.fr/∼btroup/fabprobe.html.
[5] netvigator. http://www.netvigator.com/eng/.
[6] pchar. http://www.kitchenlab.org/∼bmah/Software/pchar/.
[7] The residential broadband model. http://broadband.mpi-sws .org/residential/model.html.
[8] Sercomtel. http://home.sercomtelfixa.com.br/portalinternet/.
[9] TCPDUMP/LIBPCAP. http://www.tcpdump.org/.
[10] J. Bolot. End-to-end packet delay and loss behavior in the Internet. In *Proc. ACM SIGCOMM*, 1993.
[11] R. Carter and M. Crovella. Measuring bottleneck link speed in packet-switched networks. *Perform. Eval.*, 27-28, 1996.
[12] E. Chan, X. Luo, and R. Chang. A minimum-delay-difference method for mitigating cross-traffic impact on capacity measurement. In *Proc. ACM CoNEXT*, 2009.
[13] L. Chen, T. Sun, G. Yang, M. Sanadidi, and M. Gerla. End-to-end asymmetric link capacity estimation. In *Proc. IFIP Networking*, 2005.
[14] D. Croce, T. En-Najjary, G. Urvoy-Keller, and E. Biersack. Capacity estimation of ADSL links. In *Proc. ACM CoNEXT*, 2008.
[15] M. Dischinger, A. Haeberlen, K. Gummadi, and S. Saroiu. Characterizing residential broadband networks. In *Proc. ACM/USENIX IMC*, 2007.
[16] C. Dovrolis, P. Ramanathan, and D. Moore. Packet dispersion techniques and a capacity-estimation methodology. *IEEE/ACM Trans. Netw.*, 12(6), 2004.
[17] A. Downey. Using pathchar to estimate Internet link characteristics. In *Proc. ACM SIGCOMM*, 1999.
[18] K. Harfoush, A. Bestavros, and J. Byers. Measuring capacity bandwidth of targeted path segments. *IEEE/ACM Trans. Netw.*, 17(1), 2009.
[19] V. Jacobson. Pathchar: A tool to infer characteristics of Internet paths. ftp://ftp.ee.lbl.ogv/pathchar/.
[20] R. Kapoor, L. Chen, L. Lao, M. Gerla, and M. Sanadidi. CapProbe: A simple and accurate capacity estimation technique. In *Proc. ACM SIGCOMM*, 2004.
[21] S. Keshav. A control-theoretic approach to flow control. In *Proc. ACM SIGCOMM*, 1991.
[22] L. Kleinrock. *Queueing Systems, Vol. 2: Computer Applications*. Wiley-Interscience, 1976.
[23] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: Illuminating the edge network. In *Proc. ACM/USENIX IMC*, 2010.
[24] K. Lai and M. Baker. Measuring bandwidth. In *Proc. IEEE INFOCOM*, 1999.
[25] K. Lai and M. Baker. Measuring link bandwidths using a deterministic model of packet delay. In *Proc. ACM SIGCOMM*, 2000.
[26] K. Lai and M. Baker. Nettimer: A tool for measuring bottleneck link bandwidth. In *Proc. USENIX Symp. Internet Technologies and Systems*, 2001.
[27] T. Li, Z. Sun, C. Jia, Q. Su, and M. Lee. Using NetMagic to observe fine-grained per-flow latency measurements. In *Proc. ACM SIGCOMM (Demo)*, 2011.
[28] X. Luo, E. Chan, and R. Chang. Design and implementation of TCP data probes for reliable and metric-rich network path monitoring. In *Proc. USENIX Annual Tech. Conf.*, 2009.
[29] A. Pásztor and D. Veitch. Active probing using packet quartets. In *Proc. ACM SIGCOMM IMW*, 2002.
[30] A. Pásztor and D. Veitch. The packet size dependence of packet-pair like methods. In *Proc. IWQoS*, 2002.
[31] V. Paxson. *Measurements and Analysis of End-to-End Internet Dynamics*. PhD dissertation, University of California Berkeley, 1997.
[32] S. Saroiu, P. Gummadi, and S. Gribble. SProbe: A fast technique for measuring bottleneck bandwidth in uncooperative environments. In *Proc. IEEE INFOCOM*, 2002.
[33] S. Valcourt. *Broadband Services: Business Models and Technologies for Community Networks*. Wiley, 2005.