International Journal of Research
Available at https://edupediapublications.org/journals
Special Issue on Conference Papers
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 06
March 2018

Capacity Bandwidth Measurement of Targeted Path Segments

[1]*Dommati.Srikanth,* [2] *Pochampalli.Soundarya*

Assistant professor, Dept. Electronics & Communication Engineering
Mother Theressa College Of Engg & Tech. Peddapalli, , India
E-Mail:sri499.gate@gmail.com[1] *, soundarya.pochampalli@gmail.com*[2]

Abstract—Accurate measurement of network bandwidth is im- portant for network management applications as well as flexible Internet applications and protocols which actively manage and dynamically adapt to changing utilization of network resources. Extensive work has focused on two approaches to measuring band- width: measuring it hop-by-hop, and measuring it end-to-end along a path. Unfortunately, best-practice techniques for the former are inefficient and techniques for the latter are only able to observe bottlenecks visible at end-to-end scope. In this paper, we develop end-to-end probing methods which can measure bot- tleneck capacity bandwidth along arbitrary, targeted subpaths of a path in the network, including subpaths shared by a set of flows. We evaluate our technique through ns simulations, then provide a comparative Internet performance evaluation against hop-by-hop and end-to-end techniques. We also describe a number of appli- cations which we foresee as standing to benefit from solutions to this problem, ranging from network troubleshooting and capacity provisioning to optimizing the layout of application-level overlay networks, to optimized replica placement.

INTRODUCTION

MEASUREMENT of network bandwidth is important for many Internet applications and protocols, especiallythose involving the transfer of large files and those involving the delivery of content with real-time QoS constraints, such as streaming media. Some specific examples of applications which can leverage accurate bandwidth estimation include end-system multicast and overlay network configuration protocols [8], [24], [2], content location and delivery in peer-to-peer (P2P) net- works [43], [5], network-aware cache or replica placement policies [25], [40], and flow scheduling and admission control policies at massively-accessed content servers [9]. In addition, accurate measurements of network bandwidth are useful to network operators concerned with problems such as capacity provisioning, traffic engineering, network troubleshooting and verification of service level agreements (SLAs).

Bandwidth Measurement: Two different measures used in end-to-end network bandwidth estimation are capacity band- width, or the maximum transmission rate that could be achieved between two hosts at the endpoints of a given path in the absence of any competing traffic, and available bandwidth, the portion of the capacity bandwidth along a path that could be acquired by a given flow at a given instant in time. Both of these measures are important, and each captures different relevant properties of the network. Capacity bandwidth is a static baseline measure that applies over long time-scales (up to the time-scale at which net- work paths change), and is independent .

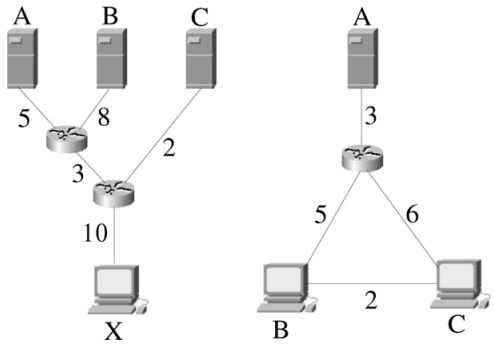

Fig. 1. Leveraging shared bandwidth measurement for optimizing parallel downloads (left) and overlay network organization (right). Numeric labels represent capacity bandwidth of path segments in Mbps.

Available bandwidth provides a dy- namic measure of the load on a path, or more precisely, the residual capacity of a path. Additional application-specific in- formation must then be applied before making meaningful use of either measure. While measures of available bandwidth are certainly more useful for control or optimization of processes operating at short time scales, processes operating at longer time scales (e.g., server selection or admission control) will find esti- mates of both measures to be helpful. On the other hand, many network management applications (e.g., capacity provisioning) are concerned primarily with capacity bandwidth. We focus on measuring capacity bandwidth in this paper.

Catalyst Applications: To exemplify the type of applications that can be leveraged by the identification of shared capacity bandwidth (or more generally, the capacity bandwidth of an ar- bitrary, targeted subpath), we consider the two scenarios illus- trated in Fig. 1. In the first scenario, a client must select two out of three sources to use to download data in parallel. This sce- nario may arise when downloading content in parallel from a subset of mirror sites or multicast sources [6], [42], [15], or from a subset of peer nodes in P2P environments [5]

In the second scenario, an overlay network must be set up between a single source and two destinations. This scenario may arise in ad-hoc networks and end-system multicast systems [8], [24]. For the first scenario illustrated in Fig. 1 (left), the greedy approach of selecting the two servers whose paths to the client have the highest end-to-end capacity bandwidth—namely, servers A and B—is not optimal, since the aggregate bandwidth to the client would be limited by the shared 3 Mbps capacity bandwidth from servers A and B to the client. To be able to select the pair of servers yielding the maximum aggregate bandwidth of 5 Mbps—namely A and C or B and C—the client needs to measure the shared capacity bandwidth between pairs of servers.Similarly, in the second scenario illustrated in Fig. 1 A. Basic Definitions and Assumptions

For the purposes of this paper, a probe is a sequence of one or more packets transmitted from a common origin. We say that any contiguous subsequence of packets within a probe are transmitted back-to-back if there is no time separation between trans- mission of the individual packets within the subsequence. As detailed in the related work section, back-to-back packets have been widely used in estimating the end-to-end bandwidth of a connection [4], [27], [7], [30], [29]. A multi-destination probe is one in which the constituent packets of the probe do not all target the same destination IP address. Multi-destination probes have begun to see wider use as emulations of notional multicast packets—many of the same end-to-end inferences that can be made with multicast packets can be made with multi-destination unicast probes (albeit with added complexity) [12], [17]. A uniform probe is one in which all of the constituent packets are of the same size; likewise, a non-uniform probe consists of packets of different sizes. Finally, we say that an individual packet is hop-limited if its TTL is set to an artificially small value so as not to reach the ostensible destination. Hop-limited packets can be used to trigger an ICMP response from an intermediate router and in other ways that we describe later in the paper.

Throughout the paper we use various probing techniques that rely on sending sequences of probes. The probing techniques differ in the number of packets constituting a probe, the size and the path traversed by each probe packet. They also differ in the host collecting the probing responses and the function used by this host to perform the required estimation.

Each packet transmitted within a probe is parameterized by its size in bits and its final destination, . In the event that a packet is hop-limited, it has a third parameter, its maximum hop-count, . To denote a probe, we refer to each probe packet with a distinct lowercase letter, and represent the sequential order in which they are transmitted from the probing host by writing them from left to right.

We denote interpacket spacing with square braces. As an ex- ample, would denote transmission of a pair of iden- tical two-packet probes followed by a single packet probe which has different characteristics; packets in each of the two-packet probes are transmitted back-to-back while probes are not trans- mitted back-to-back. As another example, denotes a se- quences of identical probe packets that are sent back-to-back. We use the term interarrival time of packets and at a link

to denote the time elapsed between the arrival of the last byte of and the arrival of the last byte of at that link. Similarily, we use the term interdeparture time to denote the time elapsed be- tween the transmission of the last byte of and the transmission of the last byte of . By these definitions, the interarrival time of packets and at a given link is the same as the interdeparture time of packets and at the preceding link on the path.

The constructions and analyses we present later in this paper, are conditioned on a set of basic assumptions about the network. These assumptions, which are common to most probing studies (e.g., [4], [7], [29], [30]), are enumerated below:

1) Routers are store-and-forward and use FIFO queueing.

2) Probing hosts can inject back-to-back packets into the net- work.

(right), the identification of the best set of routes for distributing content from source A to destinations B and C hinges on our ability to determine the capacity bandwidth of the shared portion of the AB and AC paths (as well as the end-to-end capacity bandwidth of path BC). Specifically, it is better to use the AB BC links to provide 3 Mbps to client B and 2 Mbps to client C, rather than the AB AC links for 1.5

Mbps to each (assuming fair sharing).

Paper Scope, Contributions, and Organization: In this paper we propose an efficient end-to-end measurement tech- nique that yields the capacity bandwidth of an arbitrary subpath of a route between a set of end-points. By subpath, we mean a sequence of consecutive network links between any two identifiable nodes on that path. A node s on a path between a source s and a destination d is identifiable if it is possible to coerce a packet injected at the source to exit the path at node

. One can achieve this by: 1) targeting the packet to (if 's IP address is known), or 2) forcing the packet to stop at through the use of TTL field (if the hopcount from to is known), or 3) by targeting the packet to a destination d', such that the paths from to d and from to d' are known diverge at node. Our methods are much less resource-intensive than existing hop-by-hop methods for estimating bandwidth along a path and much more general than end-to-end methods for measuring capacity bandwidth. In particular, our method provides the fol- lowing advantages over existing techniques: 1) it can estimate bandwidth on links not visible at end-to-end scope, and 2) it can measure the bandwidth of fast links following slow links as long as the ratio between the link speeds does not exceed the ratio between the largest and the smallest possible packet sizes that could be transmitted over these links.

The remainder of this paper is organized as follows. In Section II, we review existing literature. In Section III, we develop a basic probing toolkit, comprising existing methods and our new ideas. We compose several of these tools together in Sections IV and IV.E to measure capacity bandwidth along arbitrary subpaths, and capacity bandwidth shared by a set of flows, respectively. In Sections V, VI and VII, we present results of simulation, controlled laboratory experiments and Internet validation experiments, showing the effectiveness of our constructions.

III. PROBING TOOLKIT

In this section, we describe basic constructs of our probing sequences and corresponding terminology. With each probing construct, we describe its properties and point to its usefulness as a building block for the end-to-end measurement of subpath capacity bandwidth, which we describe in Section IV.

Host clock resolution is granular enough to enable accurate timing measurements.

Analytic derivations assume an environment free from cross-traffic.

IV. IMPACT OF CROSS TRAFFIC

In Section IV, we presented an analysis of our end-to-end capacity bandwidth estimation procedures. As stated in Section III, the analysis assumes an environment free from cross-traffic, and it is under this idealistic assumption that we prove the various properties of cartouche probing. Clearly, in any practical setting, cross-traffic cannot be ignored. In this section, we present results from simulations intended to char- acterize the impact of cross traffic on cartouche probing. Our goal in this section is to identify traffic conditions under which cartouche probing is and is not effective. We also find that cross-traffic is not our only worry; we demonstrate scenarios in which structural characteristics of the network path itself impact our results. However, we find that cartouche probing is highly resilient to both the impact of cross-traffic (much more so than packet-pair and tailgating techniques) and structural issues along the network path. This is further reinforced in Section VI and in Section VII, in which results from controlled laboratory experiments and Internet validation experiments are presented.

Prelude: Recall that cartouche probing relies on the preser- vation of spacing between marker packets to estimate the ca- pacity bandwidth of a path segment at endpoints. In general, cross traffic may impact marker spacing in two possible ways: it may cause marker compression, i.e., inter-packet spacing be- tween a pair of markers is reduced in transit, or marker expan- sion, i.e., inter-packet spacing between a pair of markers is in- creased in transit. Both compression and expansion can result from the arrival of cross-traffic at a link [10]. For

example, a burst arriving before the first marker causes the first packet to queue, and results in compression; a burst arriving between the markers can cause the second marker to be delayed, resulting in expansion.

Marker compression is also possible even in the absence of cross traffic. Recall our constructions in Section IV.A. There, we preserved spacing between the two markers used to measuring the capacity bandwidth $b_{i,j}$ as the markers travel over subsequent link. But if is small enough to violate the condition stated in Corollary 1, interpacket spacing is not preserved. To avoid such compression, the size of the cartouches employed must be increased so as to satisfy the conditions of Corollary 1 In effect, Corollary 1 sets a lower bound on, which must be satisfied for our probing constructions to work (as spelled out in the procedure in Section IV.D), and is due entirely to the static properties of the path.

To reduce the effects of cross traffic, a capacity bandwidth measurement experiment must be conducted repeatedly and estimates that may have been affected by marker compression or expansion must be identified and excluded using heuristics [10]. All of our methods require the end-host A conducting the experiment to compile a histogram of the frequency of each estimate it obtains (or a Cumulative Distribution Function CDF). One simple heuristic is to pick the bin with the largest frequency, i.e., the mode (the largest dip in the CDF curve). But with a more refined understanding of how marker compression or marker expansion affects our capacity bandwidth estimation in specific experiments, we develop better alternative heuristics to simply picking the mode. In all our experiments, we use a fixed bin width of 1 Mbps for the histograms.From equations in Section IV, one can see that marker compression results in overestimation of capacity bandwidth, whereas marker expansion results in underestimation of capacity bandwidth. Moreover, as we will subsequently demonstrate, marker compression due to cross traffic is more prevalent in experiments involving path prefixes, whereas marker expansion is more prevalent in experiments involving path suffixes and targeted path segments. This suggests that picking the mode of a histogram in prefix experiments and picking the last mode of a histogram for suffix/subpath experiments are better heuristics to use for filtering the effects of cross traffic.

Experimental Setup: We used the Network Simulator (ns)[34] to simulate a path connecting two hosts A and B. $L_1, L_2, \ldots L_{20}$ consists of 20 physical links L_n, . Link bandwidth values $b_1, b_2, \ldots b_{20}$ were hand-picked to illustrate various scenarios but the default link bandwidth value is set to 100 Mbps. Link latencies $d_1, d_2, \ldots d_{20}$ are all set to 10 msec since they have no impact on the results. Cross-traffic is modeled using a combination of TCP and UDP flows generating equal bit rates (using 32

Kb/sec as the mean flow rate). Cross-traffic flows are hop-persistent, that is any flow traverses only one link. By varying the

number of cross-traffic flows over each link we control the utilization of each of the links, . Packet sizes of cross-traffic flows are equally distributed between 40, 576 or 1500 bytes as suggested in measurement studies on network traffic traces [13]. Probe transmission, time measurements, logging and estimation functions were all performed at host A.

In our experiments, we vary a number of parameters to study the effects of cross traffic. These include: link utilization , cartouche size , length l of the subpath (whether a prefix, suffix, or arbitrary segment) under consideration, and the ratio of the actual bandwidth of the segment under consideration to that of the entire path. Any one of these parameters may alter the spacing of marker packets and thus the accuracy of our bandwidth estimates. in turn leads to a higher probability of cross traffic bursts further separating the markers.

The histograms corresponding to and show instances of overestimation of $b_{1,10}$. These are examples of marker compression due to bursty cross traffic, leading us to overestimate the value of $b_{1,i}$. Our overestimates of are capped at 77 Mbps and 115.5 Mbps, for and , respectively.

Again, this is a direct consequence of the inequality in Lemma

6, which in effect specifies an upper bound on the maximum observable value for $b_{i,n}$ using cartouches of size . This bound (confirmed in Fig. 5) is , which is 38.5 Mbps and 77 Mbps and 115.5 Mbps for 1,2 and 3 respectively. Clearly, when and , our filtering approach was successful in hiding both underestimates and overestimates of $b_{1,10}$ when the utilization is reasonably low (up-to in this case), but filtering becomes more challenging as the utilization increases. As increases, incorrect estimates due to marker compression become much more significant than those resulting from marker expansion. Marker compression is either caused by a violation of the preservation of spacing lemma, and/or by cases in which the first marker is delayed more than

the second marker in the network due to cross traffic. Note that the first marker is more likely to be delayed since a delay of the first marker is due to queued cross traffic at any router, while a delay in the second marker is only due to cross traffic induced during the gap between the markers. For larger utilization values filtering out the last pronounced mode, corresponding to marker expansion helps to locate the correct bandwidth value. also shows that understimates (due to marker expansion) and overestimates (due to marker compression) exist. In fact, when the majority of our trials resulted in underestimates and overestimates. When is smaller then the highest mode (largest CDF dip) is obvious and corresponds to the correct bandwidth estimate. When is larger, then filtering out the modes cor- responding to marker compression and expansion is needed. The same conclusion holds when the targeted path is long as the gaps between probe packets are more prone to dispersion. These results suggest that for long subpaths, it is not advisable to simply use (correspondingly) long cartouche trains. A better divide-and-conquer alternative may be to partition a long sub- path into segments, to which shorter cartouche trains could be applied.

Postlude: We conclude this section with a summary of our findings regarding the susceptibility of our constructions to cross traffic. Specifically, we observe that, in highly congested setups: 1) Marker compression and hence overestimation of $b_{1,i}$ presents the most significant hurdle for capacity bandwidth estimation of path prefix using cartouches. 2) Marker expansion and hence underestimation of $b_{j,n}$ presents the most significant hurdle for capacity bandwidth estimation of path suffix. 3) Both marker compression and expansion are prevalent in arbitrary path segments bandwidth measurement. This difficulty can be alleviated through the use of the smallest cartouches that would satisfy the structural constraints imposed by Lemma 6, through appropriate filtering techniques, and through a divide-and-conquer to limit the size of the cartouche probes approach. VI. CONTROLLED LABORATORY EXPERIMENTS

To evaluate our mechanisms, we incorporated our cartouche probing functionality into a measurement toolkit developed in our laboratory. This toolkit is partially implemented in user space and partially implemented in the kernel (Linux). We use BBSCOPE to refer to this embodiment of cartouche probing in our toolkit. The functionalities of BBSCOPE implemented in the kernel include orchestration of cartouche and cartouche

the hops along the set of paths we considered are fairly well es- tablished, thus giving us a reliable reference against which to test the performance of BBSCOPE.

Fig. 10 shows the histograms we obtained when using

BBSCOPE to estimate b_1, $b_{1,3}$, b_3, b_5, b_3, $b_{5,7}$, b_3 for the path to Georgia Tech and the bandwidth of the last link for the Ecole Normale Superieure path. Clearly, the estimated values are close to the a priori-known bandwidth values. **Comparison to** pchar **and** nettimer: Both pchar [31] and nettimer [30] are hop-by-hop techniques which means that in order to estimate the capacity bandwidth along a path segment they need to run tests over every hop in the segment to estimate its bandwidth and provide the lowest Estimate the final result. On the other hand, Cartouche probing directly targets the capacity bandwidth of the segment.

In comparing against pchar and nettimer, we need to consider two measures: 1) time efficiency: which reflects the time it takes to get a reliable estimate, and 2) byte efficiency: which is a measure of the number of bytes injected into the network to get a reliable estimate. In terms of time efficiency, Cartouche probing is more efficient since it does not have to or- chestrate a round of probing for every hop in a segment before returning the final estimate. Also, in terms of byte efficiency, cartouche probing is more efficient than pchar that uses linear regression in its statistical analysis for every hop which requires injecting the network with lots of extra traffic. In fact, pchar default settings, using a packet size increments of 32 bytes and

32 repetitions per hop, leads to more than 1 MB injected in the network per hop. Cartouche probing needs less than half this number to estimate to capacity bandwidth of a path segment. Cartouche probing is also more byte efficient than nettimer when estimating the capacity along a path prefix in case the path prefix length is larger than the cartouche size and is almost as byte efficient as nettimer when estimating the capacity band- width along arbitrary segments. A cartouche of size has as many bytes as nettimer tailgated pairs and a cartouche train of length and size has as many bytes as tailgated pairs.

VIII. CONCLUSION

We have described an end-to-end probing technique which is capable of inferring the capacity bandwidth

along an arbitrary set of path segments in the network, or across the portion of a path shared by a set of connections, and have presented results of simulations and preliminary Internet measurements of our techniques. The constructions we advocate are built in part upon packet-pair techniques, and the inferences we draw are accurate under a variety of simulated network conditions and are robust to network effects such as the presence of bursty cross-traffic.

While the end-to-end probing constructions we proposed in this paper are geared towards a specific problem, we believe that there will be increasing interest in techniques which conduct remote probes of network-internal characteristics, including those across arbitrary subpaths or regions of the network. We anticipate that lightweight mechanisms to facilitate measure- ment of metrics of interest, such as capacity bandwidth, will see increasing use as emerging network-aware applications optimize their performance via intelligent utilization of network resources.

REFERENCES

[1] B. Ahlgren, M. Bjorkman, and B. Melander, "Network probing using packet trains," Swedish Inst., Technical Report, Mar. 1999.
[2] D. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Re- silient overlay networks," in SOSP 2001, Banff, Canada, Oct. 2001.
[3] S. Banerjee and A. Agrawala, "Estimating available capacity of a net- work connection," in IEEE Int. Conf. Networks (ICON 01), Bangkok,Thailand, Oct. 2001.
[4] J. C. Bolot, "End-to-end packet delay and loss behavior in the Internet,"
in Proc. ACM SIGCOMM'93, Sep. 1993, pp. 289–298.
[5] J. Byers, J. Considine, M. Mitzenmacher, and S. Rost, "Informed
content delivery across adaptive overlay networks," in ACM SIG- COMM'02, Pittsburgh, PA, Aug. 2002.
[6] J. Byers, M. Luby, and M. Mitzenmacher, "Accessing multiple mirror
sites in parallel: Using Tornado codes to speed up downloads," in Proc. IEEE INFOCOM'99, Mar. 1999, pp. 275–83.
[7] R. L. Carter and M. E. Crovella, "Measuring bottleneck link speed in packet switched networks," Performance Evaluation, vol. 27&28, pp.
297–318, 1996.
[8] Y.-H. Chu, S. Rao, and H. Zhang, "A case for end-system multicast,"
in ACM SIGMETRICS'00, Santa Clara, CA, Jun. 2000.
[9] M. E. Crovella, R. Frangioso, and M. Harchol-Balter, "Connection
scheduling in web servers," in Proc. 1999 USENIX Symp. Internet
Technologies and Systems (USITS'99), Oct. 1999.
[10] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet disper- sion techniques measure?," in INFOCOM'01, Anchorage, AK, Apr.
2001.