The effect of layer-2 store-and-forward devices on per-hop capacity estimation

Ravi S. Prasad
Georgia Tech.
ravi@cc.gatech.edu

Constantinos Dovrolis
Georgia Tech.
dovrolis@cc.gatech.edu

Bruce A. Mah
Packet Design
bmah@packetdesign.com

***Abstract*— Tools such as *pathchar*, *clink*, and *pchar* attempt to measure the capacity of every Layer-3 (L3) hop in a network path. These tools use the same underlying measurement methodology, which we refer to as *Variable Packet Size (VPS)* probing. The key assumption in VPS is that each L3 hop along a path increases the delay of a packet by a "serialization latency", which is the ratio of the packet size over that hop's capacity. Unfortunately, the capacity estimates of VPS tools are sometimes wrong. In this paper, we investigate the source of these errors, and show that the presence of Layer-2 (L2) store-and-forward devices, such as Ethernet switches, have a detrimental effect on the accuracy of VPS tools. Specifically, each L2 store-and-forward device introduces an additional serialization latency in a packet's delay, which results in consistent underestimation of that L3 hop's capacity. We analyze this negative effect, deriving the measured capacity of an L3 hop as a function of the L2 link capacities at that hop. Experimental results in local, campus, and ISP networks verify the model, illustrating that L2 devices should be expected in networks of diverse type and size. Finally, we characterize some other sources of error in VPS tools, such as queueing delays, limited clock resolution, variation in ICMP generation delays, and error propagation along the measured path.**

I. Introduction

Starting from Jacobson's *pathchar*, released in 1997 [2], there has been a significant interest in techniques that can measure the *link capacities*, a.k.a. *link rates*, of an IP path. The first description of such a measurement methodology was given by Bellovin in [3]. Follow-up research papers and variations of *pathchar* are Downey's *clink* [4], Mah's *pchar* [5], and Lai's *tailgating* technique in *nettimer* [6]. The motivation for such *per-hop capacity estimation tools* is that they can be used, mainly by network managers, to debug, monitor, and characterize network paths.

These tools use a common underlying measurement methodology, which we refer to as *Variable Packet Size (VPS)* probing. The key assumption in VPS is that each hop of a path increases the one-way delay of a packet by a *serialization latency*, given by the ratio of the packet size over that hop's capacity. If this is true, the VPS technique can estimate the capacity of a hop i based on the relation between the Round-Trip Time (RTT) up to hop i and the probing packet size [4]. The required RTTs for different packet sizes can be measured using *Time Exceeded* ICMP messages [7], as done by *traceroute* [8]. For completeness, the VPS technique is reviewed in §III.

Unfortunately, the capacity estimates of VPS tools are sometimes wrong. This fact has been also reported in the literature [6], [9]. The reason for the underlying errors, however, has not been thoroughly investigated. The conventional wisdom is that routers generate ICMP replies through slow forwarding paths, or that it is impossible to avoid the measurement noise due to queueing delays. The objective of this paper is to investigate the reasons and the conditions under which VPS tools produce inaccurate estimates. Our main finding is that *store-and-forward Layer-2 (L2) devices, such as switches, in a Layer-3 (L3) hop can cause significant and consistent underestimation of that hop's capacity*. The reason is that L2 store-and-forward devices introduce additional serialization latencies, which are not taken into account by the VPS model, as those devices do not decrement the TTL field of the IP header, being effectively "invisible" to IP and higher-layer protocols. Even though this issue may be known to a small number of experts in this research area[1], it has not been discussed in the literature and, to the extent of our knowledge, it is not understood by most of the network measurements community. The negative effect of L2 devices on the accuracy of VPS tools is important because such devices are commonly used both in local and campus networks, as well as in wide-area networks. We verify this capacity underestimation effect with analytical modeling, as well as experimental results in several different network paths. Finally, we examine a few other possible sources of error, such as error propagation along the path, limited clock resolution at the probing host, queueing delays, and variability in the generation delays of ICMP messages. These error sources are probabilistic, and so they are easier to detect because they result in widely variable capacity estimates in different runs of the tools.

The paper is organized as follows. §II gives a taxonomy of the related work in bandwidth estimation. The VPS measurement methodology is reviewed in §III, and its various implementations are presented in §IV. §V analyzes the effect that L2 store-and-forward devices cause to VPS estimation, while §VI illustrates the problem with experimental results. Some other possible sources of error are examined in §VII.

This work was supported by the SciDAC program of the US Department of Energy (award # DE-FC02-01ER25467) and by an equipment donation from Intel Corporation. A short abstract of this paper was presented at the Internet Measurement Workshop (IMW), November 2002 [1].

[1] Van Jacobson mentioned that 'hidden hops' cause *pathchar* errors in his MSRI talk in 1997 [2].

We conclude in §VIII.

II. Taxonomy of bandwidth estimation tools

In this section, we review the related work in the broader area of *bandwidth estimation*. Bandwidth estimation tools can be classified in several categories, depending on the specific throughput (or "bandwidth") metric of interest, and on whether the measurements are performed on a per-hop or end-to-end basis. Table I summarizes all publicly available tools that we are aware of, together with their measurement objective and methodology.

First, there are tools that measure *end-to-end capacity*. This is the maximum throughput that a path can provide to a flow when there is no other traffic in the path. The capacity is also referred to as "bottleneck bandwidth". The underlying measurement methodology in such tools is usually a variation of packet pair or packet train probing. These methods are studied in detail in [10], [11], [12], [13], [14].

A second class is the tools that measure *per-hop capacity*. This is the maximum throughput that a single L3 hop can provide to a flow when there is no other traffic in that hop. The end-to-end capacity of a path is the minimum per-hop capacity, among all hops in that path. The underlying measurement methodology in such tools is the *Variable Packet Size (VPS) probing* technique, that we review in §III.

The *available bandwidth* of a network path is the maximum throughput that a path can provide to a flow, given the current traffic load in the path [15]. Measuring available bandwidth is much harder than measuring capacity, since the former is a dynamically varying metric. Measurement methodologies that attempt to estimate some form of available bandwidth have been proposed in [13], [15], [16], [17], [18], [19].

Another related throughput metric is the *Bulk-Transfer-Capacity (BTC)* [20]. The BTC of a path in a certain time period is the throughput of a bulk TCP transfer, when the transfer is only limited by the network resources and not by limitations at the end-systems. The BTC can be measured with *Treno* or *cap* [21]. Similarly, the throughput of large TCP transfers using parallel streams can be measured using *Iperf* [22], or similar TCP-based tools.

III. Variable Packet Size (VPS) probing

In this section, we briefly review the VPS measurement methodology; a more comprehensive study is given in [4].

An important requirement in VPS probing is to be able to measure the RTT of a packet from the sender up to a certain hop I. This is possible using the *Time-To-Live (TTL)* field of the IP header. Each L3 device along the path decrements the TTL before forwarding the packet to the next hop. If an L3 device receives a packet with zero TTL, it discards the packet and sends an ICMP *Time Exceeded* reply to the sender [7]. This technique is used by *traceroute* to identify the sequence of L3 devices in a network path [8]. VPS tools estimate the RTT up to each hop of the path sending packets with increasingly larger TTLs, and measuring the time interval until the receipt of the corresponding ICMP replies. We note that the last L3 device of the path, which is the receiving host, is detected using the ICMP *Port Unreachable* option.

Let us analyze the components of an RTT measurement (see Figure 1). Consider a path from a sender $\mathcal{SND}$ to a receiver $\mathcal{RCV}$ that consists of $H \geq 1$ hops. The capacity of each hop is C_i, for $i = 1 \ldots H$. The RTT from $\mathcal{SND}$ to hop I for a packet of size L can be measured setting the TTL of the packet to I. The RTT $T_I(L)$ of the packet is

$$T_I(L) = \sum_{i=1}^{I} \left(\frac{L}{C_i} + \tau_i^f + d_i^f + \frac{L_{ICMP}}{C_i^r} + \tau_i^r + d_i^r \right) \quad (1)$$

In the previous equation, the fraction L/C_i is the *transmission* or *serialization latency* of a packet of size L at hop i. The terms τ_i^f and d_i^f are the propagation and queueing delays, respectively, of the packet at hop i of the *forward* path, from $\mathcal{SND}$ to hop I. The terms L_{ICMP}/C_i^r, τ_i^r, and d_i^r are the serialization, propagation, and queueing delays, respectively, of the ICMP reply at hop i of the *reverse* path, from hop I to $\mathcal{SND}$. For simplicity of notation, we assume that the forward and reverse paths go through the same sequence of links and routers; this assumption is not required by the VPS methodology, however. The serialization delays are introduced in store-and-forward packet forwarding devices, such as routers, because it takes L/C_i time units to transmit a packet of size L onto a link of capacity C_i; we will return to this crucial point in §V. The delay terms τ_i^f and τ_i^r are due to the finite speed of propagation in physical-layer links, and they also include all constant per-packet processing in routers. The queueing delays d_i^f and d_i^r can be introduced in router/switch buffers, when it is not possible to transmit a packet immediately.

The VPS technique makes the following crucial assumption: *if we measure the RTT $T_I(L)$ up to hop I with several probing packets, it is likely that the minimum RTT measurement $\tilde{T}_I(L)$ resulted from a packet, and a corresponding ICMP reply, which did not experience any queueing delays*. When this is the case for a certain RTT measurement $\tilde{T}_I(L)$, we have that $d_i^f = d_i^r = 0$, and so

$$\tilde{T}_I(L) = \sum_{i=1}^{I} \left(\frac{L}{C_i} + \tau_i + \frac{L_{ICMP}}{C_i^r} \right) \quad (2)$$

where $\tau_i = \tau_i^f + \tau_i^r$ includes the constant delays at hop i, in both the forward and reverse paths. It is important that the ICMP reply's size L_{ICMP} is constant, normally 32 bytes [7][2]. Consequently, the minimum RTT measurement $\tilde{T}_I(L)$ can be expressed as

$$\tilde{T}_I(L) = \sum_{i=1}^{I} \left(D_i + \frac{L}{C_i} \right) \quad (3)$$

where $D_i = \tau_i + \frac{L_{ICMP}}{C_i^r}$ is independent of L.

Suppose now that we measure the minimum RTT for different packet sizes L, with the constraint that L is not

[2]The size of an ICMPv6 Time Exceeded packet depends on the size of the corresponding IPv6 probing packet, causing a problem for VPS probing.

Tool	Author	Measurement objective	Methodology
bprobe	Carter [16]	End-to-End Capacity	Packet Pairs
nettimer (A)	Lai [11]	End-to-End Capacity	Packet Pairs
sprobe	Saroiu [23]	End-to-End Capacity	Packet Pairs
pathrate	Dovrolis [14]	End-to-End Capacity	Packet Pairs & Trains
pathchar	Jacobson [2]	Per-Hop Capacity	Variable Packet Size
clink	Downey [24]	Per-Hop Capacity	Variable Packet Size
pchar	Mah [5]	Per-Hop Capacity	Variable Packet Size
nettimer (B)	Lai [6]	Per-Hop Capacity	Variable Packet Size (tailgating)
pipechar	Guojun [25]	End-to-End Bottleneck	Packet Trains
cprobe	Carter [16]	End-to-End Avail-BW	Packet Trains
pathload	Jain & Dovrolis [19]	End-to-End Avail-BW	Self-Loading Periodic Streams
TReno	Mathis [20]	Bulk-Transfer-Capacity	Emulated TCP throughput
cap	Allman [21]	Bulk-Transfer-Capacity	Standardized TCP throughput
IPerf	NLANR-DAST [22]	Maximum TCP throughput	Parallel TCP streams

TABLE I

TAXONOMY OF PUBLICLY-AVAILABLE BANDWIDTH ESTIMATION TOOLS.

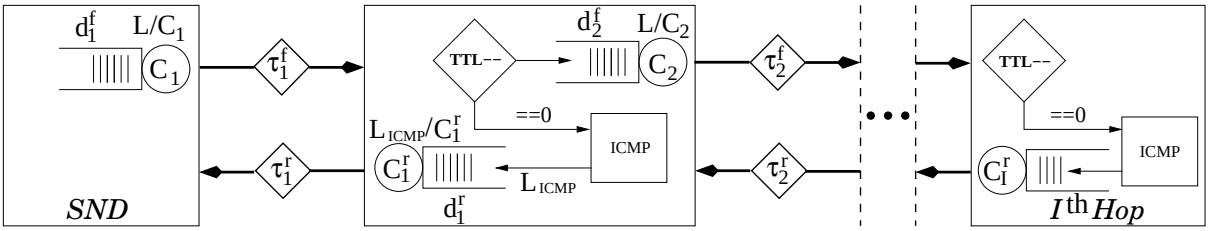

Fig. 1. RTT components in VPS probing.

larger than the path Maximum Transmission Unit (MTU)[3]. From Equation 3, it is easy to see that the relation between the minimum RTT and the packet size L is linear. Thus,

$$\tilde{T}_I(L) = \alpha_I + \beta_I L \tag{4}$$

where $\alpha_I = \sum_{i=1}^{I} D_i$, while β_I *is the slope of the minimum RTT measurements* $\tilde{T}_I(L)$ *as* L *varies*. The RTT slope at hop I is given by

$$\beta_I = \sum_{i=1}^{I} \frac{1}{C_i} \tag{5}$$

In VPS probing, we *experimentally measure the RTT-slope* β_I *at each hop* I *of the path, and then calculate the per-hop capacity* C_I *from the RTT slope differences* $\beta_I - \beta_{I-1}$ *between successive hops*. In more detail, suppose that we have measured the slope β_1 at the first hop. The capacity C_1 can be then computed as $C_1 = 1/\beta_1$. Applying induction, suppose that we have already estimated the capacity of the first I hops using the slope measurements $\beta_i, i = 1 \ldots I$. The capacity of hop $I + 1$ can be then estimated as

$$C_{I+1} = \frac{1}{\beta_{I+1} - \beta_I} \tag{6}$$

from the RTT slopes β_I and β_{I+1}. This iterative approach can be applied, for example, in the RTT measurements of Figure 2 to estimate the capacities of a two-hop path.

[3]IP fragmentation adversely affects the accuracy of VPS probing.

IV. VPS TOOLS AND VARIATIONS

In this section, we review four publicly-available per-hop capacity estimation tools, which are based on VPS probing.

pathchar: The first per-hop capacity estimation tool was *pathchar*, announced by V. Jacobson in 1997 [2]. The source code for *pathchar* was never released, however. It is known, though, that the tool follows closely the VPS methodology presented in the previous section.

clink: This is an open source implementation of the VPS probing, presented by A. Downey in 1999 [24]. The primary differences between *pathchar* and *clink* are that the latter uses an "even-odd" technique, described in [4], to generate interval capacity estimates, and that when it encounters a routing instability, it collects data for all the paths it encounters, until one of the paths generates enough data to yield an estimate.

pchar: This is also an open source implementation of the VPS probing methodology, announced by B. Mah in 1999 [5]. *pchar* runs on several Unix platforms, it provides kernel-level timestamps (via the *pcap* library), offers the option of three linear regression algorithms for the estimation of the RTT slopes, and it supports the use of several different types of probe packets.

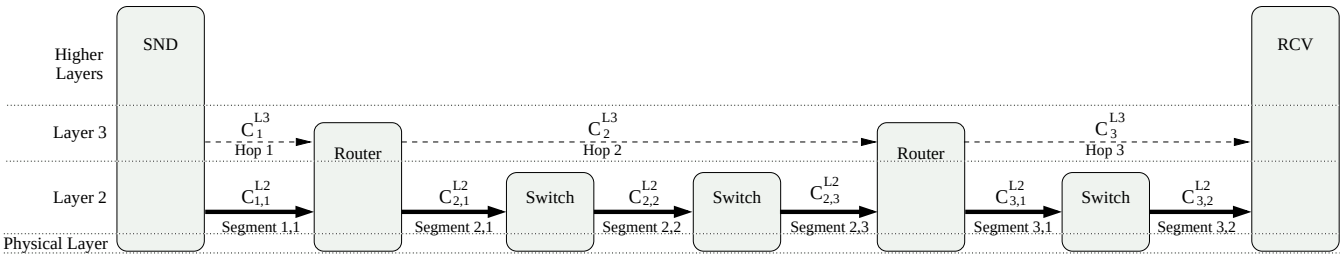

Fig. 3. The capacity of L3 hops and L2 segments in a network path.

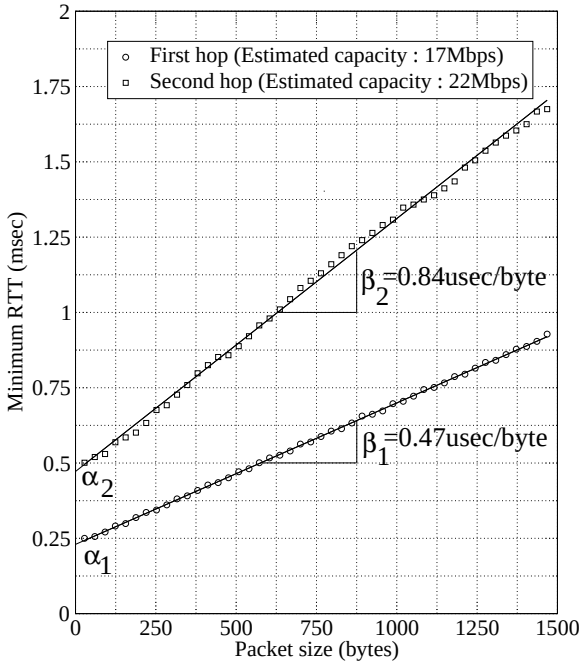

Fig. 2. VPS probing example: RTT slopes at first two hops.

nettimer: This tool is based on a combination of the VPS and packet pair probing methodologies [6][4]. As in other VPS tools, *nettimer* sends probing packets of variable size, measures the minimum delay for each size, and estimates the RTT slope at a hop using linear regression. A major difference is that the RTTs do not require ICMP *Time-Exceeded* replies from the path routers. Instead, *nettimer* uses an interesting *packet tailgating* technique, which only requires some cooperation from the receiving host. The tailgating code in nettimer is available, but its user-interface is not documented and not maintained. Consequently, the experimental results of §VI do not include measurements with *nettimer*. In personal correspondence, the author of the tool has confirmed that the tailgating technique is as susceptible to layer-2 store-and-forward devices as the previous three tools [26]. Also, the sources of error described in §VII apply to the tailgating technique, but their exact effect is likely to differ from the analysis given there.

[4]Note that *nettimer* is the name of a tool that can do either per-hop capacity estimation (using the packet tailgating technique of [6]), or end-to-end capacity estimation (using the packet pair technique of [11]). Here, we focus on the former technique.

V. The effect of L2 devices on VPS probing

In this section, we show that L2 store-and-forward devices cause capacity underestimation in VPS probing.

First though, it is important to distinguish between IP layer, or L3 links, and data-link layer, or L2 links. To differentiate between the two, we refer to L3 links as *hops* and to L2 links as *segments*. Each L3 hop includes at least one L2 segment, since the L3 connectivity is provided on top of L2 connectivity (see Figure 3). Different links are interconnected with *packet forwarding devices*, or simply *devices*. We characterize devices as L3 or L2, depending on whether they operate at the IP layer. Consequently, *L3 devices decrement the Time-To-Live (TTL) field of the IP header before forwarding the packet to the next hop, while L2 devices do not*. So, L2 devices cannot be detected using ICMP, and they are essentially invisible to VPS probing. We note that IP routers are by definition L3 devices. There are also several "L3 switch" products which operate at the IP layer, providing the forwarding performance that was traditionally feasible only with L2 switches.

Another important distinction is between store-and-forward devices and cut-through devices [27]. A *store-and-forward* device receives and buffers the entire packet before forwarding it to the next link. It is exactly this behavior that creates the serialization latency terms L/C_i in Equation 1. A *cut-through* device, on the other hand, needs to only receive a packet's header before it starts forwarding the packet to the next link. In other words, cut-through devices overlap the receipt and the transmission of a packet, avoiding most of the serialization latencies. L3 routers and many L2 switches are store-and-forward devices. Most hubs and physical-layer repeaters are cut-through devices, and, as will become clear later in this section, they have no impact on the accuracy of VPS probing.

Our model of a network path, including both L3 and L2 devices, is shown in Figure 3. An L3 hop i consists, in general, of $M_i \geq 1$ L2 segments. The first segment of a hop is the outgoing interface in the corresponding L2 device. Additional L2 devices, such as switches of various technologies, can follow in that hop, however, before the next L3 device. The presence of such intermediate L2 devices (and segments) is

common in LANs and campus networks. L2 switches are also used in some backbone networks to implement a fully-connected topology at the IP layer.

Each segment j of hop i has a capacity of $C_{i,j}^{L2}$ bits-per-second (bps), meaning that it can transmit a packet of size L bits in $L/C_{i,j}^{L2}$ seconds. In the rest of the paper we assume constant link capacities, ignoring technologies such as certain wireless links or traffic shapers, in which the capacity can vary with the underlying error rate or traffic burstiness. The capacity C_i^{L3} of hop i is defined as the minimum of the L2 segment capacities in that hop.

$$C_i^{L3} = \min_{j=1\ldots M_i} \{C_{i,j}^{L2}\} \tag{7}$$

Given that L2 devices are invisible to IP and higher-layer protocols, it is clear that VPS probing cannot measure the capacity of each segment, at least with IP and ICMP packets. Instead, the objective is to estimate C_i^{L3} for each hop $i = 1 \ldots H$ of an end-to-end path.

Let us now examine the impact of L2 store-and-forward devices on VPS probing. Suppose that the first hop of the path consists of M_1 segments, or equivalently, M_1 store-and-forward L2 devices. Each segment $j = 1 \ldots M_1$ in that hop introduces a latency of $L/C_{1,j}^{L2}$ to a packet of size L, because of the serialization latency that store-and-forward devices cause. Consequently, following the derivations that led to Equation 3, the minimum RTT for a packet of size L is expected to be

$$\tilde{T}_1(L) = D_1 + \sum_{j=1}^{M_1} \frac{L}{C_{1,j}^{L2}} \tag{8}$$

Thus, the RTT slope of the first hop is

$$\beta_1 = \sum_{j=1}^{M_1} \frac{1}{C_{1,j}^{L2}} \tag{9}$$

and so the capacity estimate for the first hop, according to VPS, is

$$\hat{C}_1^{L3} = \frac{1}{\sum_{j=1}^{M_1} \frac{1}{C_{1,j}^{L2}}} \tag{10}$$

as opposed to $C_1^{L3} = \min_{j=1\ldots M_1} C_{1,j}^{L2}$, which is the correct value of the capacity. It is easy to see that if $M_1 > 1$ then $\hat{C}_1^{L3} < C_1^{L3}$, meaning that *the capacity of the first hop will be underestimated*. For the special case that all L2 segments have the same capacity ($C_{1,j}^{L2} = C_1^{L2}$ for $j = 1 \ldots M_1$), we have that

$$\hat{C}_1^{L3} = \frac{C_1^{L2}}{M_1} \tag{11}$$

Figure 4 illustrates the previous result with an example of one hop with two segments. The timeline shows the delays that a packet of size L_1 would encounter in the first hop, together with the delays of the corresponding ICMP reply. In this example, we also assume that the switch introduces a fixed delay $\alpha_2 - \alpha_1$; that constant delay, however, has no impact on the RTT slope and the capacity estimate. Instead, it is the switch's serialization latency that causes the increased

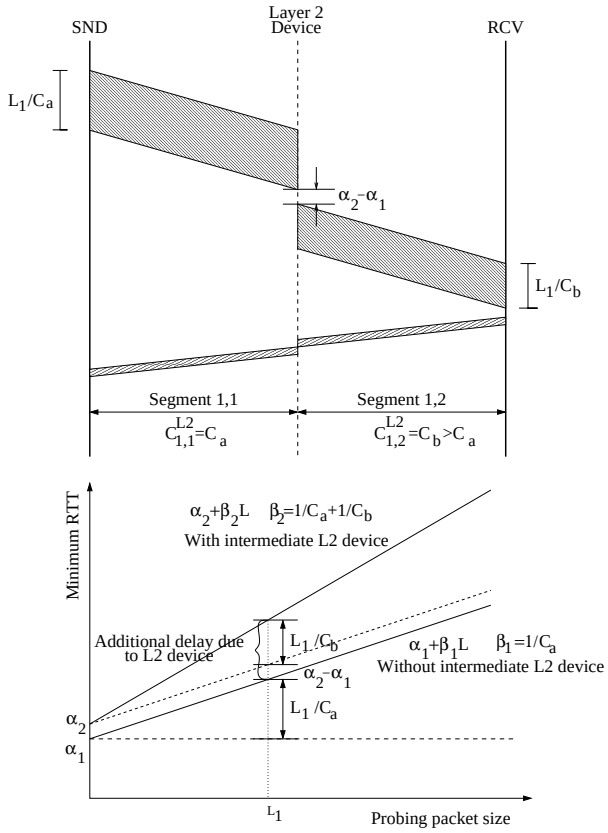

Fig. 4. An L2 device introduces an additional store-and-forward delay, causing capacity underestimation.

RTT slope, as the lower part of Figure 4 shows, leading to capacity underestimation.

Let us now examine whether L2 devices at a hop i can affect the capacity estimate of hop $i+1$. This is an important issue, because the VPS methodology estimates the capacity of each hop using the RTT slope at the previous hop (see Equation 6). Suppose that the capacity of the first hop has been underestimated as in Equation 10, and that the second hop consists of only one L2 segment with $C_2^{L3} = C_{2,1}^{L2}$. The RTT slope at the second hop will then be

$$\beta_2 = \frac{1}{C_{2,1}^{L2}} + \sum_{j=1}^{M_1} \frac{1}{C_{1,j}^{L2}} \tag{12}$$

and from Equation 6 the capacity estimate for the second hop will be

$$\hat{C}_2^{L3} = \frac{1}{\beta_2 - \beta_1} = C_{2,1}^{L2} \tag{13}$$

which is the correct value.

Even though the previous derivations were made for the first two hops in a path, it is straightforward to apply them inductively to every hop in the path. To summarize the results of this section, *the presence of more than one ($M_i > 1$) L2 store-and-forward devices at hop i causes capacity underestimation at hop i, but it does not affect the capacity estimate at hop $i+1$, or at any subsequent hop.*

A. Cut-through L2 devices

Cut-through L2 devices forward a packet to its next segment after receiving a fixed-length initial part of the packet. That part includes, normally, only the L2 (or MAC) header, which always appears at the start of a packet. Let us denote by L_H the initial number of bytes that a cut-through device needs to receive, before forwarding a packet of size L ($L_H < L$). Also, let C^{L2} be the capacity of the corresponding segment. Note that the segment will introduce a delay L_H/C^{L2} in the processing of a packet, as opposed to L/C^{L2} that a store-and-forward would introduce. This is because cut-through devices overlap the receipt and transmission of a packet, after the initial L_H bytes have been received and processed. Consequently, *cut-though devices introduce a constant delay term in the RTT of a packet, so they do not affect the RTT slope or the capacity estimate of VPS probing*.

B. ATM switches

A special case of cut-through devices is that of ATM switches. In ATM, an IP packet is segmented into a number of fixed-size cells, after some ATM Adaptation Layer (AAL) *encapsulation*. The segmentation takes place at the last store-and-forward device in the path, before entering the "cloud" of ATM segments. An ATM segment transfers cells independently, and so it does not wait to receive all cells of a packet before forwarding them to the next segment. Suppose that L_c is the ATM cell size, and L_1 is the probing packet size, ignoring for simplicity the extra bytes for AAL encapsulation. The timeline for the transmission of the packet through a hop that includes an ATM switch is shown in Figure 5. The timeline looks like a staircase, because a packet of size L_1 is fragmented into $\lceil L_1/L_c \rceil$ ATM cells. What is most important though, is that the ATM switch increases the packet's RTT by a constant, without affecting the overall RTT slope on which the VPS capacity estimate is based on. So, *ATM switches do not affect the accuracy of VPS probing, as long as the probing packet size can be much larger than the ATM cell size L_c*.

VI. Experimental results

In this section, we present experimental data that resulted from the publicly-available VPS tools *pathchar*, *clink*, and *pchar* on local-area, campus-wide, and wide-area network paths. A first objective in these experiments is to verify the negative effect of L2 store-and-forward devices on the accuracy of VPS probing. A second objective is to examine whether the problem of L2 devices appears only in local-area networks, which are often built exclusively based on Ethernet switches, or whether the problem is more general. A third objective is to observe the nature and magnitude of errors that are not related to L2 devices; such errors are investigated in more detail in §VII. In the following, we report the capacity estimates that resulted from 15-30 independent runs of each tool, as ranges.

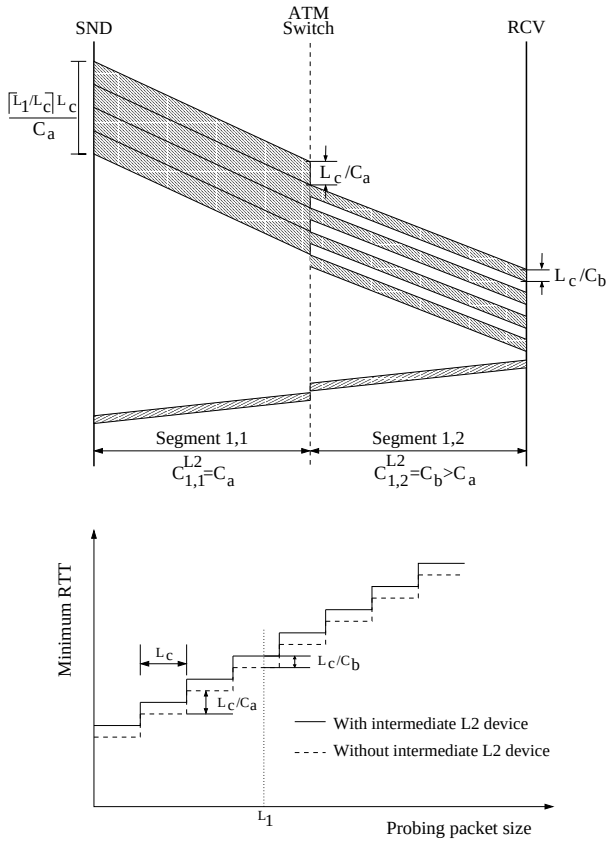

Fig. 5. Timeline of a packet transfer through an ATM segment.

A. Local Area Network (LAN) link

The first experiment refers to a single-hop path from *orion* to *sirius*, two hosts that are located at a CS lab at Univ-Delaware. Both machines have Fast Ethernet network interfaces (100Mbps), and they are connected through an HP4000 L2 Fast Ethernet switch. Note that, following the notation of §V, this is a single-hop path (H=1) with an intermediate L2 device at the first hop (M_1=2). Both L2 segments are Fast Ethernet interfaces, and so the capacity of the first hop is $C_1^{L3} = C_{1,1}^{L2} = C_{1,2}^{L2} = 100$Mbps. This is also the end-to-end capacity.

Tool	Capacity estimate
pathchar	49.0±1.5Mbps
clink	47.5±1.0Mbps
pchar	47.0±1.0Mbps
pipechar	93.5±3.0Mbps
pathrate	97.5±0.5Mbps
bprobe	95.5±2.0Mbps
Nominal capacity	100.0 Mbps

TABLE II
CAPACITY ESTIMATES FROM *orion.pc.cis.udel.edu* TO *sirius.pc.cis.udel.edu*.

Table II shows the capacity estimates for this path from six different tools. The first three are the VPS tools *pathchar*, *clink*, and *pchar*, while the last three tools (*pipechar*, *pathrate*,

and *bprobe*) are end-to-end bandwidth estimation tools that are based on the dispersion of packet pairs and trains (see Table I). The VPS tools fail to measure the nominal capacity of this path, due to the presence of the Fast Ethernet L2 switch. According to the model of §V, the (incorrect) capacity that a VPS tool would estimate in this path is $\hat{C}_1^{L3}$ = 100/2 = 50Mbps (see Equation 11). This theoretical value is actually quite close to the values that *pathchar*, *clink*, and *pchar* produce. The last three tools, on the other hand, are based on end-to-end dispersion techniques, and so they are not affected by the presence of intermediate L2 devices[5].

B. Campus path

We next experimented with the paths between two Univ-Delaware hosts, *orion.pc.cis.udel.edu* and *tsunami.coastal.udel.edu*. The two paths go through two departmental networks and through the Univ-Delaware Network Systems and Services (NSS) backbone. The L3 and L2 paths are shown in Figure 6. Note that the two paths are asymmetric.

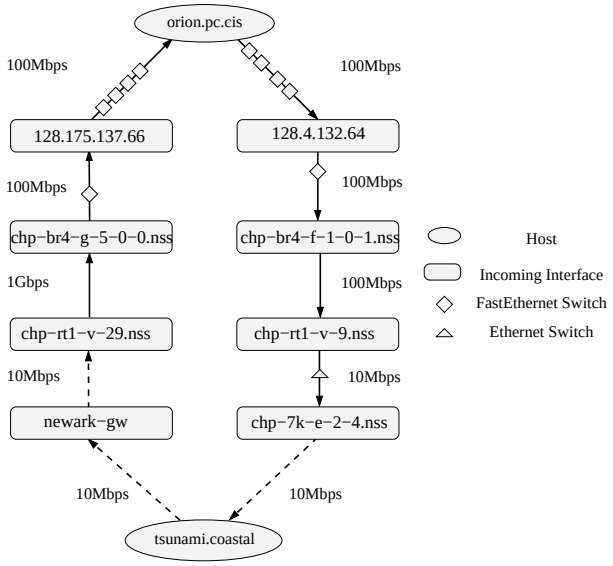

Fig. 6. The L3 and L2 topologies for two paths at the Univ-Delaware campus. All names share the *udel.edu* suffix. The dashed arrows represent hops for which we do not know the underlying L2 infrastructure.

The capacity estimates of the VPS tools are given in Tables III and IV. We summarize those results with the following observations:

1) The three VPS tools are usually consistent with each other.
2) The VPS tools are accurate in measuring Ethernet and Fast Ethernet hops, when there are no intermediate L2 switches in those hops.
3) The capacity is significantly underestimated in all hops that include intermediate L2 switches.
4) The capacity estimates that we can calculate from Equation 10, for hops with known L2 segments are close to the measurements of VPS tools, even though the agreement is far from perfect in some cases.

As an illustration of the last point, consider the path from *orion* to *tsunami*. The first hop has M_1=5 Fast Ethernet segments. According to Equation 10, the VPS capacity estimate is expected to be 100/5=20Mbps, which is close to the 17Mbps estimate of the VPS tools. Admittedly, though, the VPS measurements are closer to what one would expect with 6 Fast Ethernet segments (100/6≈16.7Mbps). Similarly, the capacity of the second hop in the same path is measured in the 55-75Mbps range, while Equation 10 predicts only 50Mbps. The third hop is a Fast Ethernet interface which is estimated correctly by all tools, as there are no intermediate L2 devices; the measurements have a significant variation however (±20Mbps). For the fourth hop we would expect 5Mbps, while the tools measure around 5.6Mbps. The measurements for the last hop indicate the presence of at least one switch, but we were unable to verify that part of the L2 infrastructure.

In the reverse path, from *tsunami* to *orion*, we do not have complete knowledge of the L2 infrastructure in the first two hops. The results seem to indicate though the presence of an Ethernet switch in the first hop, and the absence of any switches in the second hop. The third hop in the path from *tsunami* to *orion*, which is a Gigabit Ethernet link, gives widely varying results, even though it does not include L2 segments. Some other possible error sources are discussed in §VII. At the fourth hop of that path we expected 50Mbps, but the measurements varied around 40Mbps. Finally, the presence of four intermediate Fast Ethernet switches at the last hop should lead to a capacity estimate of 20Mbps. It is only *pchar* that measured something close to that value, however.

C. Access and Wide Area Network (WAN) links

We also attempted to measure per-hop capacities in several WAN paths. Unfortunately, we could not get consistent capacity estimates using any of the VPS tools in such paths. The results often vary by two orders of magnitude, from a few Mbps to hundreds of Mbps, indicating strong probabilistic sources of measurement errors. The nature of such errors is the subject of the next section. For now, we simply present capacity measurements for the two access links of the Univ-Delaware, as well as for two links in the corresponding backbone providers (Abilene and VoiceNet).

The first entry at Table V refers to the Univ-Delaware access link to VoiceNet. The outbound network interface at the Univ-Delaware gateway is a Fast Ethernet (*chp-br4-f-1-0-1.nss.udel.edu*), which is rate-limited to 45Mbps at the VoiceNet end. Note that the VPS tools underestimate that hop's capacity at about 30Mbps. This value can be explained if we assume that there is an intermediate Fast Ethernet switch in that hop at the VoiceNet end of the hop, because $1/45 + 1/100 \approx 1/31$ (see Equation 10). Unfortunately, we could not verify with the VoiceNet engineers whether such a Fast Ethernet switch exists at their end of that hop. The second

[5]As a side-note, packet dispersion techniques can also be significantly error-prone along heavily-loaded paths [14].

L3 hop	Nominal capacity	*pathchar*	*clink*	*pchar*
from *orion.ps.cis* to *128.4.132.64*	100Mbps	17.0±0.0	17.0±0.0	17.0±0.4
from *128.4.132.64* to *chp-br4-f-1-0-1.nss*	100Mbps	62.2±7.2	64.7±9.3	62.3±9.1
from *chp-br4-f-1-0-1.nss* to *chp-rt1-v-9.nss*	100Mbps	100.5±15.0	100.3±22.0	101.9±26.0
from *chp-rt1-v-9.nss* to *chp-7k-e-2-4.nss*	10Mbps	5.75±0.15	5.6±0.1	5.7±0.1
from *chp-7k-e-2-4.nss* to *tsunami.coastal*	10Mbps	4.5±0.1	3.7±0.1	6.5±0.6

TABLE III

CAPACITY ESTIMATES FOR THE PATH FROM *orion.pc.cis.udel.edu* TO *tsunami.coastal.udel.edu*.

L3 hop	Nominal capacity	*pathchar*	*clink*	*pchar*
from *tsunami.coastal* to *newark-gw*	10Mbps	4.05±0.05	4.0±0.0	4.0±1.2
from *newark-gw* to *chp-rt1-v-29.nss*	10Mbps	10.5±0.5	10.8±0.4	11.1±0.9
from *chp-rt1-v-29.nss* to *chp-br4-g-5-0-0.nss*	1000Mbps	613.33±150.0	414.70±580.0	450.2±110.0
from *chp-br4-g-5-0-0.nss* to *128.175.137.66*	100Mbps	38.3±1.7	39.9±6.0	35.6±8.8
from *128.175.137.66* to *orion.pc.cis*	100Mbps	6.95±0.5	6.1±0.2	21.5±7.8

TABLE IV

CAPACITY ESTIMATES FOR THE PATH FROM *tsunami.coastal.udel.edu* TO *orion.pc.cis.udel.edu*.

L3 hop	Nominal capacity	*pathchar*	*clink*	*pchar*
from *chp-br4-f-1-0-1.nss.udel.edu* to *delaware-gw-f2-0.voicenet.net*	45Mbps	30.5±3.5	30.3±5.6	28.3±5.6
from *delaware-gw-f2-0.voicenet.net* to *delaware2-gw-H2-0-T3.voicenet.net*	45Mbps	44.6±20.0	48.0±1.6	45.2±10.0

TABLE V

CAPACITY ESTIMATES FOR THE UNIV-DELAWARE ACCESS LINK TO VOICENET, AND FOR A VOICENET EDGE LINK.

hop at Table V is an edge T3 link in the VoiceNet network. All three VPS tools manage to estimate its capacity correctly, but with significant statistical variation in the case of *pathchar* and *pchar*.

Table VI refers to the PoS OC-3 access link from Univ-Delaware to Abilene, and to a PoS OC-48 core link in the Abilene network. The VPS tools consistently underestimate the capacity of the OC-3 link to a value that is close to 80Mbps. Unfortunately, we could not get information about the exact L2 infrastructure at the Abilene end of the hop. We note, however, that the presence of an intermediate OC-3 L2 switch in the measured hop would result (from Equation 10) to a capacity estimate of approximately $155/2 = 77.5$Mbps, which is close to what the three tools measure. Finally, the VPS tools do not manage to get a reasonable estimate for the OC-48 core link in the Abilene network. The capacity of that link is very high (2.48Gbps), and as shown in the next section, the ability of VPS tools to measure links in that bandwidth range is limited by the measurement host's clock resolution.

VII. OTHER SOURCES OF ERROR

In this section, we examine some other sources of error in VPS probing. As will become clear next, these errors are probabilistic, in the sense that successive measurements can lead to widely different results, as opposed to the consistent underestimation errors caused by L2 devices. In the following, we assume that there are no intermediate L2 devices in the measured path.

L3 hop	Nominal capacity	*pathchar*	*clink*	*pchar*
from *chp-br4-f-1-0-1.nss.udel.edu* to *abilene-wash-gsr.nss.udel.edu*	155Mbps	82.6±8	82.6±3.2	82.6±7.0
from *abilene-wash-gsr.nss.udel.edu* to *atla-wash.abilene.ucaid.edu*	2480Mbps	$460\pm^{800}_{200}$	$520\pm^{680}_{410}$	$1031\pm^{12600}_{800}$

TABLE VI

CAPACITY ESTIMATES FOR THE UNIV-DELAWARE ACCESS LINK TO ABILENE, AND FOR AN ABILENE OC-48 CORE LINK.

Path length I	ρ=0.2	ρ=0.4	ρ=0.6	ρ=0.8
1	2	3	5	11
2	3	6	14	57
4	5	17	89	1438
6	8	49	562	35977
8	13	136	3515	899447
10	21	380	21959	22486182

TABLE VII

MINIMUM NUMBER OF PACKETS K SO THAT $P(I,K) \geq 0.9$.

A. Effect of traffic load

As mentioned in §III, a key assumption in VPS probing is that the minimum RTT measurement for each packet size does not include any queueing delays. If the network load is significant, however, this may not be true even with a large number of probing packets and RTT measurements.

To examine this issue quantitatively, let us assume that probing packets arrive *at each link* as a Poisson process[6]. Even though this is a crude assumption, our objective here is simply to illustrate that measuring a queueing-free RTT can be quite hard in loaded paths. Based on the "Poisson Arrivals See Time Averages" (PASTA) property [28], the probability that a probing packet will not experience any queueing at a link i is $1-\rho_i$, where ρ_i is the fraction of time that the link i is busy, or equivalently, the utilization of link i. Consequently, the probability that a probing packet will not experience any queueing delay at the first I links of the path is

$$P(I) = \prod_{i=1}^{I}(1-\rho_i) \tag{14}$$

and so, the probability that at least one out of K probing packets will not see any queuing delay in the first I links is

$$P(I,K) = 1 - [1-P(I)]^K \tag{15}$$

Table VII shows the number of probing packets K per link and per packet size, so that the probability that at least one probing packet sees no queuing delays is more than 90%. The utilization ρ_i is assumed to be the same at all links. Note that K is impractically large when the path includes more than 5-6 links that are 60%, or more, loaded. We note that the default number of probing packets per link and per packet size is 32 for *pathchar*, 8 for *clink*, and 32 for *pchar*. Consequently, it is likely that some of the reported errors in VPS tools are due to non-zero queueing delays, and they may be avoided with a larger number of probing packets, especially for links that are further away along the path[7].

[6]This may not be true even when probing packets are sent from their source as a Poisson stream, because the interarrivals of probing packets can be modified in the network.

B. Effect of non-zero queuing delays

As shown in the previous paragraph, high utilization or long paths can introduce non-zero queuing delays, even in the minimum RTT measurement. Suppose that we want to estimate the capacity C of a link with only two probing packet sizes L_1 and L_2. Let the minimum RTT measurement at these packet sizes be

$$T_1 = \alpha + L_1\beta + \frac{q_1}{C} \tag{16}$$

$$T_2 = \alpha + L_2\beta + \frac{q_2}{C} \tag{17}$$

where q_1 and q_2 are the minimum queue sizes seen by probing packets of size L_1 and L_2, respectively. The RTT slope that we would measure then, is

$$\frac{\Delta T}{\Delta L} = \frac{T_2 - T_1}{L_2 - L_1} = \beta + \frac{\Delta q}{C\Delta L} \tag{18}$$

where $\beta = 1/C$, $\Delta T = T_2 - T_1$, $\Delta L = L_2 - L_1$, and $\Delta q = q_2 - q_1$. So, the estimated capacity will be

$$\hat{C} = \frac{\Delta L}{\Delta T} = \frac{C}{\left(1 + \frac{\Delta q}{\Delta L}\right)} \tag{19}$$

Therefore, non-zero queueing delays cause a multiplicative error term in the capacity estimate. This error factor can be reduced with a larger packet size variation ΔL; usually however, ΔL is limited by the 1500 byte Ethernet MTU. It is important to note that all VPS tools measure the RTT slope at a link using tens of different packet sizes and elaborate linear regression algorithms. Consequently, VPS tools are more robust to queueing delays than what the previous model implies.

C. Effect of limited clock resolution

The accuracy of RTT measurements is also limited by the resolution of the clock at the measurement host. If the clock resolution is 2σ, any time instant between $t_0-\sigma$ and $t_0+\sigma$ will be measured as t_0. The minimum RTT measurement for two

[7]The transmission of probing packets should be of course rate-limited to avoid 'self-queueing'.

packet sizes L_1 and L_2 can then be anywhere in the following ranges

$$T_1 = \alpha + L_1\beta \pm \sigma \quad (20)$$

$$T_2 = \alpha + L_2\beta \pm \sigma \quad (21)$$

In the worst-case, the RTT slope can be estimated as

$$\frac{\Delta T}{\Delta L} = \beta \pm \frac{2\sigma}{\Delta L} \quad (22)$$

The estimated capacity would then be

$$\hat{C} = \frac{C}{1 \pm \frac{2\sigma C}{\Delta L}} \quad (23)$$

Table VIII shows the range of capacity estimates that can result at high-bandwidth links, when the clock resolution is 2σ=1μsec, and the packet size variation is limited by the Ethernet MTU ($\Delta L \approx$1500 bytes). Such a high resolution is typical for workstations today[8]. Note that with these values of σ and ΔL, it is basically futile to accurately measure OC-48 or higher bandwidth links with a VPS tool.

D. Error propagation from previous links

In VPS, the capacity estimate for a link i depends on the RTT slope that was measured at link $i-1$ (see Equation 6). To examine how estimation errors can propagate along a path, consider a two-hop path with C_1 and C_2 being the capacities of the two links. Ideally, the RTT slopes should be measured as $\beta_1 = 1/C_1$ at the first link, and $\beta_2 = 1/C_1 + 1/C_2$ at the first two links. Suppose now that the first link introduces an error, bounded by ϵ, in the RTT slope measurements. Then, even if there is no error introduced at the second link, the two RTT slopes would be measured as

$$\hat{\beta}_1 = \beta_1(1+\epsilon_1), \hat{\beta}_2 = \beta_1(1+\epsilon_2) + \beta_2 \quad (24)$$

where ϵ_1 and ϵ_2 are the errors introduced by the first link ($|\epsilon_1|,\ |\epsilon_2| \leq \epsilon$). The capacity of the second link will be estimated as

$$\hat{C}_2 = \frac{1}{\hat{\beta}_2 - \hat{\beta}_1} = \frac{C_2}{1 + (\epsilon_2 - \epsilon_1)\frac{C_2}{C_1}} \quad (25)$$

The estimation error is maximized when $\epsilon_2 = -\epsilon_1 = \epsilon$. Then,

$$\hat{C}_2 = \frac{C_2}{1 + 2\epsilon C_2/C_1} \quad (26)$$

The previous expression shows that if the capacity ratio C_2/C_1 of two successive links is larger than one, any estimation error at the first link is "magnified" by that capacity ratio at the second link. For example, if an Ethernet link is followed by a Gigabit Ethernet link (C_2/C_1=100), an error factor of 0.1% at the first link can result in a 10% error in the capacity estimate of the second link.

[8]A resolution of just a few microseconds is often based on interpolation between successive clock interrupts, meaning that it may not be exact [29].

E. Effect of the ICMP messages generation latency

It is a conventional wisdom that VPS tools are sometimes inaccurate because router ICMP replies, which are required for the RTT measurements, are generated from "slow processing and forwarding paths". First of all, this may not even be the case, given the recent measurements of [30]. Even if it is true for some routers, however, it should be clear from §III that it is not the latency of generating ICMP replies that can affect the RTT slope measurements. Instead, it is the *variation of those latencies* that can affect the measured RTT slope. In other words, ICMP messages will not affect the accuracy of VPS probing *as long as the minimum ICMP reply generation latency $\tilde{T}_{ICMP}$ at a router does not vary with the size L of the packet that caused the ICMP reply*. If this is the case, the impact of ICMP packet generation on VPS probing is no different than that of queueing delays: a sufficiently large number of probing packets is required, so that the minimum RTT measurement corresponds to an ICMP reply with the minimum generation latency $\tilde{T}_{ICMP}$.

VIII. Related work and conclusions

This paper examined different sources of error in VPS probing. We focused on the effect of L2 store-and-forward devices, which introduce serialization latencies just like L3 routers, but without decrementing the TTL field of probing packets. It has been shown that such L2 devices introduce significant and consistent estimation errors, which are impossible to detect unless the L2 path infrastructure is known. Some other sources of error, such as queueing delays, limited clock resolution, propagation of errors from previous hops, have been also examined. An important point is that these latter effects are probabilistic. So, when the effects of these error sources are large enough, repeated runs of a VPS tool over the same path will produce significantly different capacity estimates. In this sense, probabilistic error sources in capacity estimation are easier to detect.

A different technique to measure the capacity of each L3 hop would be to send pairs (or trains) of probing packets with an expiring TTL, and then measure the resulting dispersion at each hop. This technique, which is similar to what *pipechar* does [25], is not affected by the presence of L2 devices in the path. The problem with this technique, however, is that it cannot measure the capacity C of a link if there is a previous link in the path with capacity $C' < C$.

Recently, [31] presented five new methodologies for estimating per-hop capacities, using sequences of four packets called *packet quartets*. A packet quartet consists of two independent packet-pairs, with the first packet of each pair, called *pacesetter*, followed by a much smaller packet called *probe*. The pacesetter's TTL is set to expire at a particular hop in the path. The proposed methodologies are based on the end-to-end delay variation of the two probe packets. Two of those methods are referred to as PQ1 and PQ2. In both PQ1 and PQ2, the pacesetters expire at the same hop. In PQ1 the probes have the same size, while in PQ2 the pacesetters have the same size. The probe packet delay variations in PQ1 are up

FastEthernet	OC-3	OC-12	Gig-E	OC-48	OC-196	10Gig-E
99.2-100.9M	153.0-157.0M	589.6-653.7M	0.92-1.09G	2.0-3.1G	5.4-57.2G	5.4-66.7G

TABLE VIII

RANGE OF CAPACITY ESTIMATES WHEN THE CLOCK RESOLUTION IS ONE MICROSECOND (ΔL=1500B).

to the last hop before the pacesetters expire, while the delay variations in PQ2 are from that hop till the receiving end of the path.

An important contribution of [31] is that it proposes a way to detect the presence of layer-2 switches in a path (but not to estimate the capacity of each layer-2 segment in a hop). Specifically, if the PQ1 and PQ2 methods result in different estimates for a particular layer-3 hop, then that hop includes layer-2 switches. It is important to note however that PQ2 can be error-prone. The reason is that for PQ2 to be accurate the two probes must have significantly different packet sizes. The probes however need to be significantly smaller that the pacesetters, and the latter are limited by the path MTU. As shown in the Table II of [31], the PQ2 capacity estimate is only 20Mbps for a 100Mbps hop (without layer-2 segments), when that hop is 30% utilized. It should be thus expected that PQ2 may result in a different estimate than PQ1, even if the corresponding hop does not include layer-2 switches.

We finally note that L2 devices may have a similar negative effect in other areas of network measurement. For instance, several recent studies use *traceroute* to create Internet maps or to study the topological and performance properties of Internet paths [32], [33]. It should be clear, however, that *traceroute* cannot detect L2 devices, while sometimes it is the L2 devices that determine at large the geographical or logical topology of a network. Additionally, per-hop delay and loss rate measurement studies should be aware that packets can be delayed or lost not only in routers, but also in the L2 devices of a network path.

REFERENCES

[1] R. S. Prasad, C. Dovrolis, and B. A. Mah, "The Effect of Layer-2 Switches on Pathchar-like Tools," in *Proceedings Internet Measurement Workshop (IMW) (short abstract)*, 2002.

[2] V. Jacobson, "Pathchar: A Tool to Infer Characteristics of Internet Paths," ftp://ftp.ee.lbl.gov/pathchar/, Apr. 1997.

[3] S. Bellovin, "A Best-Case Network Performance Model," ATT Research, Tech. Rep., Feb. 1992.

[4] A. Downey, "Using Pathchar to Estimate Internet Link Characteristics," in *Proceedings of ACM SIGCOMM*, Sept. 1999, pp. 241–294.

[5] B. A. Mah, "pchar: a Tool for Measuring Internet Path Characteristics," http://www.employees.org/∼bmah/Software/pchar/, Feb. 1999.

[6] K. Lai and M. Baker, "Measuring Link Bandwidths Using a Deterministic Model of Packet Delay," in *Proceedings of ACM SIGCOMM*, Sept. 2000, pp. 283–294.

[7] J. Postel, *Internet Control Message Protocol*, Sept. 1981, IETF RFC 792.

[8] V. Jacobson, "Traceroute: A Tool to Show The Route Packets Take to a Network Host," ftp://ftp.ee.lbl.gov/.

[9] M. Alves, L. Corsello, D. Karrenberg, C. Ogut, M. Santcroos, R. Sojka, H. Uijterwaak, and R. Wilhelm, "New Measurements with the RIPE NCC Test Traffic Measurements Setup," in *Proceedings of Passive and Active Measurements (PAM) workshop*, Mar. 2002, pp. 66–75.

[10] J. C. Bolot, "Characterizing End-to-End Packet Delay and Loss in the Internet," in *Proceedings of ACM SIGCOMM*, 1993, pp. 289–298.

[11] K. Lai and M. Baker, "Measuring Bandwidth," in *Proceedings of IEEE INFOCOM*, Apr. 1999, pp. 235–245.

[12] V. Paxson, "End-to-End Internet Packet Dynamics," *IEEE/ACM Transaction on Networking*, vol. 7, no. 3, pp. 277–292, June 1999.

[13] B. Melander, M. Bjorkman, and P. Gunningberg, "A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks," in *Global Internet Symposium*, 2000.

[14] C. Dovrolis, P. Ramanathan, and D. Moore, "What do Packet Dispersion Techniques Measure?" in *Proceedings of IEEE INFOCOM*, Apr. 2001, pp. 905–914.

[15] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput," in *Proceedings of ACM SIGCOMM*, Aug. 2002.

[16] R. L. Carter and M. E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," *Performance Evaluation*, vol. 27,28, pp. 297–318, 1996.

[17] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk, "Multifractal Cross-Traffic Estimation," in *Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling, and Management*, Sept. 2000.

[18] G. Jin, G. Yang, B. Crowley, and D. Agarwal, "Network Characterization Service (NCS)," in *Proceedings of 10th IEEE Symposium on High Performance Distributed Computing*, Aug. 2001.

[19] M. Jain and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," in *Proceedings of Passive and Active Measurements (PAM) Workshop*, Mar. 2002, pp. 14–25.

[20] M. Mathis and M. Allman, *A Framework for Defining Empirical Bulk Transfer Capacity Metrics*, July 2001, RFC 3148.

[21] M. Allman, "Measuring End-to-End Bulk Transfer Capacity," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001, pp. 139–144.

[22] NLANR Distributed Applications Support Team, "Iperf Version 1.2," http://dast.nlanr.net/Projects/Iperf/, May 2001.

[23] S. Saroiu, "Sprobe: A Fast Tool for Measuring Bottleneck Bandwidth in Uncooperative Environments," http://www.cs.washington.edu/homes/tzoompy/sprobe/, Sept. 2001.

[24] A. Downey, "clink: a Tool for Estimating Internet Link Characteristics," http://rocky.wellesley.edu/downey/clink/, June 1999.

[25] J. Guojun, "Network Characterization Service," http://www-didc.lbl.gov/pipechar/, July 2001.

[26] K. Lai, "Effect of L2 Devices on Nettimer," Personal Correspondence, July 2002.

[27] J. Kurose and K. Ross, *Computer Networking: a Top-Down Approach Featuring the Internet*. Addison-Wesley, 2001.

[28] R. Wolff, "Poisson Arrivals See Time Averages," *Operations Research*, vol. 30, no. 2, pp. 223–231, 1982.

[29] A. Pásztor and D. Veitch, "A Precision Infrastructure for Active Probing," in *Proceedings of Passive and Active Measurements (PAM) workshop*, 2001.

[30] R. Govindan and V. Paxson, "Estimating Router ICMP Generation Delays," in *Proceedings of Passive and Active Measurements (PAM) workshop*, Mar. 2002, pp. 6–13.

[31] A. Pásztor and D. Veitch, "Active probing using packet quartets," in *Proceedings of Internet Measrement Workshop (IMW)*, Nov. 2002, pp. 293–306.

[32] R. Govindan and H. Tangmunarunkit, "Heuristics for Internet Map Discovery," in *Proceedings of IEEE INFOCOM*, Mar. 2000, pp. 1371–1380.

[33] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The End-to-End Effects of Internet Path Selection," in *Proceedings of ACM SIGCOMM*, Sept. 1999, pp. 289–300.