Towards Tunable Measurement Techniques for Available Bandwidth

Ningning Hu [1], Peter Steenkiste[1,2]
[1]School of Computer Science, [2]Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA 15213, USA
{hnn, prs}@cs.cmu.edu

1. Abstract

In this paper, we propose the notion of a *tunable available bandwidth measurement tool* that can be adapted to the execution environment. We first briefly describe our work on IGI and PTR, and then classify and compare the current available bandwidth measurement techniques. We then report on our experience in adapting PTR for the use during TCP Slow Start, which leads to the idea of a tunable measurement tool. Finally, we present the research challenges that must be addressed to develop such a tool.

2. Classification

In [Hu and Steenkiste, 2003a], we present two available bandwidth measurement algorithms — the *Initial Gap Increasing* (IGI) and the *Packet Transmission Rate* (PTR). Both tools are based on the insight that the sending rate of the probing train is the most important parameter that must be tuned to obtain an accurate measurement. The right measurement point is at the turning point, which is the highest rate for which the packet train rate at source and destination are the same.

IGI and PTR are only two of the active probing techniques that have been designed to measure the end-to-end available bandwidth. The others include Pathload [Jain and Dovrolis, 2002], pathChirp [Ribeiro et al., 2003], and Spruce [Strauss et al., 2003]. Table 1 shows a classification of these techniques. We use two criteria in the classification:

1) *What to measure*. We can directly measure the transmission *rate* of a packet train to estimate the available bandwidth, as is done in pathChirp, Pathload, and PTR. Alternatively, we can measure the amount of competing traffic on the bottleneck link to indirectly estimate the residual bandwidth. This is done by measuring the changes in the probing packet *gap*, as is done in Spruce and IGI.
2) *How to measure*. All tools use packet pairs, either sent individually or as a train, but they differ in how the packet pair gaps are controlled by the sender. Pathload, IGI, and PTR use packet trains with uniform intervals. In contrast, in pathChirp and Spruce, the packet intervals are statistically constructed, thus the packet train or the sequence of packet pairs is non-uniform.

Different categories have different properties and consequently, they have different advantages and disadvantages:

1) *Assumption.* The techniques that measure the background traffic to estimate the available bandwidth need to know the path capacity. Spruce assumes it is known, while IGI estimates it using existing probing techniques. The problem is that any error in the path capacity estimate directly impacts the available bandwidth measurement accuracy. Rate-based techniques do not have this problem.
2) *Measurement interval.* How the probing trains are constructed affects the averaging interval that is used for the available bandwidth estimate. The uniform probing techniques generally use short packet trains, so they get a relatively short-term snapshot of network performance. Since they measure the available bandwidth averaged over a very short time interval, the estimates will change quickly when the background traffic is very bursty. In contrast, non-uniform probing techniques use statistical sampling over a longer period thus, for example, average out the effects of bursty traffic.

Besides the above differences, all these available bandwidth measurement techniques also share some problems:

1) *System related timer problem.* All techniques rely on the correctness and accuracy of the system timer and the network packet delivery model: any errors that the sending and receiving systems introduce in the timing of the packets will reduce the accuracy of the tools. The timing accuracy becomes more important as the available bandwidth increases. This could be a serious problem on very high speed network (VHSN), not only because of the limits of timer resolution, but also because they use different packet delivery mechanisms (e.g. batching). Note that techniques that use the timing of individual packet gaps are more sensitive to this effect than techniques that measure packet train rates.
2) *Two-end control.* All current techniques need two-end control, which significantly hinders deployment. Control at the destination is needed to accurately measure the packet gap or packet train rate.

3. Tunability

As discussed above, no single technique works best in all network environments. This leads us to the notion of *tunable measurement techniques*. Here "tunable" means that the measurement algorithms can be adapted to the *execution environment*, which includes both the network and the application requirements. Understanding the network properties may allow us to improve the measurement accuracy (e.g. adjust the length of packet trains), while understanding the application needs can help in making sure that we measure the right value efficiently (e.g. tune the tradeoff between accuracy and overhead).

Let us use the development of *Paced Start* (PaSt) [Hu and Steenkiste, 2003b], a new TCP startup algorithm, to demonstrates the idea of tunability. PaSt is an adaptation of the PTR available bandwidth probing algorithm. PaSt adjusts the sequence of TCP data packets transmitted during the TCP

Table 1. Comparison of current available bandwidth measurement algorithms

		how to measure			
		non-uniform probing	uniform probing	difference	common
what to measure	a_bw (rate)	**pathChirp**	**Pathload, PTR**	not need B	timer problem
	c_bw (gap)	**Spruce**	**IGI**	need B	two-end control
	difference	long interval	small interval		

(a_bw: available bandwidth; c_bw: background traffic throughput; B: bottleneck link capacity)

startup period so that they function as probing trains. As a result, they can be used to estimate the path available bandwidth, thus *quickly* identify the proper congestion window size, often without packet loss. Many properties of PaSt directly follow from the application (i.e. TCP) requirements:

1) *Accuracy and overhead*: TCP is an adaptive protocol and the purpose of the startup phase is to get a reasonable starting point for the congestion avoidance phase. At the same time, we would like to switch to congestion avoidance mode quickly, so it is important to keep the overhead (number of probing packets) low. Given these two requirements, PaSt cuts off the measurement more quickly than PTR.
2) *Measurement interval*: Since the congestion control tries to track the available bandwidth, it needs the available bandwidth averaged over a short interval. PaSt uses trains up to a roundtrip time in length.
3) *Two-end control*: Since TCP is a two-end protocol, PaSt automatically has access to the destination node. Note however that for the ease of deployment, PaSt measures the packet train rate at the source (based on ACKs), i.e. it is a one-end implementation.

Other applications will of course have needs that are different from TCP, so we need a tunable measurement technique for available bandwidth. Such a tool will have to consider the following network and application factors.

Averaging interval. As discussed above, non-uniform probing like sampling can be used to improve the accuracy by using large measurement intervals, but it could also have a very big probing overhead. On the other hand, uniform probing, as used by IGI/PTR, has a small probing overhead. However, it may be possible to combine the two, for example by using sequences of short packet trains.

Accuracy versus overhead. The right tradeoff is application dependent but it also depends on a deep understanding on how to control the accuracy and how to control the overhead, given the properties of the network path. There are two properties to start with. One is to estimate the burstiness of the network path. [Hu and Steenkiste, 2003a] has mentioned a possible way to obtain that information from the available bandwidth probing. The other is to get the confidence interval, as is done in [Jain and Dovrolis, 2002] and is also implied by the theoretical model in [Hu and Steenkiste, 2003b].

Extreme environment. By "extreme environment", we mean network scenarios that are not common today but will be in the future (e.g. VHSN and wireless). [Jin and Tierney, 2003] is a good start by discussing the problems of current active probing techniques on VHSN. How well active probing works in VHSN is currently an open question. A first question is where the *measurement bottleneck* is. If the end host system is the bottleneck, the available bandwidth may be system dependent or even application dependent. If the measurement bottleneck is in the network, we need to be very careful about the algorithm design, for example because of timer granularity limits. However, active probing may still be practical, for example, by using a larger packet train. In this context, a high level research question is: *is there an upperbound/lowerbound for the active bandwidth measurement techniques? If yes, what are the factors that determine these bounds?*

Two-end control. It would be very advantageous to be able to run available bandwidth probes without support from the destination node. However, all current techniques assume support on both the sender and the receiver. The fundamental reason for using two-end control is that by measuring the gap values or packet train rate on the destination node, we eliminate the effects of queueing in the reverse path and the asymmetry of Internet paths. To develop one-end solutions, we will have to improve our understanding of how and to what extend reverse-path queueing affects the measurement. Note that this is not just a low-level technical question, it can also be a question for the whole network architecture.

While the original motivation for our PaSt research was to improve the performance of TCP Slow Start, we discovered that the work also improved our understanding of available bandwidth probing techniques such as PTR. The reason is that the best available bandwidth probing technique depends in part on the precise application requirement. As we use available bandwidth probing in more applications, we will improve our understanding of the design space. The goal is to develop a tool that not only adapts to the network path properties but can also be tuned to meet specific application needs.

References

Ningning Hu and Peter Steenkiste. Evaluation and characterization of available bandwidth probing techniques. *IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling*, 21 (6), August 2003a.

Ningning Hu and Peter Steenkiste. Improving TCP startup performance using active measurements: Algorithm and evaluation. In *IEEE International Conference on Network Protocols (ICNP) 2003*, Atlanta, Georgia, November 2003b.

Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In *SIGCOMM 2002*, Pittsburgh, PA, August 2002.

Guojun Jin and Brian Tierney. System capability effect on algorithms for network bandwidth measurement. In *Internet Measurement Conference (IMC) 2003*, Miami, Florida, October 2003.

Vinay Ribeiro, Rudolf Riedi, Richard Baraniuk, Jiri Navratil, and Les Cottrell. pathchirp: Efficient available bandwidth estimation for network paths. In *Passive and Active Measurement Workshop 2003*, La Jolla, CA, April 2003.

Jacob Strauss, Dina Katabi, and Frans Kaashoek. A measurement study of available bandwidth estimation tools. In *Internet Measurement Conference (IMC) 2003*, Miami, Florida, October 2003.