

FEAT: Improving Accuracy in End-to-end Available Bandwidth Measurement

Qiang Wang and Liang Cheng

Laboratory Of Networking Group (LONGLAB, <http://long.cse.lehigh.edu>)
Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA
qiw3@lehigh.edu, cheng@cse.lehigh.edu

Abstract—This report describes our ongoing work on a new end-to-end available-bandwidth measurement tool: FEAT (Fish-Eye Available-bandwidth Tool). FEAT features a new dynamic pattern of probes called a *Fisheye Stream*. One fisheye stream covers a range of packet probing rates. A fisheye stream consists of a focus region, where the probing rates are sampled more frequently and the number of packets used at each rate is larger. This creates a fisheye effect that the focus region enables an easily identified “turning point” for accurate measurements. When the dynamic available bandwidth is outside the region, the surrounding regions enable the tool to automatically “refocus”. Fisheye streams offer several advantages over current probing schemes that are based on packet pairs, packet trains, or packet chirps. FEAT is also fast, nearly non-intrusive, and clock resolution insensitive. Experiments show that FEAT provides accurate estimations of the available bandwidth with low overhead compared to the existing techniques.

Keyword—available bandwidth, measurement, fisheye stream

I. INTRODUCTION

In recent years, available bandwidth (“avi-bw” for short) measurement techniques have received extensive attentions in the research community. Avi-bw measurement tools have been used in network-aware adaptive applications [3], admission control [2], route selection [1], TCP slow start [5], and TCP throughput [14] etc. Generally speaking, end-to-end available bandwidth is the unused capacity of a network path, and it is mathematically formulized in [15] etc.

The avi-bw measurement requires a fast, accurate, and non-intrusive technique. These requirements raise many challenges in the decentralized Internet. End-to-end avi-bw

measurement techniques usually use active probing packets from the network edge. Several active probing tools have been studied in recent years, including TOPP [12], pathload [7], IGI [6], pathChirp [15], spruce [16], netest [9], Pathmon [10], etc.

This report describes our on-going work on a new end-to-end avi-bw tool: FEAT (Fish-Eye Avi-bw Tool). It has been studied in experiments to achieve accurate estimations of the available bandwidth comparing with other tools. FEAT also provides a fast measurement in seconds. Moreover, FEAT is nearly non-intrusive by carefully scheduling the fisheye streams.

II. BANDWIDTH MEASUREMENT MODELS AND TOOLS

Probe Gap Model (e.g. Spruce, IGI) and Probe Rate Model (e.g. Pathload, Pathchirp) have been studied for the avi-bw measurements [16]. FEAT is a tool based on the rate model.

A. Rate Model

The rate model (a.k.a, self-induced congestion) is based on the following heuristic argument: If the rate of a probing stream exceeds the avi-bw of the path from the sender to the receiver, a short-term congestion happens at the tight link (i.e. the link with the smallest avi-bw). A queue builds up on the tight link with the interleaving of probe packets and cross traffic packets.

1) Metrics of OWD and relative OWD

The self-induced congestion causes that the queuing delay and the OWD (one-way delay) increase when the congestion happens. The process of estimating the avi-bw is then transformed into identifying the turning point at which the OWD sees an increasing trend. Relative

OWD is also used as a metric by some tools.

Pathload [7] uses a CBR (constant bit rate) probing stream and changes the rate in its successive rounds. **Pathload** adjusts the rate of the packet train based on the receiver’s feedback. **Pathload** narrows the probing rate space using a binary search scheme. The final output after multiple rounds of a measurement is a value range that contains the available bandwidth.

pathChirp [15] uses a VBR (varying bit rate) probing stream called a chirp. Packets in a chirp are exponentially spaced. Effectively, a chirp probes the network over a range of rates. If the OWD shows an increasing trend starting from a particular packet, the corresponding packet rate is used to estimate the avi-bw. In case of VBR cross traffic, **pathChirp** identifies excursions. It should be noticed that a chirp stream samples the sending rate with different frequency because packets are exponentially spaced. **FEAT** also uses VBR probing streams covering a range of probing rate. However, the pattern of a fisheye stream in **FEAT** is different.

2) Metrics of offered rate and measured rate

Besides the OWD, several other metrics are used by different measurement tools. **TOPP** uses a metric of offered rate R_0 (packet sending rate) and a measured rate R_m at the receiver (packet receiving rate). If $R_0 > \text{avi-bw}$, then $R_m < R_0$. The turning point is the maximum offered rate such that $R_m \approx R_0$.

3) Metrics of cumulative sending gap and received gap

Similar to the offered/measured rate metrics, if $\text{sending_rate} > \text{avi-bw}$, the inter-packet intervals increase at the receiver side. The relationship between the sending gap and the receiving gap is difficult to be identified due to measurement noises. Thus **PathMon** uses a cumulative inter-packet gap, which is relatively easier to detect the divergence point of two gap curves.

The accuracy of a tool highly depends on efficient filtering technologies. Through experiments and observations, **FEAT** uses the OWD as the metric since it does not need

advanced filtering technologies to find the turning point. Moreover, **FEAT** uses a fisheye stream to make the turning point earlier to be identified and thus it improves the measurement accuracy. Our future work will study whether one metric is more robust than the others in the presence of measurement noises.

B. Gap Model

The gap model has several assumptions. (1) Cross traffic is of a fluid type. (2) The narrow link and the tight link are identical. (3) Packets must be queued together before being transmitted to the narrow link. (4) Accurate timing is needed to compute the packet gaps.

Spruce sends out a pair of packets in each round. The intra-pair gap in **Spruce** is the transmission time of the first packet on the tight link (under the assumption #2). This enables the two packets to be queued together. When the gap model assumptions do not hold, **Spruce** is expected not performing well. IGI experimentally determines the initial packet-pair gap that gives a high correlation between the packet-gap changes and the cross traffic on the tight link. The turning point is where the average gap of a packet pair equals the average destination gap of the pair.

III. FEAT: FISH-EYE AVAILABLE-BANDWIDTH TOOL

A. Fisheye Stream

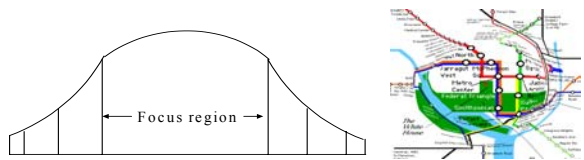


Figure 1: A fisheye magnifier and a city map under the fisheye lens

The fisheye-stream measurement can be analogized as a fish-eye magnifier, where objects in the middle are shown in great details while objects in the surroundings are visible but not in great details as shown in Fig. 1.

The fisheye stream is based on observations that the smaller the sample intervals around the turning

point, the more accurately we can find the turning point. The larger the number of packets at the same rate around the turn point, the more quickly the queue will be built up. A fisheye stream consists of K packets of equal size that are sent at a changing rate from L (the lower bound) to U (the upper bound). Inside this range, there is an area called a focus region, where the avi-bw to be measured is most likely to be in this range. (We will explain how to refocus the region when the avi-bw is not inside the region). Around the center, packet instantaneous-rate sampling is more frequent. The number of packets at each sampling rate is larger. Outside the focus region, the rate sampling is less frequent; however, it still covers the range from L to U .

A fisheye stream is different from a CBR stream used by `pathload` and a chirp stream by `pathChirp`. Compared with the CBR stream, the fisheye stream covers a range of rates in one stream. `Pathload` has to send a sequence of streams in order to have a proper coverage from the lower-bound rate to the upper bound. The process of the binary search is time-consuming.

Comparing to the chirp stream in `pathChirp`, the fisheye stream is similar in the sense that one stream covers a range of rates. However, the fisheye stream has two distinct features. First, the rate sampling method is different. A chirp stream samples the low rate more frequently than the high rate, and its probing-packet pattern is not adaptive to historical values of the avi-bw. For example, for a spread factor $r=1.2$, if a chirp uses 27 packets for a rate range from 1 Mbps to 120Mbps, the first 13 packet rates are all less than 10 Mbps, while the last 5 packet rates corresponds to the avi-bw from 55.2 Mbps to 114.5 Mbps. In contrast, a fisheye stream has more sampling frequency around the center of its focus region. Second, the number of packets at each rate is different. The chirp stream uses one packet for each packet rate, while the fisheye stream uses more packets for rates within the focus region. The self-induced congestion requires a stream to build up a queue with a certain number of packets whose rates are larger than the avi-bw. The high sampling frequency and the large number of

packets at each rate in the focus region create a turning point that is easier to be identified. Previous technologies have focused on how to find a turning point, while the fisheye stream focuses on how to create a more identifiable turning point.

The actual function of creating a fisheye stream is shown in Fig. 2. The default value of the lower bound L is set as 10 Kbps. The upper bound U is the Average Dispersion Rate (ADR) as used in `pathload` [4]. The focus center C is adjusted after each round of the measurement, and it is initially at the center of L and U . The focus length is also adjustable and it is initially set to the 1/5 of the total range ($U-L$). N is the height of the focus lens. Outside the focus region, the number of packets used for each rate drops from N to 1.

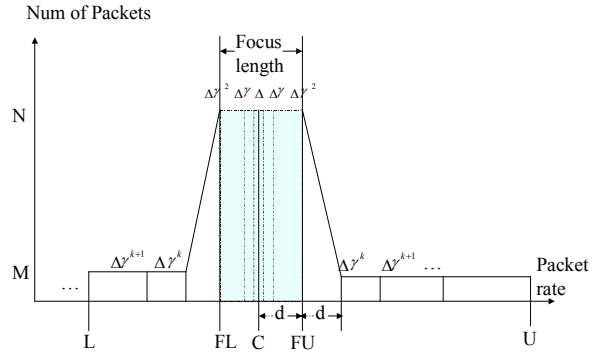


Figure 2: A fisheye stream

The rate sampling function is similar to that of `pathChirp`. Assume that the instantaneous packet rate at the focus center is C . Starting from C towards the upper bound U , instantaneous packet rates are $C+\Delta$, $C+\Delta r^1$, $C+\Delta r^2$, ... It is essentially an exponential function as a chirp. However, from C towards the lower bound L , instantaneous packet rates are $C-\Delta$, $C-\Delta r^1$, $C-\Delta r^2$, Thus it is a reversed chirp. The default value of r is 1.2 and the default value of Δ is $(U-L)/100$.

B. Dynamics of the Available Bandwidth

The design philosophy of refocusing the fisheye stream is based on the long-term stability and predictability of the Internet. Results in [17] show that the change-free durations of the throughput often last more than one hour. The available bandwidth of an Internet path usually shows a

certain degree of constancy within a few minutes or even hours.

In order to decide the width of the focus region, we collect the MRTG (Multi-Router Traffic Grapher [13]) data from 123 routers in our campus network in a two-day time period and calculate the avi-bw change percentage, which is the difference of two sequential measurements divided by the link capacity. Our data shows about 90% of the avi-bw change percentages are within 5%. Significant avi-bw changes are relatively less frequent. The MRTG gives a 5-minute average and the short-term avi-bw would be more bursty. Therefore we use 20% of the link capacity as the width of the focus region in order to accommodate the variance of avi-bw. FEAT can also automatically adjust the focus-region width based on the dynamics of historical data.

C. Measurement Process

The first step of FEAT is to estimate the upper bound U of the avi-bw capacity using the well-known Average Dispersion Rate technology [4]. This process only needs to be executed once whenever a new path is measured.

FEAT takes advantage of the constancy of the Internet by refocusing on previously measured avi-bw value. The sender sends the fisheye stream to the receiver. The receiver uses similar filtering functions as those in pathChirp to find the turning point and compute the corresponding avi-bw x . If x is in the focus region, the sender displays x as one reportable measurement result and refocuses the focus region by moving the center C to x . If x is not in the focus region, FEAT refocuses the center C to x and uses additional round(s) to obtain a reportable result until x falls in the focus region.

IV. VALIDATIONS

A. Testbed Validation

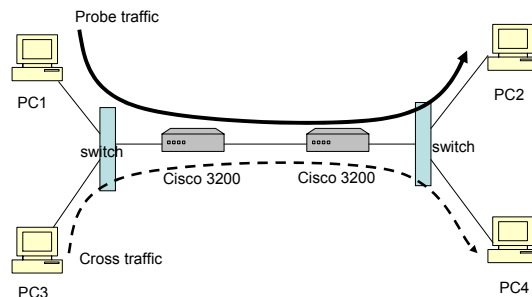


Figure 3: Testbed configuration

Fig. 3 shows the testbed configuration. Two Cisco 3200 routers are connected back-to-back using fast serial interfaces. The speed of the back-to-back connection is adjustable by setting the clock rate of the interfaces. Each router connects to a Cisco Catalyst 2900 switch. Two Dell Dimension PCs, i.e. PC1 and PC2, using Linux Red Hat 9 run FEAT and other avi-bw tools. Another two PCs, i.e., PC3 and PC4, run a commercial traffic generator [11] to generate controllable cross traffics. We set the inter-router link bandwidth as 1.3 Mbps and generate 0.2, 0.4, ..., 1.2 Mbps CBR cross traffic. We do not tune these tools, except set the capacity parameter 1.3 Mbps for Spruce. We test each tool 20 times in each cross traffic configuration and plot the averaged results in Fig. 4. The true value of the avi-bw is the straight line with a slope -1 shown in Fig. 4.

PathChirp does not have the process of detecting the upper bound of avi-bw. In every new measurement, pathChirp adjusts the probe range from [10M .. 200M], [1.43M .. 70M] to [0.204M .. 10M]. We only use results in the [0.204M .. 10M] range. Via discussions with pathChirp author, the [0.204 .. 10M] range is still too large for our configuration, which could explain why pathChirp did not perform well in our tests.

It takes about 270 seconds in average for pathload to finish a test. IGI uses 1 to 3 seconds when the cross traffic is light, and increases to about 40 seconds when the cross traffic approaches 1.2 Mbps. Spruce, pathChirp and FEAT are all under one second per test.

Our experiments show that **pathload** constantly underestimates the avi-bw. **Pathchirp** and **IGI** constantly overestimate the avi-bw. **Spruce** results in a mixed pattern. **IGI** frequently underestimate the tight link capacity in our tests, with very occasions that the tight link capacity is many times higher than the true value. These exceptions have been excluded in the statistics as outliers. All tools are fairly accurate when the link utilization is low, while relative errors are large when the link utilization is high. When the link utilization is from low to high, the performance of **FEAT** is consistent, e.g. with 90% of times the relative errors are less than 20%.

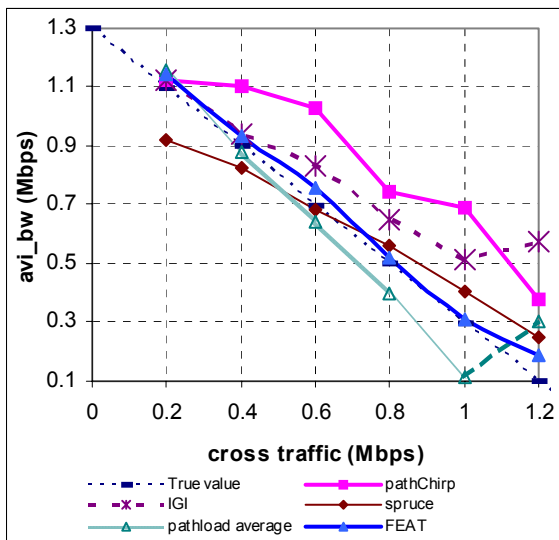


Figure 4: Available-bandwidth (avi-bw) test results

B. Internet Test Methodology

Testbed validations and ns-2 simulations could verify the idea of a tool. However, there is no better approach than using actual network paths. Similar to [16], we validate **FEAT** with the help of the **MRTG** tool on network paths. Although the 5-minute resolution of the **MRTG** data is low, the **MRTG** tool is so far the most accurate way to validate the output of an avi-bw tool.

MRTG validation requires accesses to **MRTG** log data from all links of a path and the capacity of each link. We have collected **MRTG** data of 123 routers in Lehigh campus and downloaded the freely available Abilene network **MRTG** logs. We have applied the **MRTG** test to several paths that

traverse the Lehigh campus network. Our ongoing tests include paths from Berkeley, U. Northwestern, U. Florida to Lehigh. The initial Internet test shows positive results. However, extensive Internet testing is still an ongoing work and we expect the result in next few months. Our Internet validation will test the relative error, absolute errors as well as the agility of the tools over these selected Internet paths.

V. CONCLUSIONS AND FUTURE WORK

This report introduces **FEAT**, an end-to-end available bandwidth tool that features a fisheye probing packet stream. The tool has been tested in the lab testbed setting as well as a small number of Internet paths. Current experiments indicate that **FEAT** is accurate, fast and nearly non-intrusive.

We will deploy **FEAT** in larger scales, integrate it with middleware [18] for adaptive distributed applications, and investigate the sources of errors to improve its accuracy. We also plan to test it on Emulab [19], which can create controllable multi-hop scenarios that are hard to test in the Internet. An extensive study of algorithms in various tools is also needed in order to better understand the theoretical and practical issues of the available bandwidth measurement.

ACKNOWLEDGEMENT

This work is sponsored by National Science Foundation under the NSF Award #0438300 – SCI: NMI DEVELOPMENT: Middleware for Adaptive Robust Collaborations across Heterogeneous Environments and Systems (MARCHES).

REFERENCES

- [1] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, Resilient overlay networks, in *Proc. ACM SOSP*, pp. 131-145, 2001.
- [2] L. Breslau, E. Knightly, S. Shenker, and I. Stoica, Endpoint admission control: architectural issues and performance, in *Proc. ACM SIGCOMM*, pp. 57-69, 2000.
- [3] S.-W. Cheng and D. Garlan, Software architecture-based adaptation for pervasive

- systems, in *Proc. of Intl. Conf. on Arch. of Computing*, 2002.
- [4] C. Dovrolis, P. Ramanathan, and D. Moore, What do packet dispersion techniques measure? in *Proc. of IEEE INFOCOM*, pp. 905-914, 2001.
- [5] N. Hu and P. Steenkiste, Improving TCP startup performance using active measurements: algorithm and evaluation, in *Proc. of ICNP 2003*.
- [6] N. Hu and P. Steenkiste, Evaluation and characterization of available bandwidth probing techniques, *IEEE JSAC*, Vol. 21, No. 6, 2003.
- [7] M. Jain and C. Dovrolis, Pathload: a measurement tool for end-to-end available bandwidth, in *Proc. of PAM*, pp. 14-25, March 2002.
- [8] G. Jin and B. L. Tierney, System capability effects on algorithms for network bandwidth measurement, in *Proc. of ACM SIGCOMM Internet Measurement Workshop 2001*, pp. 27-38, 2001.
- [9] G. Jin and B. Tierney, Netest: a tool to measure the maximum burst size, available bandwidth and achievable throughput, LBNL report #48350.
- [10] D. Kiwior, J. Kingston, and A. Spratt, PATHMON: a methodology for determining available bandwidth over an unknown network, in *Proc. of IEEE Sarnoff 2004 Symposium*, pp. 27-30, April 26-27, 2004.
- [11] ZTI Co. Ltd., LanTraffic V2.0., www.zti-telecom.com
- [12] B. Melander, M. Bjrkman, and P. Gunningberg, A new end-to-end probing and analysis method for estimating bandwidth bottlenecks, in *Proc. of IEEE Globecom*, pp. 415-420, 2000.
- [13] T. Oetiker, MRTG: the multi router traffic grapher, in *Proc. of the 12th Conference on Systems Administration (LISA'98)*, pp. 141-148, 1998.
- [14] R. Prasad, M. Jain, and C. Dovrolis, Socket buffer auto-sizing for high-performance data transfers, *Journal of Grid Computing*, 1(4), 2004.
- [15] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, pathChirp: efficient available bandwidth estimation for network paths, in *Proc. of Passive and Active Measurement Workshop*, April 2003.
- [16] J. Strauss, D. Katabi, and F. Kaashoek, A measurement study of available bandwidth estimation tools, in *Proc. of 2003 ACM SIGCOMM Conf. on Internet Measurement*, pp. 39-44, Oct. 2003.
- [17] Y. Zhang, N.G. Duffield, V. Paxson, and S. Shenker, On the constancy of Internet path properties, in *Proc. of ACM SIGCOMM Internet Measurement Workshop 2001*, pp. 197-211, 2001.
- [18] Q. Wang and L. Cheng, AwareWare: an adaptation middleware for heterogeneous environments, in *Proc. of ICC*, pp. 1406-1410, 2004.
- [19] Emulab: <http://www.emulab.com/>