

# Survey of Bandwidth Estimation Techniques

Ashok Singh Sairam

ashok@iitp.ac.in

Dept. of Computer Science and Engineering

Indian Institute of Technology Patna

Patna 800013, India

Gautam Barua

gb@iitg.ernet.in

Dept. of Computer Science and Engineering

Indian Institute of Technology Guwahati

Guwahati 781039, India

**Abstract**—The ability to locate bottleneck link along Internet paths is of great interest to end users as well as network operators. End users can use the information to estimate performance of a network path to a given destination. The knowledge of bottlenecks can provide network operators insight into the causes of congestion and ways of circumventing it. Flows that share a common bottleneck resource can benefit from cooperative congestion control strategies ([16]). Due to the intriguing nature of the problem and its practical importance, a number of techniques as well as tools have been proposed to measure bandwidth on Internet paths. In this paper we survey a large number of publicly available tools and techniques to estimate bandwidth in stub networks. We found that although there are some tools available that can be adapted to meet our requirement, these tools do not give consistent performance. We therefore devise a simple strategy that provides an upper bound to the estimation results. Our scheme ensures that the results of these tools do not stray beyond acceptable limits. The methodology was validated in actual Internet paths.

## I. INTRODUCTION

Traffic engineering techniques cannot create additional bandwidth; they give preferential treatment to some packets while restricting other packets. In a multihomed site all the links may not be fully utilized at any given instant. While one link is congested other links may be lightly loaded. In order to load balance the links of an end network, the first step would be to monitor and detect congestion on the links. We understand there exists a common bottleneck point in the provider's network through which all flows of the end network has to pass through. We call such a path as the so called *connecting path*. A more detailed explanation of *connecting path* is given in ([4]). The main congestion is assumed to be on this path. In this paper our aim is to devise a methodology for measuring the available bandwidth of the *connecting path*. There are two key technical issues underlying the design of such a tool. Firstly, the tool should be able to identify the common bottleneck point. Secondly it should provide an estimate of the available bandwidth of this bottleneck path. In general, the tool that we seek to design should be able to measure hop-by-hop available bandwidth of Internet paths and locate bottleneck positions in the Internet. Additionally, as our focus is on stub networks the tool should perform the measurements without cooperation from the remote end. The Internet being a collection of uncooperative networks, this is an important requirement from a deployment point of view. Finally the tool should be light-weight.

Given the large array of tools and techniques publicly available to estimate bandwidth, we first review these open source tools. Our aim is to inspect the possibility of tailoring one or more of these tools to meet our prescribed design. The survey is in two parts. Initially in section II we examine those tools which can measure available bandwidth on Internet paths. The intention of this first part of our evaluation is to review the existing technologies used to estimate available bandwidth and comment on their accuracy. In the second part of our survey in section III we analyze those tools that can locate bottleneck points on Internet paths. Based on our evaluation of a large number of tools, we describe the property of a single-ended tool pathneck in section IV, which nearly meets all our requirement. Empirical results however show that the tool pathneck fail to provide accurate results in certain scenarios. In section V, we devise a methodology to check the results of pathneck and validate our scheme on actual Internet paths. The concluding remarks are given in section VI.

## II. BANDWIDTH ESTIMATION TECHNIQUES

In the context of data networks, the term capacity or bandwidth of a path means the maximum bandwidth that a flow can achieve when no other traffic is present. Available bandwidth means the maximum bandwidth that such a path can provide to flows, given the existing traffic it is already carrying. Network measurement techniques are of two categories: passive measurement ([8]) and active measurement ([6], [10]). Passive measurements work on network traces collected earlier. Although they are efficient and accurate, their scope is limited to network paths that have recently carried user traffic. In this work we concentrate on the active probing techniques only. Active probing techniques on the other hand are more useful since they can be used to determine the instantaneous bandwidth. However, active probing techniques will require that additional probe packets are injected into a network path. Typical methods of active probing scheme are packet pair/train dispersion (PPTD), variable packet size (VPS) probing, Self-Loading Periodic streams (SLoPS) and Trains of Packet Pairs (TOPP).

In packet pair probing ([6], [7]), the source sends multiple packet pairs to the receiver. From the measure of the dispersion experienced by the packet pairs, the receiver computes the end-to-end capacity. In order to cancel the influence of cross-traffic on the measurement of capacity a number of packet-pairs are

sent and statistical methods are used to filter out erroneous measurements. In VPS probing ([11], [5]), multiple packets of a given size are sent. This technique uses the TTL field of the IP header to force the probing packets to expire at a particular hop. The source uses the ICMP error messages received from the routers to measure the RTT to that hop. TTLs of the probe packets can be suitably designed such that the TTL of a pair of probe packets expire at each hop. Thus VPS probing can be used to measure the capacity of each hop along a path. In SLoPS ([12]), a series of equal-sized packet probe trains is sent at a particular rate. Depending on the trend of one-way delays experienced by the stream, the sender varies its sending rate and attempts to bring the stream rate close to the available bandwidth. If the streaming rate  $R$  is greater than the path's available bandwidth, the stream will cause a short term overload in the queue of the bottleneck link increasing the one way delays of the probing packets. On the other hand, if the streaming rate is lower than the available bandwidth, the one way delays of the probing packets will not increase. While SLoPS overcomes the inaccuracy in existing probing techniques, it requires a large number of packet streams and a very long measurement time which makes it unsuitable for real-time applications. TOPP ([13]) like SLoPS, sends packet streams and gradually increases the stream rate to measure the available bandwidth. The difference between the two methods is in the statistical processing of the measurements. More details on bandwidth estimation techniques and a taxonomy of public available bandwidth estimation tools can be found in the survey paper by Prasad et. al. ([14]). An up-to-date list of tools for measuring Internet performance is maintained online at [2].

Bandwidth estimation tools can be classified as either a double end-host (DE) or a single end-host (SE) tool ([18]). DE tools must be installed in both the source and destination hosts of target paths, while SE tools in only the source hosts. Since DE tools have access to both ends they are generally more accurate than SE tools, but they are less scalable since cooperation of both ends is required. Our main aim being to find tools that can be deployed in an uncooperative environment, in table I we provide a list of SE tools along with their measurement metric and methodology used. Measuring available bandwidth is complex in practice since it depends on many other factors including latency, loss rate, network path load and TCP implementation details. A number of studies have shown that existing techniques to measure available bandwidth provide only a rough estimate. Estimating available bandwidth requires more coordination between the two ends. This is evident from the limited number of SE tools that are present to measure available bandwidth. Zhou et. al. ([18]) presents a comprehensive analysis of the difficulties faced by DE as well as SE tools in estimating available bandwidth.

### III. LOCATING BOTTLENECK LINKS

In the preceding section, we have examined a number of techniques and tools that can measure the bandwidth of bottleneck links. Our next objective towards measuring

TABLE I  
BANDWIDTH ESTIMATION TOOLS WITH SINGLE-ENDED CONTROL.

Tool	Measurement Metric	Methodology
bing	End-to-end capacity	PPTD
bprobe	End-to-end capacity	Packet Pair
cprobe	End-to-end available bandwidth	Packet Pair
pipechar	End-to-end available bandwidth	Packet Pair
clink	Per-hop capacity	VPS
nettimer	End-to-end capacity	Packet pair
pathchar	Per-hop capacity	VPS
pchar	Per-hop capacity, Latency and Loss	VPS
sprobe	End-to-end capacity	packet pairs
treno	Bulk transfer capacity	Emulated TCP throughput

the available bandwidth of the bottleneck path is to find the location of the bottleneck point. Intuitively, if we can locate the bottleneck point, then we can measure the available bandwidth using one of the tools reviewed in the previous section. In this section, we review four tools that can locate bottleneck points along Internet paths - BFind, Spatio-temporal available bandwidth (STAB), Pathneck and BNeck.

BFind ([3]) operates in a single-ended mode. First, BFind obtains the propagation delay of each hop to the destination. For each hop along the path, the minimum of the measured delays is used as an estimate for the propagation delay on the hop. The minimum is taken over delay samples from 5 traceroutes ([1]). After this step, BFind starts a UDP stream to the destination at a low sending rate and monitors the delay. It also starts concurrently a variant of traceroute. The hop-by-hop delays measured are combined with the raw propagation delay computed initially to obtain rough estimates of the queue lengths on the path. BFind gradually increases the sending rate to build up the queues of intermediate routers. BFind then finds the routers whose queue sizes have increased and the corresponding link is identified as the bottleneck link. The lowest probe rate that induces build up of the queues is regarded as the available bandwidth. The main drawback of BFind is that it needs to send a large amount of data to fill up the links.

STAB ([15]) is a tool that can identify the link with the least available bandwidth on an end-to-end path. The tool introduces several novel concepts like *self induced congestion*, *packet tailgating* and *chirp trains*. Unlike, BFind it is a relatively light-weight tool. However, its main drawback is that it requires access at both end points of the path.

Pathneck ([9]) is an active probing tool based on a novel probing technique called Recursive Packet Train (RPT) which allows end users to locate bottleneck links on the Internet. A choke link is defined as any link that has a lower available bandwidth than the partial path from the source to that link. The upstream router for the choke link is called the choke point or choke router which corresponds to our definition

of congestion point as defined in ([4]). Figure 1 shows an example of a RPT. The idea is to combine measurement and load packets in a single probing packet train. Each box is a UDP packet and the number in the box is its TTL value. Measurement packets are 60 byte UDP packets with properly filled-in payload fields. Load packets are 500 byte UDP packets used to emulate the behaviour of regular data traffic. The probing source sends the RPT packets in a back-to-back fashion. When they arrive at the first router, the first and last measurement packets of the train expire, since their TTL values were set to 1. As a result, the packets are dropped and the router sends two ICMP packets back to the source. The time difference between the arrivals of the two ICMP packets is called the packet gap. Due to the way TTL values are set in the RPT, the above process is repeated at each subsequent router. The sequence of packet gap values from routers along a path is called the gap sequence. The core part of the algorithm for calculating candidate choke points involves matching the gap sequence to a step function, where each step corresponds to a candidate choke point. For each path, the choke points are ranked based on their average gap value in the probing set. Pathneck does not measure available bandwidth of the bottleneck path. However, based on the average per-hop gap values, it provides an upper or lower bound for the available bandwidth of each link.

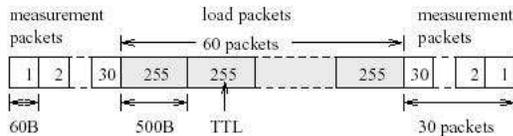


Fig. 1. Recursive Packet Train (RPT).

Alluding to the usefulness of their tool the authors, Ningning Hu et.al claim that pathneck can be used to address the following two questions in a multihomed network - (i) Given a set of popular destinations, which upstream provider should be considered, and (ii) Given a set of upstream providers, which provider should be used to reach a given destination? In actual networks, one would probe from the same source to several destinations through different upstream providers to measure available bandwidth on the uplink paths. In the experiment carried out by Ningning Hu et.al, the probing nodes were not located in the same city. They, therefore, simulate multihomed networks by choosing several nodes and grouping these nodes based on their geographic proximity. For each geographic group, bounds on the available bandwidth of the paths from each member in the group to the same destination are measured. If the lower or upper bound on available bandwidth from the worst path compared with any other path in the group is more than 50 percent, multihoming is declared to be useful. Results show that more than 70 percent of the cases are useful ones.

BNeck ([17]) is an active probing tool that tries to combine the functionality of locating the bottleneck link, measuring link capacity and available bandwidth in a single tool. The design of its packet stream is similar to that of pathneck. The TTL of the packets in the stream increment linearly from both ends but all packets are of the same size. The source estimates the dispersion of the train by measuring the time gap between the two ICMP packets from each router. BNeck once again uses the idea proposed in pathneck to locate the bottleneck points. However, to measure the capacity and available bandwidth of the links, additional probe packets of different sizes are sent by the source. This makes the tool relatively more heavy weight than pathneck.

#### IV. PROPERTIES OF SINGLE-ENDED TOOLS

Based on our evaluation of a large number of publicly available tools to locate bottleneck points and measure its available bandwidth, we find that the tool pathneck nearly fulfills all our requirement. Pathneck has a single-ended control, it is light-weight and can detect bottleneck links on end-to-end path. The only objective that pathneck does not fully meet is that it does not measure the available bandwidth of the bottleneck link but provides an upper or lower bound. Given that the bottleneck point can be identified using pathneck, we can use tools like cprobe or pipechar to estimate the available bandwidth of the bottleneck path. This would, however, mean generating additional probe packets to measure available bandwidth (similar to the idea used in BNeck). Besides, Dovrolis et. al. have demonstrated that what cprobe and pipechar estimate is not the available bandwidth but asymptotic dispersion rate, a quantity which is related to available bandwidth but is not the same ([6]). In view of the above facts, the available bandwidth information provided by pathneck is the cheapest to obtain and is considered good enough.

From our analysis of pathneck, it is evident that the technique employed by the authors is a variant of the variable packet size probing (VPS) bandwidth estimation technique, described above in section II. The measured RTT to each hop consists of three delay components in the forward and reverse paths: serialization (transmission) delays, propagation delays and queuing delays. Thus single end-host tools like pathneck, although primarily want to measure the metrics of the forward path, they unavoidably conflate the delays of both the forward and reverse paths ([18]). While this property of conflating the delays of the reverse path may be a cause of concern for others, in the context of stub networks, it works to their favour. This is because most stub networks are *access networks* that primarily want to download traffic from the Internet. Thus in the case of access networks it basically want to measure the metrics of incoming traffic. Assuming there is very little traffic on the forward path, the metrics measured by pathneck primarily reflect the properties of the backward path. Our validation of pathneck on actual Internet paths in an access network reveals that the delays reported by

pathneck are indeed for the reverse path, destination to source.

Estimates of single-ended tools like pathneck which are largely built on "echo" techniques rely on the timing of the ICMP replies from the target. However, many routers on the Internet ignore ICMP and do not send back ICMP error replies. Tools based on ICMP can also yield significant capacity underestimation errors if the measured path includes store-and-forward layer 2 switches which introduce serialization delays but do not generate ICMP replies because they are not visible at the IP layer.

## V. INTERNET VALIDATION OF PATHNECK

In this section we present the results of validating pathneck on Internet paths. As mentioned in the documentation of the tool, the number of load packets used is critical to its performance. Currently, the number of load packets required for a link is empirical and needs to be discovered using trial and error. In short pathneck does not accurately measure available bandwidth and in certain scenarios it may fail to provide results. To check whether the results provided by pathneck are acceptable, we need to know the instantaneous available bandwidth of all the links on the path. However, it is not feasible to obtain this information for all hops on Internet paths. We, therefore, devise a scheme to provide a check on the results reported by pathneck. We SNMP probe the edge router and measure the incoming octets in a given time period. This value divided by the time period yield the incoming bandwidth utilization of the particular link. The difference of the capacity and utilization of the link is assumed to be the available bandwidth of the incoming path. Strictly speaking, the metric so computed is the available bandwidth of the link that connects the end network to that of the provider's network. Assuming that there is only one path that connects the end network to the provider's network, this link will always be the first hop of any choke point reported by pathneck. Thus our computed value of available bandwidth will be an upper bound on the available bandwidth reported by pathneck.

The tool was tested on different Internet paths. In tables II, III and IV we show the comparison of the output of pathneck with our measured upper bound of available bandwidth on three Internet paths provided by three different ISPs. All bandwidth measurements are in Mbps; "ub", "lb", and "uk" stands for upper bound, lower bound and unknown respectively. If the available bandwidth reported by pathneck is less than our computed upper bound, the output of pathneck is accepted else discarded. We found that under high traffic scenario, pathneck fails to detect choke points. In such cases, the value of the upper bound as computed by probing the edge router is considered as the available bandwidth of the bottleneck path.

<i>Output of SNMP Probe (Upper Bound): 1.449</i>				
<i>Output of Pathneck</i>				
Hop Count	RTT	Choke Point	Estimated Bandwidth	Hop Hostname
01	18.222	.	1.149 (lb)	static115-2.staticcal.vsnl.net.in
02	312.664	.	588.674 (lb)	static191-11.staticcal.vsnl.net.in
03	34.126	1	1.056 (ub)	59.163.16.73.static.vsnl.net.in
04	0.000	.	0.000 (uk)	0.0.0.0
05	78.743	2	1.038 (ub)	59.163.16.13.static.vsnl.net.in
<b>Final Output (Available Bandwidth): 1.056</b>				
Internet Service Provider: VSNL (Capacity: 2 Mbps)				

TABLE II  
DETECTING BOTTLENECK LINKS AND MEASURING ITS AVAILABLE BANDWIDTH ON A LOW BANDWIDTH INTERNET PATH.

<i>Output of SNMP Probe (Upper Bound): 1.16</i>				
<i>Output of Pathneck</i>				
Hop Count	RTT	Choke Point	Estimated Bandwidth	Hop Hostname
01	39.944	1	1.447 (ub)	kol-guwahati-backbone.ernet.in
02	39.678	.	1.447 (lb)	202.141.139.4
03	99.008	.	1.169 (lb)	125.20.0.253
04	118.277	2	0.000 (uk)	125.21.167.25
<b>Final Output (Available Bandwidth): 1.16 Mbps</b>				
Internet Service Provider: ERNET (Capacity: 2Mbps)				

TABLE III  
DETECTING BOTTLENECK LINKS AND MEASURING ITS AVAILABLE BANDWIDTH ON A LOW BANDWIDTH INTERNET PATH.

## VI. CONCLUSION

In this paper we look for an active measurement tool that can be used by stub networks to identify a common bottleneck point in the path connecting the network to its provider network. The tool should also be able to estimate the available bandwidth of the bottleneck path. It should have single-ended control so that an end network can make measurements without cooperation from its upstream providers. Finally the tool should be light-weight and should not introduce a large number of probe packets. We survey the large array of bandwidth measurement tools available in the public domain. We found that the tool pathneck has a single-ended control, it is light weight and it can identify choke points on Internet paths. However, the tool only provides an upper or lower bound of the available bandwidth of the bottleneck link. In order

<i>Output of SNMP Probe (Upper Bound): 31.60</i>				
<i>Output of Pathneck</i>				
Hop Count	RTT	Choke Point	Estimated Bandwidth	Hop Hostname
01	0.492	.	34.422 (ub)	210.212.8.57
02	110.553	.	12905.551 (lb)	210.212.8.50
03	15.616	2	21.954 (ub)	218.248.255.10
04	58.382	3	17.880 (ub)	220.227.53.238
05	58.371	1	11.547 (ub)	220.224.140.33
<b>Final Output (Available Bandwidth): 11.547</b>				
Internet Service Provider: BSNL (Capacity: 34 Mbps)				

TABLE IV  
DETECTING BOTTLENECK LINKS AND MEASURING ITS AVAILABLE BANDWIDTH ON A HIGH BANDWIDTH INTERNET PATH.

to provide a check on the accuracy of pathneck, we probe the edge router of the end network and compute available bandwidth of the Internet path. The value of the available bandwidth so calculated is an upper bound on the output of pathneck. The accuracy of pathneck depends on the number of *load packets* used. The number of load packets required for a link needs to be determined empirically by trial and error. Empirical results show that the results reported by the tool in certain scenarios is higher than the upper bound computed by us. We found that the tool does not provide reasonable estimates or does not produce any output when the network traffic load is very high. Thus in this paper we demonstrate the need to validate the results reported by an estimation tool since their result in certain scenarios may be grossly incorrect. The other point that we establish is that although single-ended tools primarily want to measure the metrics of the forward path, in the case of access networks they measure the delays of the reverse path. This is a positive outcome since access networks basically want to measure the metrics of the incoming traffic. As a future work we can develop generic modules that can be used with existing bandwidth estimation tools to provide an upper bound on the output reported by the tool.

#### REFERENCES

- [1] Traceroute [online]. Available: <ftp://ftp.ee.lbl.gov/traceroute.tar.z>.
- [2] Performance Measurement Tools Taxonomy [online]. <http://www.caida.org/tools/taxonomy/performance.xml>, March 2007.
- [3] A. Akella, S. Seshan, and A. Shaikh. An Empirical Evaluation of Wide-Area Internet Bottlenecks. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 101–114, New York, NY, USA, 2003. ACM.
- [4] A.S. Sairam and G. Barua. Load Balancing in Multihomed Stub Networks. *THE First International Conference on COMMunication Systems and NETWORKS (COMSNETS) (Accepted)*, January 2009.
- [5] S. Bellovin. A Best-Case Network Performance Model. Tech. Rep., ATT Research, Feb. 1992.
- [6] P. Ramanathan C. Dovrolis and D. Moore. What do Packet Dispersion Techniques Measure? In *INFOCOM 2001*, Jan 2001.
- [7] Robert L. Carter and Mark E. Crovella. Measuring Bottleneck Link Speed in Packet-Switched Networks. *Perform. Eval.*, 27-28:297–318, 1996.
- [8] Janey C. Hoe. Improving the Start-up Behavior of a Congestion Control Scheme for TCP. In *SIGCOMM '96: Conference proceedings on Applications, technologies, architectures, and protocols for computer communications*, pages 270–280, New York, NY, USA, 1996. ACM.
- [9] Ningning Hu, Li (Erran) Li, Zhuoqing Morley Mao, Peter Steenkiste, and Jia Wang. Locating Internet Bottlenecks: Algorithms, Measurements, and Implications. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 41–54, New York, NY, USA, 2004. ACM Press.
- [10] Ningning Hu and Peter Steenkiste. Evaluation and Characterization of Available Bandwidth Probing Techniques. *IEEE Journal on Selected Areas in Communications*, 21(6), August 2003.
- [11] V. Jacobson. Pathchar: A tool to Infer Characteristics of Internet Paths. <ftp://ftp.ee.lbl.gov/pathchar/>, April 1997.
- [12] Manish Jain and Constantinovs Dovrolis. End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput. *IEEE/ACM Trans. Netw.*, 11(4):537–549, 2003.
- [13] B. Melander, M. Bjorkman, and P. Gunningberg. A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks. *Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE*, 1:415–420 vol.1, 2000.
- [14] R. Prasad and C. Dovrolis and M. Murray and K. Claffy. Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. In *IEEE Network*, pages 27–35, Nov.-Dec. 2003.
- [15] Vinay J. Ribeiro, Rudolf H. Riedi, and Richard G. Baraniuk. Spatio-Temporal Available Bandwidth Estimation with STAB. *SIGMETRICS Perform. Eval. Rev.*, 32(1):394–395, 2004.
- [16] Dan Rubenstein, Jim Kurose, and Don Towsley. Detecting Shared Congestion of Flows via End-to-End Measurement. In *SIGMETRICS '00: Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 145–155, New York, NY, USA, 2000. ACM.
- [17] H. Zhou, Y. Wang, and Q. Wang. Measuring Internet Bottlenecks: Location, Capacity, and Available Bandwidth. *Networking and Mobile Computing*, 3619/2005:1052–1062, 2005.
- [18] H. Zhou, Y. Wang, X. Wang, and X. Huai. Difficulties in Estimating Available Bandwidth. *Communications, 2006. ICC '06. IEEE International Conference on*, 2:704–709, June 2006.