Appears in PAM 2005 ©*2005 Springer-Verlag*

Applying Principles of Active Available Bandwidth Algorithms to Passive TCP Traces

Marcia Zangrilli and Bruce B. Lowekamp

College of William and Mary, Williamsburg VA 23187, USA

Abstract. While several algorithms have been created to actively measure the end-to-end available bandwidth of a network path, they require instrumentation at both ends of the path, and the traffic injected by these algorithms may affect the performance of other applications on the path. Our goal is to apply the self-induced congestion principle to passive traces of existing TCP traffic instead of actively probing the path. The primary challenge is that, unlike active algorithms, we have *no control over the traffic pattern* in the passive TCP traces. As part of the Wren bandwidth monitoring tool, we are developing techniques that use single-sided packet traces of existing application traffic to measure available bandwidth. In this paper, we describe our implementation of available bandwidth analysis using passive traces of TCP traffic and evaluate our approach using bursty traffic on a 100 Mb testbed.

1 Introduction

Available bandwidth is typically measured by actively injecting data probes into the network. The active approach often produces accurate measurements, but it may cause competition between application traffic and the measurement traffic, reducing the performance of useful applications. Most of these active algorithms rely on UDP traffic to probe the path for available bandwidth, but most applications use TCP traffic. However, UDP traffic may be packet-shaped differently than TCP traffic and measurements made with UDP traffic may not reflect the actual bandwidth available to TCP applications. Furthermore, these available bandwidth algorithms require instrumentation on both ends of the path, which may not always be possible.

Our goal is to use passive traces of existing TCP traffic instead of actively generating the traffic being used to measure available bandwidth. By monitoring the traffic that an application generates or receives, we can calculate the available bandwidth on the path even when the application has not generated sufficient traffic to saturate that path. Our available bandwidth measurements can be used by an application already generating traffic to determine if it can increase its sending rate, by network managers who are interested in observing traffic, capacity planning, SLA monitoring, etc., or by central monitoring systems[1, 2] that store measurements for future use or use by other applications.

Because our approach uses existing application traffic to measure available bandwidth, it does not place additional burden on the network path and it experiences the same packet shaping issues affecting applications. To achieve the necessary accuracy and avoid intrusiveness, our passive monitoring system uses the Wren packet trace facility [3] to collect kernel-level traces of application traffic and analyzes the traces in the user-level. Our trace facilities can be deployed on one or two end hosts or on a single packet capture box, which is an advantage over tools that must be deployed on both ends of the path.

This paper describes how to apply the self-induced congestion principle to passive traces of application traffic, a task complicated because we have no control over the application traffic pattern. We describe an algorithm for applying the self-induced congestion principle of active probing to passive, one-sided traces of TCP traffic and demonstrate that our algorithm produces measurements that are responsive to changes in available bandwidth.

2 Background

Available bandwidth describes what portion of the path is currently unused by traffic. More precisely, available bandwidth is determined by subtracting the utilization from the capacity of the network path [4, 5]. In practice, available bandwidth may also be affected by traffic shapers that allow some traffic to consume more or less bandwidth than other traffic can consume.

The basic principle of the self-induced congestion (SIC) technique is that if packets are sent at a rate greater than the available bandwidth, the queuing delays will have an increasing trend, and the rate the packets arrive at the receiver will be less than the sending rate. If the one-way delays are not increasing and the rate the packets arrive is the same as the sending rate of the packets, then the available bandwidth is less than or equal to the sending rate. Tools that utilize this concept [6–10] probe the network path for the largest sending rate that does not result in queuing delays with an increasing trend because this sending rate reflects the available bandwidth of the path.

Proposed improvements to the TCP protocol have set a precedent for measuring available bandwidth on a single end host. Paced Start (PaST) [11] proposed incorporating self-induced congestion principles used by the PTR tool to reduce the amount of time taken before transitioning into the congestion avoidance phase of TCP.

3 Passive One-Sided SIC Implementation

Our one-sided passive SIC implementation uses the timestamps of data and ACK packets on the sending host to calculate the round trip times (RTT) and the initial sending rates of the stream of packets. Our implementation is similar to the pathload tool [8], which uses trends in one-way delays to determine the available bandwidth.

We group packets together into streams and identify the trend in RTTs of each packet group. We impose the condition that grouped packets are the same size so that all packets we consider have experienced the same store-and-forward delays at the links along the path. Because congestion window size often determines the sending rate of the TCP application, we also ensure that all packets grouped together have the same congestion window size. The number of packets in each group is determined by the congestion window size. For each stream of packets, we calculate the RTTs of each packet, calculate the initial sending rate, and determine if there is an increasing trend in the RTTs. We group several streams together and try to identify the maximum value for the available bandwidth. For each group, the stream with the largest sending rate and no increasing trend determines the available bandwidth.

To emulate traffic generated by bursty applications, we created traffic generators that send 256K messages with a variable delay. The variable delay causes the throughput of the generators to oscillate.

Figure 1 presents the results of applying our passive SIC approach to one-sided traces of two traffic generators. In the left graph, the average throughput of the traffic generator is 65 Mbps on an uncongested LAN. The traffic generator was run on a 100 Mb testbed with varying amounts of cross traffic present. This graph shows distinct bands that demonstrate our algorithm can detect the changes in the amount of available bandwidth.

The second traffic generator was designed to send out bursts of messages with varying throughput. In the right graph in Fig. 1, the line represents the throughput of the traffic generator and the points are the measurements produced by our passive algorithm. In this experiment, there is 20 Mbps of cross traffic present for the first 15 seconds and 40 Mbps of cross traffic present for the last 15 seconds. This graph shows that our SIC algorithm can accurately measure changes in the available bandwidth using one-sided traces of applications with bursty communication patterns.

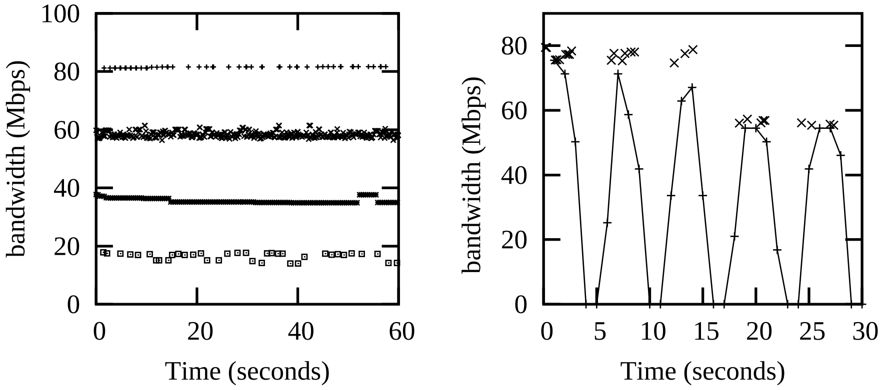

Fig. 1: The left graph shows our SIC algorithm applied to one-sided traces of bursty traffic on a 1OO Mb testbed with 20, 40, 60, and 80 Mbps of cross traffic. The right graph shows the SIC measurements (points) produced by monitoring bursty application traffic (tput represented as the line) on a testbed with 20 Mbps of cross traffic present during 0–15 seconds and 40 Mbps of cross traffic during 15–30 seconds.

4 Conclusion

We have described the implementation of a passive available bandwidth technique based on the self-induced congestion principle. Our preliminary evaluation of our one-sided passive SIC technique is quite promising and shows that we can obtain valid available bandwidth measurements in congested environments using bursty application traffic.

We are continuing to evaluate our one-sided passive SIC algorithms. We are interested in qualifying what types of traffic patterns are best suited for our algorithm to produce valid measurements, and performing more detailed analysis on the affects of traffic burstiness, bottlenecks, and delayed ACKs on the accuracy of our algorithm's measurements.

References

1. Wolski, R.: Forecasting network performance to support dynamic scheduling using the network weather service. In: Proceedings of the 6th High Performance Distributed Computing Conference (HPDC). (1997)
2. Adams, A., Mahdavi, J., Mathis, M., Paxson, V.: creating a scalable architecture for internet measurement. In: Proceedings of INET'98. (1998)
3. Zangrilli, M., Lowekamp, B.B.: Using passive traces of application traffic in a network monitoring system. In: Proceedings of High Performance Distributed Computing (HPDC 13), IEEE (2004)
4. Prasad, R., Murray, M., Dovrolis, C., Claffy, K.: Bandwidth estimation: Metrics, measurement techniques, and tools. In: IEEE Network. (2003)
5. Lowekamp, B.B., Tierney, B., Cottrell, L., Hughes-Jones, R., Kielmann, T., Swany, M.: Enabling network measurement portability through a hierarchy of characteristics. In: Proceedings of the 4th Workshop on Grid Computing (GRID). (2003)
6. Jin, G., Tierney, B.: Netest: A tool to measure the maximum burst size, available bandwidth and achievable throughput. In: International Conference on Information Technology Research and Education. (2003)
7. Ribeiro, V., Riedi, R.H., Baraniuk, R.G., Navratil, J., Cottrell, L.: pathChirp:Efficient Available Bandwidth Estimation for Network Paths. In: Passive and Active Measurement Workshop (PAM). (2003)
8. Jain, M., Dovrolis, C.: Pathload: a measurement tool for end-to-end available bandwidth. In: Proceedings of 3rd Passive and Active Measurements Workshop. (2002)
9. Hu, N., Steenkiste, P.: Evaluation and characterization of available bandwidth techniques. IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling (2003)
10. Melander, B., Bjorkman, M., Gunningberg, P.: A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. In: Global Internet Symposium. (2000)
11. Hu, N., Steenkiste, P.: Improving tcp startup performance using active measurements: Algorithm and evaluation. In: International Conference on Network Protocols (ICNP). (2003)