Telescope: Measuring Capacity between Remote Hosts

Student Name: Aishwarya Jaiswal
Roll Number: 2014007

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science & Engineering
on April 21, 2018

BTP Track: Research Track

BTP Advisor
Dr. Sambuddho Chakravarty

Indraprastha Institute of Information Technology
New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled **Telescope: Measuring Capacity between Remote Hosts** submitted by me for the partial fulfillment of the requirements for the degree of *Bachelor of Technology* in *Computer Science & Engineering* at Indraprastha Institute of Information Technology, New Delhi, is an authentic record of my work carried out under guidance of **Dr. Sambuddho Chakravarty**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

..... **Place & Date: New Delhi, April 21, 2018**
Aishwarya Jaiswal

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

..... **Place & Date: New Delhi, April 21, 2018**
Dr. Sambuddho Chakravarty

Abstract

The term *bandwidth*, *capacity*, and *throughput* characterize the amount of data that can be transferred per unit time. Accurate estimation of bandwidth can be leveraged by various applications which seek to provide best user experience such as peer to peer file transfer. Bandwidth estimation is an age-old problem and many people have sought to estimate it correctly. Many tools exist which employ various strategies in an attempt to measure the capacity of the path such as pathrate, sprobe, nettimer, etc. However, to the best of our knowledge, all of the existing tools require access to atleast one of the endhosts to estimate the capacity. No tool exists which can measure capacity between two remote hosts.

In this project we have proposed a tool which can with sufficient confidence estimate the *one way capacity* between two systems (endhosts or routers) of which we do not possess any access, using a single probing machine, under certain basic assumptions.

Keywords: Network, Capacity, Packet train, Assymetric paths

Acknowledgments

I would like to thank Dr. Sambuddho Chakravarty and his PhD students Devashish Gosain and Piyush Sharma for their constant support in doing this project. Their guidance has enabled me to find my interests in the field of Networks and Security. They have helped me understand various concepts of networks and security. If it wasn't for their guidance and the involving discussions that we engage in, the progress in the project would have been difficult. I thank them for showing me the way to do good research.

Last but not the least, I would also like to thank my family and friends for their constant support and encouragement without which it would have been difficult for me to pursue B.Tech from IIIT-Delhi.

Work Distribution

Extensive experiments have been done in Winter Semester of 2018 with modifications to the approach that was proposed in Monsoon Semester of 2017.

Contents

Chapter 1

Introduction

Bandwidth is a key parameter for measuring network performance and is crucial for many internet applications and protocols, especially those involving transfer of large files and those involving delivery of real time data such as live video streaming. Some examples of applications which can leverage the use of accurate bandwidth estimation could be content delivery in P2P networks, flow scheduling in CDNs, leasing links to customers by ISP's etc. In addition, such measurements can also be useful in traffic engineering, network operations, network trouble shooting, etc.

Bandwidth quantifies the data rate at which the network link or network path can transfer, i.e. the amount of data that can be transmitted over the network link per second from *source* to *destination*. There are multiple measures which are used to quantify bandwidth such as 'available bandwidth', 'capacity', 'throughput', 'good-put', etc (explained in section 3.1). A lot of work has already been done in this field and different works use different measures, such as [2, 11, 13, 18] estimate available bandwidth and [6, 9, 15, 19] estimate capacity of a path (aka, bottleneck bandwidth). Not only this, there are different methodologies that have been exploited by the researchers in this community to measure the said parameter such as packet pairs, packet trains, TCP behaviour emulation, etc(refer [17]).

However, despite of the extensive literature in the field, to the best of our knowledge *no work has yet been done to estimate the bandwidth between the two remote machines*, *i.e.*, the approaches cited above require access to *at least one* of the endhosts for measuring the bandwidth/capacity between them, which also makes them incapable for calculating one way bandwidth. Internet, by itself has an asymmetric property, *viz*, the path taken by the packets to reach from $A \rightarrow B$ might not be the same as the path traversed between $B \rightarrow A$ (This can be validated by running traceroute between two accessible endhosts). Hence, a prominent consequence one can observe is that, the bandwidth/capacity might not be same on both paths. Hence, approaches which observe packet dispersion at one end might not be able to capture this property effectively.

In this project, we propose our new tool "Telescope" which finds the *one way capacity* between two remote hosts in the network, given that both the remote hosts are neither in our network and nor we have access to them. We introduce a novel approach to infer bandwidth using IP Identifiers.

Our tool only requires the use of a single measurement machine, that is the machine that is going to estimate the capacity between the off path remote hosts. The only requirement is that one of the host has an open port and that the second host responds to the unsolicited SYN/ACK packets with RST packets on a closed port.

1.1 Motivation and Problem Description

Different works that have been done in the field of bandwidth measurement require access to at least one of the end host. Example, Iperf [21], a very popular bandwidth estimation tool, requires access to both the hosts, where one acts as the server and the other acts as the client. Other tools like Linkwidth [5, 12] require access to only one of the machines for active measurement of bandwidth. To the best of our knowledge, no work has yet been done in estimating capacity between two off-path remote hosts, i.e., hosts which we cannot access. Since Capacity of network path is a static measure independent of the traffic at any given instant, it is a more robust metric to compare bandwidth between any 2 given paths. Such a tool could be useful in applications which requires a long uninterrupted communication of data. Since the path can not be full at all times, the application would be able to utilize the maximum capacity at some point which is what is desired.
Also, the existing tools of bandwidth measurement rely on the assumption that the network path between two hosts A and B is symmetric, i.e. the path from $A \rightarrow B$ is same as the path from $B \rightarrow A$ and thus would have the same bottleneck bandwidth [9–12, 18, 21]. However, this is not the case. There exist paths in the network where path from $A \rightarrow B$ is not the same as path from $B \rightarrow A$. Hence in such cases the existing tools may give inaccurate results because of path inconsistencies. Also, such works do not consider the asymmetricity in upload and download link rates which also affects the estimation of bandwidth.

Problem Description

In this project, we developed a tool to measure capacity with certain confidence interval, between arbitrary internet nodes using IP Identifier fields with the assumption that the network paths are asymmetric. The two machines are neither in the same network nor the probing machine has any direct access to the them.

Chapter 2

Background and Relevant Work

Ample amount of research has already been done in estimating bandwidth between two hosts such as Pathchar [12], Pathchirp [18], Linkwidth [5], Nettimer [15], Iperf [21] and many more. Prasad and Dovrolis in their seminal work [17] compared the various bandwidth estimation tools, metrics and measurement techniques where all of them use a different approach for measuring bandwidth between the hosts.
One of the other parameters that can be used for comparing bandwidth estimation techniques is to check whether they require access to one machine or both machines. Another parameter could be whether they measure one-way-bandwidth or 2-way-bandwidth. Tools like [5,12,15,19] require access to only one machine to estimate bandwidth, whereas [4,6,9,11,18,21] require access to both the endhosts for their measurements. Similarly, all the tools which requires access to only one endhost, measures one-way-bandwidth, as it is someway or the other dependent on the responses from the other machine.

Another way one could classify the existing measurement strategies is the bandwidth metric[1] they measure. Almost the entire bandwidth measuring community is divided into two categories in this respect - the ones which measure available bandwidth across a path and ones which measure the maximum capacity of the path. Former is estimated by [4,13,18,21], and the latter is estimated by [6,15,19].

There also exists abundant literature on the use the global IP Identifier field (of the IP header). In 1998, Antirez was the first person who identified the use of this global IP Identifier (global IP-ID) field for TCP Idle Scan in NMAP [1]. Subsequently, global IP-ID field was extensively exploited by many works, like [3,7,8,14,16,20].

[1] Refer [17] for more details

Chapter 3

Definitions

3.1 Bandwidth

Bandwidth is the amount of data that can be transmitted from one host to the other over the network path per second. The term is generally confused between a variety of throughput related concepts. We define the various concepts here
Bottleneck bandwidth aka Capacity: It is the maximum bandwidth that can be achieved between two hosts at the endpoints of a given path in the absence of any competing traffic, i.e., it is equal to the slowest link rate (i.e., link with least transmission rate) in the given path
Available Bandwidth: It is the portion of the bottleneck bandwidth along a path that can be acquired by any given flow at a given instant of time
Throughput: It is the rate at which the data is traversing in the link
Good-put: It is the amount of useful data that traverses a link (e.g. in a TCP scenario, good-put is the amount of data that reaches the destination - including re-transmissions)

Bottleneck bandwidth and available bandwidth are both important measures and both capture different relevant properties of the network. Bottleneck bandwidth is a static baseline method which applies over long time-scales and is independent of the traffic dynamics at a particular time instant. Available bandwidth however is a dynamic measure of the load on the path, more precisely the residual capacity of the path, and hence can be different over different time instants. While both measures are of great importance, both find use in different applications. Here, however, we are interested in finding the bottleneck bandwidth or capacity of the network, that is the maximum rate at which data can travel in the absence of traffic.

3.1.1 Factors on which Bandwidth depends

Bandwidth depends on a number of factors:

- **Transmission rate** - the rate at which the NIC can transfer data onto the wire

- **Queuing delay** - delay which occurs at the routers due to traffic from other flows
- **Queue size** - If the size of the queue at the router is really small, then more packet drops would occur, decreasing good-put of the network path
- A number of other factors like processing delays at routers, application level delays, drop of packets due to error control and flow control, number of clients connected to server, etc.

Bottleneck bandwidth, however depends only on the minimum transmission rate of the individual links in the network and Available bandwidth/ good-put depends on the queue size and the queuing delay of the routers on the network path between SND and RCV.
Let P be a network path with H hops, that transfers packets from SND to RCV. We assume that the path is fixed and unique and no multi-paths exists between the SND and RCV. Each link i can transmit data with rate C_i, which is its transmission rate. The capacity/ *bottleneck bandwidth*[1] is defined as

$$C = \min_{i=1...H} C_i$$

Please note, that in measuring bottleneck bandwidth, the link i with the least link rate, would create the maximum dispersion in traffic sent from SND (as this is the link which would take the maximum time to transfer data on to the link - rate inversely proportional to time). Hence, at the receiver, if we are able to measure this dispersion in the traffic, we would thus be able to measure the minimum link rate of the path P, that is the capacity.

3.2 Packet Train

A packet train is nothing but large number of packets send in succession. Packet trains have been used for bandwidth measurement since a long time now. One example of a packet train which is very well known is the Recursive Packet Train, used to find the *location* of the bottleneck bandwidth in [10].

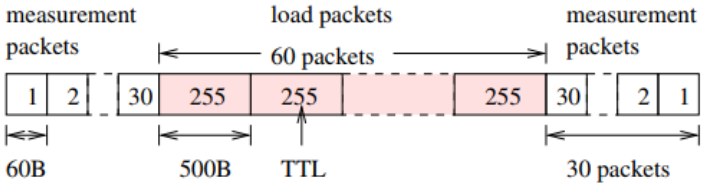

Figure 3.1: Recursive packet train in [10]

The Recursive Packet Train (RPT), includes sending 30 small ICMP packets at the start with increasing TTL values, followed by 60 large UDP packets, again followed by 30 trailing ICMP packets with decreasing TTL values. Linkwidth, [5] also uses a similar packet train to measure the available bandwidth between two hosts A and B.

[1] as defined by [13]

3.3 IP-spoofing

Ip-spoofing is a technique using which a user/adversary modifies the IP field within the IP packet (i.e., changes selfIP $\rightarrow$ newIP) to impersonate some other host in the network or hiding one-self's identity from the proxy server. This way, receiver believes the packets to be coming from the host with $IP\ address = newIP$.
One legitimate place where IP spoofing is put into use is in NATing, where the NATing device, modifies the private IP of the internal user to the public IP of the NATing device.

3.4 IP Packet Identifier aka IP-ID

When an Internet hosts generates and sends IP Packets, each generated IP packet contains a 16-bit IP Identifier value (referred to as IP-ID in the rest of the report) that is intended to assist end points in reassembling the fragmented IPv4 packets. There are only 2^{16} possible values, but the intent of the field is that, packets from the same host, should have different IP-ID values (unless fragmented, in which case, all fragmented packets would have same IP-ID value).
Depending on the implementation, IP-ID values could be of 4 types:

1. **Single Global IP-ID counter**: Here, there is only a single IP-ID counter that is maintained across all flows. That is IP-ID is incremented for all packets that originate from that host regardless of whether the packets bear any relationship to each other

2. **Separate IP-ID counter per host**: Here each Packets from the same flow/ connection, have different IP-ID values. However, across flows, there is no relationship between the IP-ID values

3. **Constant**: As the name suggests, for each generated packet at the host IP-ID is a constant value for e.g., 0

4. **Random**: In this case, for any packet generated at the host, the IP-ID field is set to any random number generated by the pseudo random generator

Chapter 4

Telescope

In this section, we present our approach for estimating the capacity between the remote machines.
Let the machines between which we wish to measure the one-way capacity be R_f (Reflector- which reflects the data) and R_e (Receptor - which receives the data). Let the vantage point (whose access we possess) be MM (the measurement machine).

4.1 Pre-requisites

In this section we entail the necessary requirements for different entities involved in our model. In general we assume that all $\{R_f$, R_e and $MM\}$ machines are **not** behind any firewall nor do they perform any kind of ingress or egress filtering.

The specific requirements of different entities are enlisted below.

Measurement Machine Requirements

- IP-Spoofing: MM is capable of IP spoofing as it impersonates as R_e to R_f, throughout the experiments.
- Capacity constraints: capacity between MM and R_f must be greater than or equal to capacity between R_f and R_e *viz.,* $C(MM, R_f) >= C(R_f, R_e)$. If this condition fails, Telescope would rather measure $C(MM, R_f)$, and will only give a lower bound on the $C(R_f, R_e)$[1].

[1]We first measure $C(MM, R_f)$ and then $C(R_f, R_e)$; if $C(MM, R_f) >= C(R_f, R_e)$ holds good, then only we report the exact value of $C(R_f, R_e)$, else we provide the lower bound on the capacity.

Reflector Node Requirements

- Port constraint: at least one TCP port must be open *i.e.*, R_f must reply with SYN/ACK packet for a new SYN packet
- IP-ID counter: The machine should implement local/global counter

Receptor Node Requirements

- Port constraint: It must have at least one TCP port closed *i.e.*, R_e must generate RST packets in response to SYN/ACKs as per RFC-793
- IP-ID counter: it should have a **global IP-ID counter**

4.2 Approach

We detail our approach below.

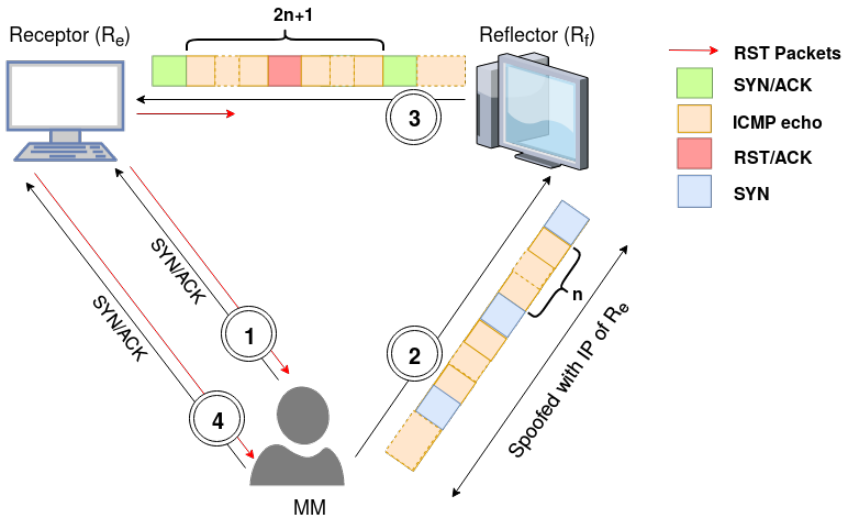

Figure 4.1: Identifying capacity using Telescope: MM estimates the capacity between R_f and R_e machines

1. MM sends unsolicited SYN/ACK packet to R_e, which replies with RST packet to MM. The reply contains the *global* IP-ID value of R_e recorded at time t.
2. MM sends a packet train (as shown in figure 4.2) to R_f with spoofed source IP address of R_e. The packet train constitutes of multiple *chunks* and each chunk is composed of one SYN (with destination port as any open port on R_f and source port as any closed port on R_e) and n ICMP echo request packets (each of 1400 bytes). All SYN packets are identical to each other, except for the sequence numbers *viz.*, the difference between the sequence numbers of two successive SYN packets is always one.

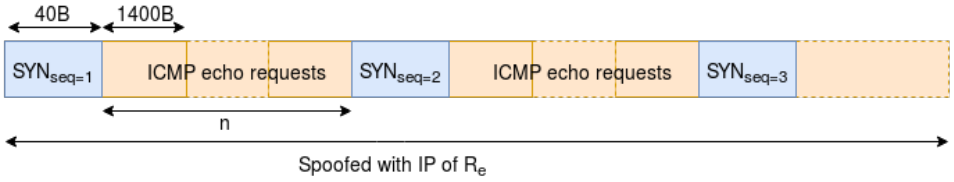

Figure 4.2: Packet Train which MM sends to R_f

3. (a) R_f on reception of aforementioned packets, sends a new packet train as described in figure 4.3 to R_e. For first SYN received, it generates SYN/ACK; for n ICMP echo requests, it generates n ICMP echo reply; and for next SYN received, it generates RST/ACK. The same pattern repeats, but for all received SYN packets R_f alternatively generates SYN/ACK and RST/ACK packets[2]. This behavior — alternating between sending SYN/ACK and RST/ACK for successive TCP SYN packets — can be attributed to pre-RFC 5961. [It suggests that for multiple SYN request arriving within the receive window, the receiver responds with a RST packet otherwise, it signals its peer with ACK.]

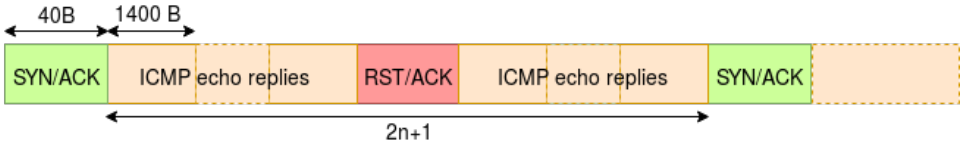

Figure 4.3: Packet Train which R_f sends to R_e

For first SYN received, R_f generates a SYN/ACK packet assuming new connection has been initiated, but on reception of second in window SYN packet with same source IP and port, it sends a RST/ACK packet[3]. On reception of third SYN packet it again generates a SYN/ACK packet assuming a new connection has been initiated. This process is repeated for every successive SYN packet received. Between two SYN packets there are 'n' ICMP echo request and they will be processed normally *viz.,* for every received ICMP echo request, echo reply will be generated by R_f

(b) When R_e, receives the aforementioned packet train (shown in Figure 4.3, for every SYN/ACK packet it generates a RST packet[4], causing its global IP-ID counter to be incremented by the number of SYN/ACK packets it receives. It must be noted that, between two SYN/ACK packets, there will be $2n$ ICMP echo reply messages (of 1400 bytes each) and one RST/ACK packet (refer to figure 4.3) which will be dropped by the kernel of R_e.

4. In order to determine the increment in IP-ID, MM probes R_e with SYN/ACK packets

[2]For eg., for total of 10 SYN packets, R_f generates 5 RST/ACK and 5 SYN/ACK.

[3]This happens because R_f does not receive RST packet from R_e for the previously generated SYN/ACK in time before the next SYN arrives. Hence R_f closes the existing half open connection with R_e and generates a RST/ACK.

[4]For machine R_e, SYN/ACK packets are unsolicited as it has not sent any SYN packet and are also received at its closed port. Thus as a response, its kernel generates a RST packet

every δ time units. R_e replies with RST packets that contain the *global* IP-ID value of R_e at that instant. The difference in IP-ID, obtained from the responses of two successive measurement probes, is proportionate the volume of traffic that has been received at R_e in δ time. If in $\delta + RTT_{MM,R_e}$ unit time, MM observes the IP-ID to be incremented by X, it estimates the capacity between R_f and R_e as:

$$C_{est}(R_f, R_e) = \frac{(X) * (2 * n) * 1400}{\delta + RTT_{MM,R_e}}$$

Chapter 5

Experimental Setup and Results

We have performed some preliminary experiments with our tool - Telescope. However, extensive experiments on a larger scale are yet to be performed.

5.1 Experimental Setup

For our experiments we hosted 2 unfirewalled public IPs in our IIIT-D network - one with global IP-ID and the other which has IP-Spoofing enabled in it. Both the systems have 1Gbps. We have also created a number of Digital Ocean machines and also have got access to public IP machines in few institutes outside having similar configurations to perform extensive experiments. For different experiments we shape the outbound link of Reflector (R_f) to link rates like 10Mbps, 100Mbps and 1000Mbps. For all our experiments, we have kept our MM fixed in IIIT-D.

We have divided our experiments into 3 categories:

- Fully Controlled - We possess access to both R_f and R_e, and both are machines are installed in our lab. This set of experiments are done without cross-traffic. Here we created a local network of 4 Systems in the lab, where the R_f and R_e are at 2-hop distance with one another, with the MM being connected to both R_f and R_e
- Semi-Controlled - The machines R_f and R_e are in our control[1]. However these machines are placed out in the wild (e.g., - in NewYork, Frankfurt, Singapore, Bangalore, etc) and have real internet traffic between them, over which we do not have any control
- Fully Random - Both the machines are not in our control and are placed in the wild (in real internet)

For the Fully Random experiment, we have created a script which finds public IPs over the internet which have global IP-IDs.

[1]That is, we can shape the outbound link of these systems and also monitor their traffic using packet capturing tools like wireshark, tcpdump

5.2 Results

All the results shown below are obtained over an average of 5 iterations. We have performed Fully Controlled and Semi-Controlled Experiments. However, Fully Random Experiments are still left to be performed.

5.2.1 Fully Controlled

We first performed our experiments in the fully controlled environment, i.e, in the absence of traffic. Results are shown in Table 5.1.

Actual Capacity between R_f and R_e (in Mbps)	**Capacity Estimated using Telescope (in Mbps)**
10	9.26
100	95.41
1000	878.6

Table 5.1: Results in Fully controlled environment: We can see that Telescope is able to measure Capacity sufficiently well in the absence of any cross Traffic

5.2.2 Semi-Controlled

Below are a subset of the experiments we conducted for Semi controlled environment[2] over the internet, i.e., in the presence of real traffic.

R_f	R_e	**Shaping at R_f (Mbps)**	**Capacity Estimated using Telescope (Mbps)**
Columbia	IIIT-D	10	9.10
Columbia	IIIT-D	100	84.23
Columbia	IIIT-D	1000	743.28
Columbia	New York	10	9.29
Columbia	New York	100	88.56
Columbia	New York	1000	838.70

Table 5.2: Results in Semi Controlled environment:We can see that, even in the presence of real internet traffic, Telescope is able to estimate Capacity fairly accurately.

[2]In these experiments R_e at IIITD denotes the public IP machine with global IP-ID.

Chapter 6

Work in Progress

We have developed our tool and have performed some preliminary experiments. However, we are yet to perform the more extensive experiments on a large scale.

- Fully Controlled experiments done
- We need to add more location diversity for Semi Controlled Experiments by testing our tool for more R_f and R_e pairs distributed across the world, such that both are in our control
- We also need to perform fully Random Experiments in which both R_f and R_e are not in our control. Currently, we have already found some 1500 public IPs world wide which have global-IPs in them (using a script which we have developed ourself). We need to create R_f-R_e pairs for these and run our tool on this to obtain final results.

Bibliography

[1] Tcp idle scan. `https://nmap.org/book/idlescan.html`. Accessed: 2018-04-21.

[2] Beaumont, O., Eyraud-Dubois, L., and Won, Y. J. Using the last-mile model as a distributed scheme for available bandwidth prediction. In *European Conference on Parallel Processing* (2011), Springer, pp. 103–116.

[3] Bellovin, S. M. A technique for counting natted hosts. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment* (2002), ACM, pp. 267–272.

[4] Carter, R. L., and Crovella, M. E. Measuring bottleneck link speed in packet-switched networks. *Performance evaluation 27* (1996), 297–318.

[5] Chakravarty, S., Stavrou, A., and Keromytis, A. D. Linkwidth: a method to measure link capacity and available bandwidth using single-end probes. *Computer Science Department Technical Report (CUCS Tech Report) CUCS-002-08, Columbia University* (2008).

[6] Dovrolis, C., Ramanathan, P., and Moore, D. What do packet dispersion techniques measure? In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (2001), vol. 2, IEEE, pp. 905–914.

[7] Ensafi, R., Knockel, J., Alexander, G., and Crandall, J. R. Detecting intentional packet drops on the internet via tcp/ip side channels: Extended version. *arXiv preprint arXiv:1312.5739* (2013).

[8] Ensafi, R., Winter, P., Mueen, A., and Crandall, J. R. Analyzing the great firewall of china over space and time. *Proceedings on privacy enhancing technologies 2015*, 1 (2015), 61–76.

[9] Harfoush, K., Bestavros, A., and Byers, J. Measuring bottleneck bandwidth of targeted path segments. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies* (2003), vol. 3, IEEE, pp. 2079–2089.

[10] Hu, N., Li, L. E., Mao, Z. M., Steenkiste, P., and Wang, J. Locating internet bottlenecks: Algorithms, measurements, and implications. In *ACM SIGCOMM Computer Communication Review* (2004), vol. 34, ACM, pp. 41–54.

[11] Hu, N., and Steenkiste, P. Estimating available bandwidth using packet pair probing. Tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2002.

[12] Jacobson, V. Pathchar: A tool to infer characteristics of internet paths, 1997.

[13] Jain, M., and Dovrolis, C. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput. *ACM SIGCOMM Computer Communication Review 32*, 4 (2002), 295–308.

[14] Keys, K., Hyun, Y., Luckie, M., and Claffy, K. Internet-scale ipv4 alias resolution with midar. *IEEE/ACM Transactions on Networking (TON) 21*, 2 (2013), 383–399.

[15] Lai, K., and Baker, M. Nettimer: A tool for measuring bottleneck link bandwidth. In *USITS* (2001), vol. 1, pp. 11–11.

[16] Pearce, P., Ensafi, R., Li, F., Feamster, N., and Paxson, V. Augur: Internet-wide detection of connectivity disruptions. In *Security and Privacy (SP), 2017 IEEE Symposium on* (2017), IEEE, pp. 427–443.

[17] Prasad, R., Dovrolis, C., Murray, M., and Claffy, K. Bandwidth estimation: metrics, measurement techniques, and tools. *IEEE network 17*, 6 (2003), 27–35.

[18] Ribeiro, V. J., Riedi, R. H., Baraniuk, R. G., Navratil, J., and Cottrell, L. pathchirp: Efficient available bandwidth estimation for network paths. In *Passive and active measurement workshop* (2003).

[19] Saroiu, S., Gummadi, P. K., and Gribble, S. D. Sprobe: A fast technique for measuring bottleneck bandwidth in uncooperative environments. In *IEEE INFOCOM* (2002), p. 1.

[20] Spring, N., Mahajan, R., Wetherall, D., and Anderson, T. Measuring isp topologies with rocketfuel. *IEEE/ACM Transactions on networking 12*, 1 (2004), 2–16.

[21] Tirumala, A., Qin, F., Dugan, J., Ferguson, J., and Gibbs, K. Iperf: The tcp/udp bandwidth measurement tool. htt p. *dast. nlanr. net/Projects* (2005).