Passive Capacity Estimation: Comparison of Existing Tools

Taoufik En-Najjary and Guillaume Urvoy-Keller
Institut Eurecom
Sophia–Antipolis, France
ennajjar, urvoy@eurecom.fr

Abstract

In this paper, we focus on the issue of passive capacity estimation. We consider all the existing passive tools devised so far, namely PPrate, Nettimer and MultiQ. We first compare those passive tools to one state-of-the-art active tool, Pathrate, using Planetlab. We next investigate the performance of the passive tools in a more realistic and challenging environment, namely an ADSL platform. Our study reveals that Nettimer is unable to work properly on TCP ack streams and tends to underestimate path capacities when working on TCP data streams. In contrast, MultiQ works on both TCP ack and TCP data streams but tends to overestimate path capacities. Overall, PPrate offers a good compromise in most situations.

Keywords: Passive measurement, Network capacity, Packet-pair dispersion

1. INTRODUCTION

Measuring path capacity in the Internet is a problem that has received a significant attention from the research community [7, 12, 3, 4, 1, 14, 10, 8]. The capacity of a path is formally defined as the maximum IP-layer throughput that a flow can get on that path and is determined by the link with the minimum capacity among all links on a path.

Existing capacity estimation tools can be classified in three categories. First, active tools that rely on the injection of measurement packets in the network. Most existing tools are active, including Pathchar [7], Pathrate [3] or Capprobe [8]. The second category consists of so-called embedded tools that alter the pattern of users traffic to estimate capacity without introducing additional measurement traffic. A typical example is TFRC Probe [2], which is an embedded version of CapProbe [8]. The third category consists of purely passive tools that aim at extracting capacity estimates from passively collected traces of traffic. To the best of our knowledge, three passive capacity estimation tools have been proposed so far: PPrate [5], Nettimer [10] and MultiQ [9]. MutliQ is an element of the click modular router[1], while PPrate is integrated in the Intrabase analysis tool[2]. PPrate, Nettimer and MultiQ are based on packet dispersion techniques and work with TCP connections. The reason why they focus on TCP traffic is twofold. First, TCP has been adopted by the majority of new applications, e.g., p2p file sharing or podcasting and remains the most popular transport protocol in the Internet today. Second, TCP is a natural candidate for capacity measurement techniques based on the packet dispersion principle since TCP often injects packets in pairs in the network, due to the delayed acknowledgment strategy.

A "good" passive capacity estimation tool should be able to extract the capacity of a path regardless of the measurement point or the connection direction, data stream or ack stream. The latter (ack stream) is challenging as information on the packet dispersion of the data stream is partly hidden by the delayed acknowledgement strategy.

The amount of samples necessary to estimate the capacity is also a challenge for passive tools. If applied on traces collected in a link located in the core of the Internet, there is a high chance that capacity information can be extracted for only a small fraction of the connections since most of them are in general small - say less than 20 packets - in the Internet. On the other hand, passive tools can be extremely valuable for an ADSL provider that wants to measure the access link capacity of its customers (which differs from their subscription rate). In such a context, there is a high chance that if the capture lasts long enough, one large enough sample will be available for every customer. Note in addition that the use of active tools in ADSL environments can be extremely intrusive due to the low capacities of the downlinks and even more of the uplinks.

In this work, we aim at addressing the following questions:

- **Are all passive tools able to work both on ack and data streams?** No. It turns out that only MultiQ and PPrate are able to handle properly ack streams while Nettimer works only for data streams.
- **How accurate are passive tools as compared to active tools?** We compared passive tools to one state-of-the-art active tool, namely Pathrate, on Planetlab, and we observed that the results returned by passive tools are very close in general to the ones of Pathrate. However, the number of packets needed to obtain a good estimate is different for the three tools. PPrate is able to work with around 300 packets, which is equivalent to about 450 kbytes (assuming MSS packets of 1500 bytes). Nettimer and even more MultiQ require more packets to offer satisfactory results.

[1] http://www.read.cs.ucla.edu/click/
[2] http://intrabase.eurecom.fr

- **Are experiments on Planetlab enough to evaluate and compare passive tools?** No. We did observe that on Planetlab all three tools behave similarly. However, results are clearly different when using ADSL traces as we do in Section 3. We observed when working with ADSL traces that Nettimer almost consistently underestimates path capacity. On the other hand, MultiQ tends to often overestimate the capacity. PPrate offers a good compromise in general but underestimates path capacity when working on ack streams.

2. PLANETLAB EXPERIMENTS

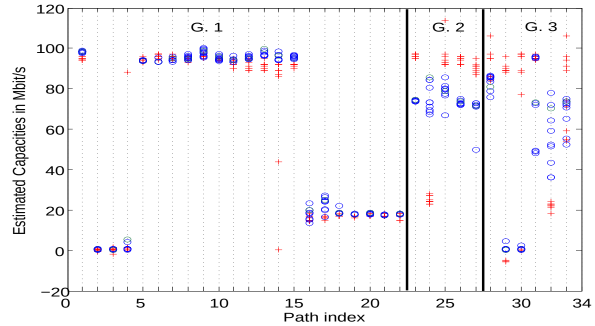
In this section, we compare PPrate, Nettimer and MultiQ to one active tool, Pathrate. We used Planetlab, which enables us to control the two ends of a path; a mandatory condition to run an active tool like Pathrate. Our focus is on receiver side experiments, i.e., we capture the data stream at the receiver side.

We selected 33 paths between randomly chosen PlanetLab nodes. On each path, we first perform one *scp* transfer (of a 20 Mbytes file), and simultaneously dump the data packets stream at the receiver side. Note that it is not possible to collect traces at both end points of a path on PlanetLab. Indeed, Tcpdump can trace only the connections that have been initiated by the host on which it runs. Immediately after the *scp* transfer, we run Pathrate. The overall procedure lasts a few hours for each path, as it takes 15 to 30 minutes for Pathrate to return an estimate and 1 to 5 minutes for each *scp* transfer. For each transfer, we further collected the capacity estimates computed using PPrate, MultiQ and Nettimer.

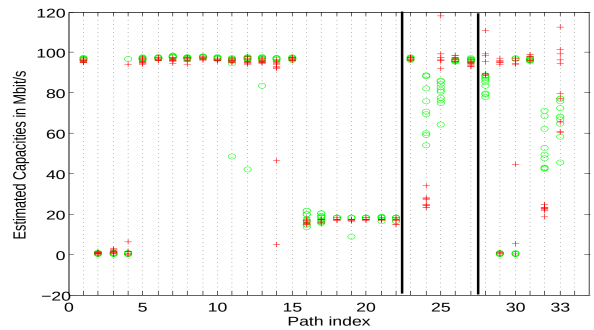
Figures 1(a) to 1(c) summarize the results for the 33 paths that we considered. A given index on the x axis corresponds to a single path. For each figure we use crosses for Pathrate and circles for the considered passive tool.

To ease interpretation of the results, we have formed 3 groups of paths labeled as Group 1, 2 and 3. The grouping stems from the comparison between Pathrate and PPrate in Figure 1(a) and we use the same grouping for Figures 1(b) and 1(c). Group 1 contains the paths for which Pathrate and PPrate return consistent estimates (i.e., each tool returns approximately the same values for the 10 experiments) and, in addition, the two tools agree with each other. There are 22 paths, that is two thirds of the paths, in Group 1. Group 2 consists of paths for which each tool returns consistent estimates but the two tools do not agree with one another, which the case of relatively small number (4). Group 3 consists of seven paths for which one or both tools return inconsistent estimates.

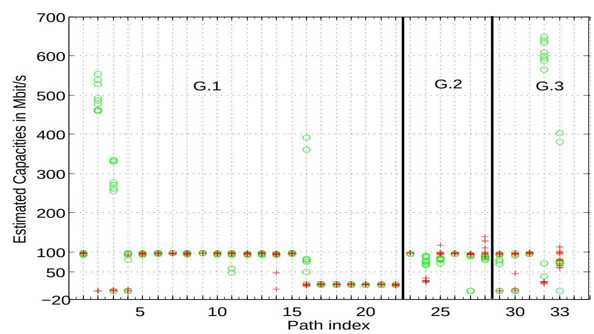
Overall, we observe from Figures 1(b) and 1(c), that for each group, PPrate, Nettimer and MultiQ behave similarly to Pathrate. Note that for the case of MultiQ the scale on the y-axis goes up to 800 Mbits/s, as we wanted to emphasize the



(a) Pathrate(+) vs PPrate(o)



(b) Pathrate(+) vs Nettimer(o)



(c) Pathrate(+) vs MultiQ(o)

Figure 1. PlanetLab Experiments

tendency of MultiQ to return, from time to time, large values that are, with high probability, overestimates. Note however that if we rescale Figure 1(c) to a maximum of 140 Mbits/s like the two others, all three figures become qualitatively similar. All tools can exhibit a high variance in their estimates for the paths in Group 3. It is difficult to comment on the paths in this group as many factors might explain why the tools do not work properly. A possible reason for these high variances is the use of slices and rate limitations on Planetlab nodes[3]. We note that previous studies of Pathrate over Planetlab nodes have also observed a number of problems [11].

The main conclusion that we draw from those experiments

[3] http://www.planet-lab.org/doc/BandwidthLimits.php

is that for a significant number of cases, the three passive tools fully or partially agree with our reference tool, namely Pathrate.

2.1. Consistency

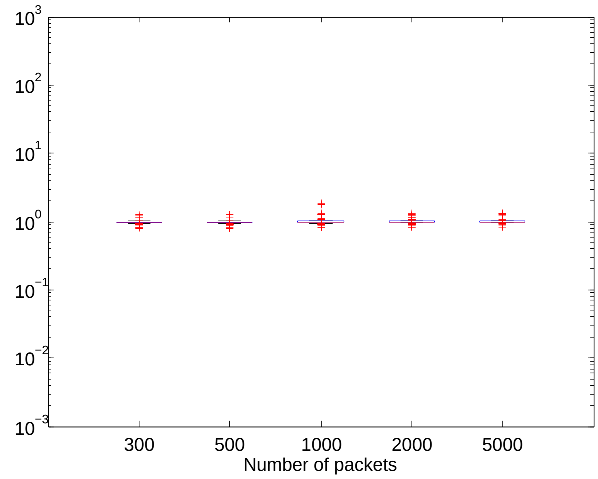
The number of samples required to perform a correct estimation of the capacity of a path is a crucial point for passive tools as they are applied on traces for which it is not possible to control the duration of a transfer. As we do not know the exact value of the path capacities on Planetlab, we will address this question in an indirect manner by focusing on the consistency of each tool.

We proceed as follows. We consider the 330 *scp* transfers that we did (10 experiments per path with a total of 33 paths) and applied, for each transfer, PPrate, Nettimer and MultiQ on the first 300, 500, 1000, 2000 or 5000 first packets of the transfer (each transfer contains about 12000 packets). We next compute for each case the ratio between the estimate obtained with the first x packets and the estimate obtained using all packets. The consistency of a tool will be measured as the fraction of ratios close to 1.

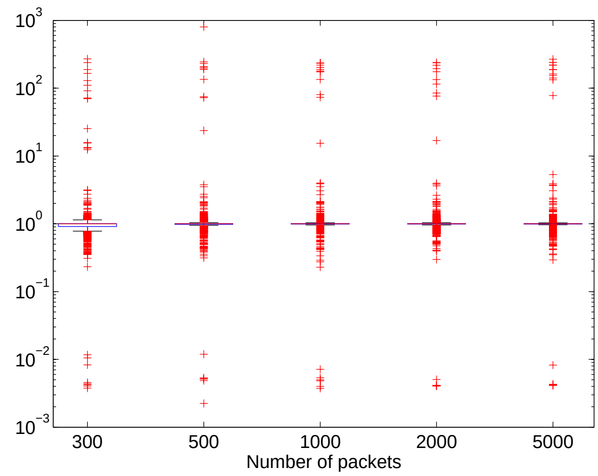
We plot in Figures 2(a) to 2(c) boxplots[4] of the estimations for the different number of packets used and for all the transfers on all the paths.

We first observe from Figures 2(a) to 2(c) that the boxplots for each tool have their upper and lower edges close to 1, meaning that most of the time, the estimate obtained by a given tool with $x < 12000$ samples is consistent with the one obtained with 12000 samples. Extreme values, i.e., values that are far from the core of the distribution, are marked with a cross. With respect to those extreme values, we observe that Nettimer and MultiQ are more prone to generate large deviations (i.e., large under or over estimations) than PPrate. Indeed, Nettimer and MultiQ exhibit large outliers even when using 5000 samples, as compared to PPrate. Identifying the reason behind those observations is not straightforward. The three tools rely on the histogram of packet pairs dispersion to infer the capacity of the path. However, they differ in the filtering technique they use to uncover the capacity mode. Nettimer and MultiQ both rely on kernel density-based technique to filter the dispersion histogram. PPrate uses a lower bound estimate of the capacity to guide the search for the capacity mode, after a preliminary phase where outliers due either to the application on top or to measurement errors (context switching in the OS) are removed. It is difficult to assess a priori the relative merits of both approaches. We conjecture

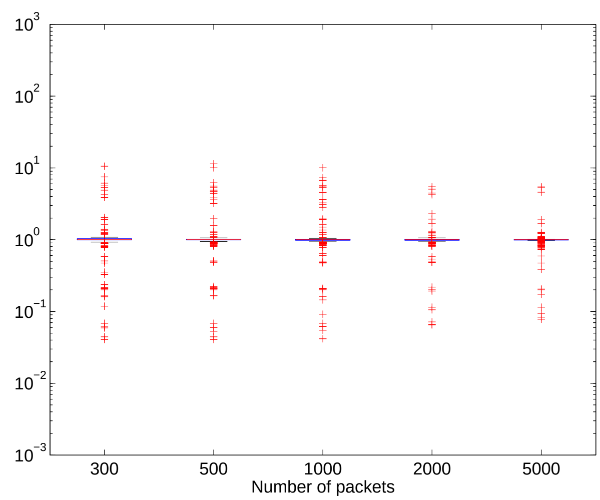
[4] A boxplot is a graphical representation of a distribution where the upper line of the box is the 75[th] percentile p_{75} of the distribution, the lower line is the 25[th] percentile p_{25} and the central line is the median of the distribution. In addition, two lines are added on each side of the box at $+/-1.5IQR$. 50% of the mass of the distribution in the box while, intuitively, one expects most of the samples to lie between the two outer lines. Values outside those boundaries are marked with a cross as they are extreme values of the distribution.



(a) PPrate



(b) Nettimer



(c) MultiQ

Figure 2. Boxplots of ratio between the estimates obtained with x packets and the ones obtained with the full FTP transfer

that the extreme values generated by Nettimer and MultiQ might stem from the kernel density estimation technique used in Nettimer and MultiQ that, in general, require a quite large number of samples to rip the full benefit of the method.

3. ANALYSIS OF ADSL TRACES

In this section, we further compare PPrate, Nettimer and MultiQ on traces from an ADSL access network[5] made available by the University of Twente. We focus on two specific traces. The first trace consists of FTP connections while the second trace consists of HTTP connections. We use only connections with more than 300 packets, which represent 20% of the observed *ack* streams for the FTP trace and 8% of the *ack* streams for the HTTP trace. The majority of these connections are large (more than 1000 packets), the FTP connections being globally larger than the HTTP connections. All traffic was collected by a probe close to the ADSL clients. The only information we have on the clients is that their access link capacities range from 256 kbits/s to 8 Mbits/s. By working on the ack stream collected by the probe, passive capacity estimation tools should be able to estimate the downlink capacity of the clients, as the latter should constitute the bottleneck of the path (Section 3.2.). Conversely, working on the data streams allows us to infer the upstream capacity, i.e. the capacity of the portion of the path between the sender and the measurement point. We will present results only for the HTTP trace. Results on the FTP trace (which are qualitatively similar) are available in our technical report [6].

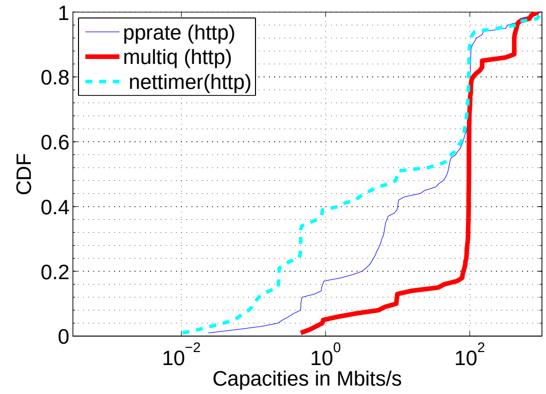
3.1. TCP data streams

We applied PPrate, Nettimer and MultiQ on the data stream of HTTP connections to estimate the upstream capacity. Given the low utilization often observed in core networks, the upstream capacity should often equal the capacity of the server in client server applications like HTTP.

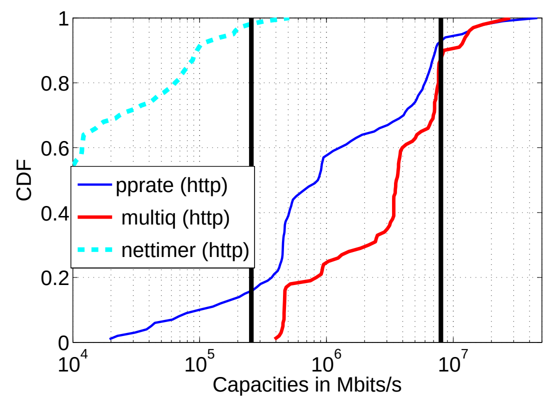
Figure 3(a) shows the distribution of the capacity estimates for HTTP connections. We observe peaks at 10 and 100 Mbits/s for the three tools, in accordance with the intuition that HTTP servers have an Ethernet or FastEthernet network interface.

MultiQ returns an estimate of 100 Mbit/s for 75% of the connections, while PPrate returns 100 Mbit/s for only 25% of the connections. To understand this phenomenon, we plot in Figure 4(a) a scatter plot of the results of PPrate and MultiQ. To ease interpretation, we highlight four regions in Figure 4(a) : region 1 where MultiQ return values larger than 200 Mbit/s, which represent 14.3% of the connections; region 2 where both tool return values around 100 Mbit/s; region 3 where PParte returns values around 10 Mbit/s and MultiQ returns values around 100 Mbit/s and region 4 where PPrate

[5]M2C Measurement Data Repository: http://m2c-a.cs.utwente.nl/



(a) Data streams - HTTP traffic



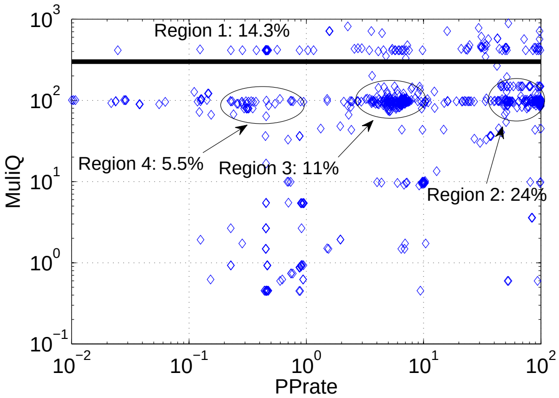
(b) Ack streams - HTTP traffic

Figure 3. Distributions of the capacity estimates for HTTP traces

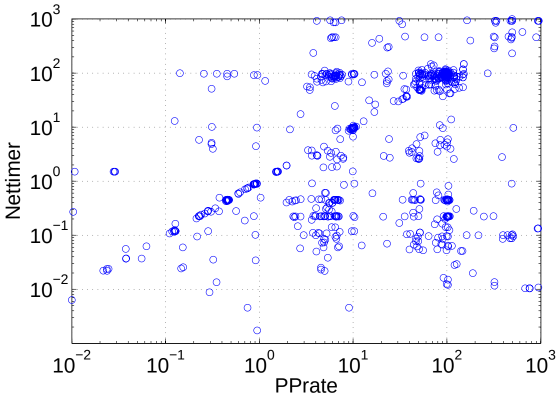
returns values around 1 Mbit/s and MultiQ returns values around 100 Mbit/s. Region 1 corresponds, with a high probability, to cases where MultiQ overestimates path capacities. Region 2, which represents about 24% of the cases, corresponds to cases where both PPrate and MultiQ agree with each other. Regions 3 and 4, which sum around 16.5% of the samples, correspond to cases where MultiQ returns capacity values that are one or two orders of magnitudes larger than the ones of PPrate. To understand the difference between PPrate and MultiQ, we plot the capacity histograms for connections in regions 1, 3 and 4.

Figure 5(a) shows an example of capacity histogram in the first region. In this example, PPrate returns a capacity of 50 Mbit/s while MultiQ returns 400Mbit/s. Although we do not know the exact value of the capacity, 50Mbit/s is a reasonable value for an HTTP Server. However, it is clear that MultiQ fails to estimate the capacity in this case as there no capacity samples larger than 150 Mbit/s in this example.

Figure 5(b) shows an example of capacity histogram in region 3. In this example, MultiQ elects the second peak as the



(a) PPrate vs MultiQ



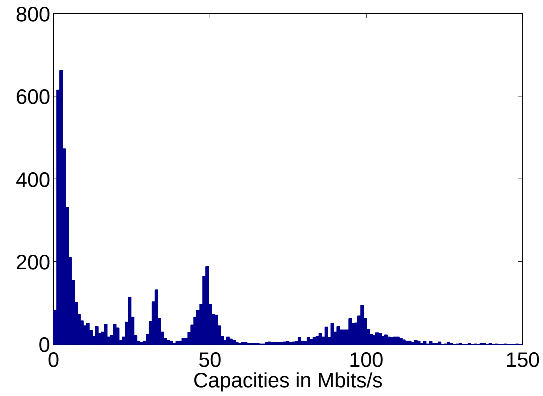
(b) PPrate vs Nettimer

Figure 4. Scatter plots for HTTP connections

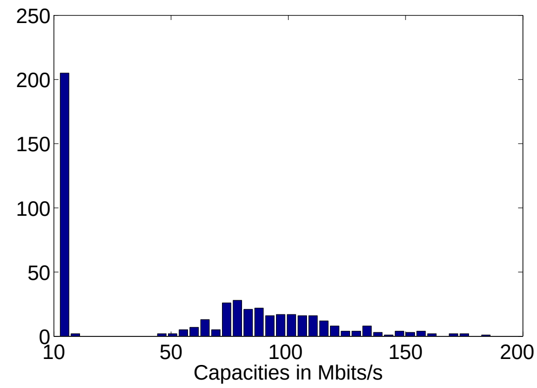
capacity peak, while PPrate chooses the first peak. PPrate excludes the peak around 100 Mbit/s as its size is an order of magnitude smaller than the dominant peak. It is difficult to know which tool is right. Note however that the first peak is so narrow that it is not unlikely that the second peak be due to compression of some of the packets at a 100 Mbits/s link along the path.

Figure 5(c) shows an example of capacity histograms in region 4. This case is similar to the one depicted in Figure 5(a): there is no mass at the value returned by MultiQ. It is difficult to diagnose the cause behind this strange behavior of MultiQ. Note that it can not be any obvious programming error as one fails to identify a consistent behavior, e.g, a consistent overestimation, of MultiQ. Indeed, there are a number of cases, e.g. region 2 of Figure 4(a) for the HTTP traffic, where MultiQ agrees with PPrate and both tools return reasonable results.

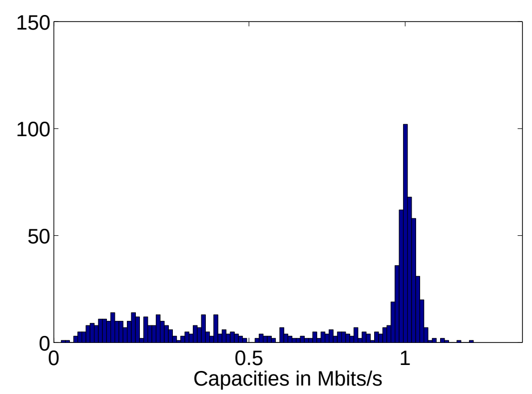
Let us now focus on Nettimer vs PPrate. We first note the tendency of Nettimer to underestimate the capacity. Indeed, about 50% of the estimates are less than 1Mbits/s which are far from typical values for HTTP servers. The scatterplot of Figure 4(b) indicates that the major difference between PPrate and Nettimer is that in a number of cases, PPrate estimates capacity values around 10 or 100 Mbit/s while Nettimer returns values smaller than 1 Mbit/s. Inspection of specific histograms (as for the case of MultiQ) reveals that Nettimer often elects a minor peak as the peak of the capacity.



(a) PPrate ∼50 Mbits/s, MultiQ ∼400 Mbits/s (No sample larger than 150 Mbits/s)



(b) PPrate ∼10 Mbits/s, MultiQ ∼100 Mbits/s



(c) PPrate ∼1 Mbits/s, MultiQ ∼100 Mbits/s (No sample larger than 1.5 Mbits/s)

Figure 5. Examples of Capacity Histograms

Overall our conclusions for the comparison of the three tools when applied on the data streams of HTTP (and also FTP - see [6]) trace are:

- Nettimer apparently often underestimate paths capacities.
- In contrast, MultiQ sometimes overestimates path capacities. This is in line with our observions in Section 2..
- There is a number of cases where determining which tool is actually right is difficult, if not impossible.
- Experiments on Planetlab are not suffisient to assess the accuracy of measurement tools.

3.2. TCP ack streams

First note that the vast majority of HTTP servers observed in our traces should not to be hosted by the ADSL clients we observe. As a consequence, *ack* streams of the FTP and HTTP traces should reveal the bottleneck of the path from servers in the Internet to the ADSL clients, which should be with high probability the downlink of the clients[6].

Figure 3(b) represents the cumulative distribution functions of capacity estimates of the 3 tools for the two types of traffic. The two vertical bars mark the lower bound (256 kbits/s) and the upper bound (8 Mbits/s) on the clients download capacities.

We observe from Figure 3(b) that Nettimer is way off as compared to the two other tools, as over 75% of the values returned by Nettimer for the FTP and HTTP traces are below the lower bound of 256 kbits/s. Note however that Nettimer was not optimized to work on ack streams, and all the results in [10] were obtained for data streams.

We also observe that approximately 70% (resp. 85%) of the estimated capacities fall in the interval [256kbits/s, 8Mbits/s] with PPrate (resp. with MultiQ). For both tools, we do observe peaks around values close to 500 kbits/s, 1 Mbits/s and 8 Mbits/s, which are typical capacities of ADSL clients. The reason why we do not observe mass only at those characteristic values might be due to the fact that the actual ADSL capacity decreases with increasing distance of the phone line between the customer premise and the ADSL concentrator. We further observe that the minimum value returned by MultiQ is 400 kbits/s while PPrate apparently underestimates the capacity of the path for around 20% of the cases. Note that when MultiQ is estimating a capacity it makes use of a table containing characteristic capacity values (Ethernet, Fast Ethernet, etc.). This might explain the minimum value we observe here for MultiQ.

[6]We assume here that the uplink of the clients should not slow down ack streams. It is however possible that the uplink constitutes the bottleneck if, e.g., a p2p application is saturating the link. We however note that p2p applications often use rate limiters to avoid those situations. A recent study on ADSL users has observed that utilization of the uplinks are in general low [15].

As for the discrepency between the cdfs of PPrate and MutliQ in the interval between 256 kbits/s and 8 Mbits/s (remember that as we estimate the download capacity of the users, they should be similar),we applied the same technique as we did in the previous section to try to understand why MultiQ returns larger values than PPrate. Observation of some specific histograms revealed again that MultiQ sometimes sees mass where it is absent in the capacity histogram.

4. CONCLUSION

In this paper, we have compared the three existing passive capacity estimation tools with publicly available implementations we are aware of, namely PPrate, MultiQ and Nettimer. To the best of our knowledge, this is the first time that such a comparision is conducted. Our main conclusions are that (i) Nettimer, which is the oldest tool, performs badly with ADSL traces (even if it provides accurate results on Planetlab) and is not able to handle TCP ack streams; (ii) MultiQ tends to have quite erratic behaviors from time to time where it grossly overestimates the path capacity and consequently, is less consistent than the other tools; and(iii) PPrate often returns reasonnable results even if it can underestimate the path capacity when working on TCP ack streams.

This study also highlights the need as well as the difficulty to compare tools in a variety of situation and not only in a controlled environement like Planetlab. For the case of traces collected out in the wild and for which the ground truth is not available, a visual inspection of the capacity histogram can help to determine in most situations which tool is more likely to return the correct value. A straightforwad direction for future work would be to derive a number of heuristics to be applied automatically on capacity histograms to rate the choice of each passive tool.

ACKNOWLEDGEMENT

This work has been partly supported by France Telecom, project CRE-46126878.

REFERENCES

[1] R. Carter and M. Crovella, “Dynamic Server Selection using Bandwidth Probing in Wide-Area Networks”, BU-CS-007, Boston University, 1996.

[2] L.-J. L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla, “Monitoring access link capacity using TFRC probe”, *Computer Communications*, 29(10):1605–1613, 2006.

[3] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion techniques and a capacity-estimation methodology", *IEEE/ACM Trans. Netw.*, 12(6):963–977, 2004.

[4] A. Downey, "Clink: A tool for Esimating Internet Link Characteristics", *http://rocky.wellesley.edu/doweney/clink/*, 1999.

[5] T. En-Najjary and G. Urvoy-Keller, "PPrate: A Passive Capacity Estimation", In *IEEE e2emon*, Vancouver, CANADA, 2006.

[6] T. En-Najjary and G. Urvoy-Keller, "Passive capacity estimation: Comparison of existing tools", *Technical Report RR-07-204*, October 15th, 2007.

[7] V. Jacobson, "Pathchar", http://www.caida.org/Pathchar/ Source: ftp://ftp.ee.lbl.gov/pa thchar.

[8] R. Kapoor, L. Chen, L. Lao, M. Gerla, and M. Sanadidi, "CapProbe: A Simple and Accurate Capacity Estimation Technique", *In Proceeding ACM SIGCOMM*, 2004.

[9] S. Katti, D. Katabi, C. Blake, E. Kohler, and J. Strauss, "MultiQ: automated detection of multiple bottleneck capacities along a path", In *IMC '04*, pp. 245–250, 2004.

[10] K. Lai and M. Baker, "Nettimer: A Tool for Measuring Bottleneck Link Bandwidth", *In Proceeding of USENIX*, 1999.

[11] S.-J. Lee, P. Sharma, S. Banerjee, S. Basu, and R. Fonseca, "Measuring Bandwidth between PlanetLab Nodes", In *Passive and Active Measurements 2005*, March 2005.

[12] B. Mah, "pchar: A tool for measuring Internet Paths Characteristics", *http://www.employees.org/bmah/Software/pchar/*, 2000.

[13] V. Paxson, *Measurements and analysis of End-to-End Internet Dynamics*, Ph.D. Thesis, University of California, Berkeley, April 1997.

[14] S. Saroiu, P. Gummadi, and S. Gribble, "Sprobe: A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments", *http://sprobe.cs.washinton.edu*, 2002.

[15] M. Siekkinen, D. Collange, G. Urvoy-Keller, and E. W. Biersack, "Performance Limitations of ADSL Users: A Case Study", In *Proc. Passive and Active Measurement: PAM 2007*, April 2007.