A Merged Inline Measurement Method for Capacity and Available Bandwidth

Cao Le Thanh Man, Go Hasegawa and Masayuki Murata

Graduate School of Information Science and Technology, Osaka University
{mlt-cao, hasegawa, murata}@ist.osaka-u.ac.jp

Abstract. We have proposed a new TCP version, called ImTCP (Inline measurement TCP), in [1]. The ImTCP sender adjusts the transmission intervals of data packets, and then utilizes the arrival intervals of ACK packets for the *available bandwidth* estimation. This type of active measurement in a TCP connection (inline measurement) is preferred because it delivers measurement results that are as accurate as active measurement, even though no extra probe traffic is injected into the network. In the present research, we combine a new *capacity* measurement function with the currently used measurement method to enable simultaneous measurement of both capacity and available bandwidth in ImTCP. The capacity measurement algorithm is essentially based on the packet pair technique, but also consider the estimated available bandwidth values for data filtering or data calculation, so that this algorithm promises better measurement results than current packet-pair-based measurement algorithms.

Extended abstract

The capacity of an end-to-end network path is the maximum possible throughput that the network path can provide. Traffic may reach this maximum throughput when there is no other traffic along the path. The available bandwidth indicates the unused bandwidth of a network path, which is the maximum throughput that newly injected traffic may reach without affecting the existing traffic. The two bandwidth-related values are obviously important with respect to adaptive control of the network. In addition, these two values are often both required at the same time. Although network transport protocols optimize link utilization according to capacity, congestion is also avoided by using the available bandwidth information. For routing or server selection in service overlay networks, information about both capacity and available bandwidth offers a better selection than either capacity or available bandwidth information alone. For example, when the available bandwidth fluctuates often and the transmission time is long, the capacity information may be a better criterion for the selection. However, when the available bandwidth appears steady during the transmission, then the available bandwidth should be used for the selection. Moreover, the billing policy of the Internet Service Provider should be based on both the capacity and the available bandwidth of the access link they are providing to the customer.

Several passive and active measurement approaches exist for capacity or available bandwidth. Active approaches are preferred because of their accuracy and speed. However, sending extra traffic onto the network is a common disadvantage that is shared by all active measurement tools.

We propose herein an active measurement method that does not add probe traffic to the network. The proposed method uses the concept of "plugging" the new measurement mechanism into an active TCP connection (*inline measurement*). Passive inline measurement appeared in TCP Westwood [2], in which the sender checks the ACK packet arrival intervals to infer the available bandwidth. We herein introduce ImTCP (Inline measurement TCP), a Reno-based TCP that deploys active inline measurement. The ImTCP sender not only observes the ACK packet arrival intervals, but also actively adjusts the transmission interval of data packets, just as active measurement tools use probe packets. When the corresponding ACK packets return, the sender utilizes the arrival intervals thereof to calculate the measurement values. The measurement algorithm in ImTCP combines the available bandwidth and capacity measurement algorithms. The available bandwidth measurement algorithm utilizes Self Loading Periodic streams (SLoPS) proposed in [3]. However, SLoPS is changed so that the algorithm can be applied to inline measurement. The available bandwidth algorithm is described in detail in [4]. The measured values of available bandwidth are then used to supplement the packet pair technique to deliver a better capacity estimation than traditional packet pair based techniques.

We insert a measurement program into the sender program of TCP Reno to create an ImTCP sender. The measurement program is located at the bottom of the TCP layer. When a new data packet is generated at the TCP layer and is ready to be transmitted, the packet is stored in an intermediate FIFO buffer. The measurement program decides the time at which to send the packets in the buffer. The program waits until the number of packets in the intermediate buffer is sufficient to form a packet stream for available bandwidth measurment and a packet pair for capacity measurement, in each RTT. After sending packets required for measurement, the program then passes all data packets immediately to the IP layer while waiting for the corresponding ACK packets. The measurement program does not require any special changes in the TCP receiver program, except that an ACK packet must be sent back for each received packet. Therefore, delayed ACKs must be disabled at the TCP receiver; otherwise ImTCP will not perform measurement properly.

The principle of the packet-pair-based measurement technique for capacity is that, if the packet pairs are transmitted in a back-to-back manner at the bottleneck link (the link of smallest capacity bandwidth in the network path) and the time interval until they reach the receiver remains unchanged, then the capacity of the bottleneck link C (which is also the capacity of the network path) is calculated as:

$$C = \frac{P}{Gap} \tag{1}$$

where P is the packet size and Gap is the arrival time dispersion of the two packets at the receiver. The packet pairs are referred to as the C-indicator. Their time dispersion indicates the exact capacity value. If the packet pair is cut by packets from other traffic, then its dispersion can not be used to calculate capacity via Equation (1).

Current packet-pair-based measurement techniques have various mechanisms for determining C-indicators from packet pair measurement results. Some tools assume a high frequency of appearance of the C-indicator, and so search for the C-indicator from a frequency histogram (Pathrate [5]) or a weighting function (Nettimer [6]). CapProbe [7] repeatedly sends packet pairs until it discovers a C-indicator, based on the transmission delay of the packets. However, as shown in the following equation, when the available bandwidth is small, the C-indicator does not appear frequently. Thus, current existing tools may not discover the correct capacity.

Let δ be the time space of the packet pair when it arrive at the bottleneck link. We then assume that the links before the bottleneck link do not have a noticeable effect on the time space, so that δ is the approximate time interval in which the sender sends the packets. During the time of δ, the average amount of cross traffic that arrives at the bottleneck link, which is denoted as L, is

$$L = \delta \cdot (C - A) \tag{2}$$

where A is the available bandwidth at the time the packet pair is sent. We can see that when A is small, L is large, which means that the probability for a packet pair to pass the bottleneck link without being cut by the traffic of another packet is low. In other words, the available bandwidth of the path is an important factor in measuring the capacity. Based on the above observation, we develop a new capacity measurement algorithm, which exploits the advantage of awareness of the available bandwidth of ImTCP.

From Equation (2) we can estimate that the dispersion of the packet pair when leaving the bottleneck link is:

$$Gap = \frac{P + L}{C} = \frac{P + \delta \cdot (C - A)}{C}$$

Therefore, the capacity can be calculated as:

$$C = \frac{P - \delta \cdot A}{Gap - \delta} \tag{3}$$

There is a problem with current capacity measurement tools when every packet pair that passes the bottleneck link is cut by other packets, due to either a heavy load or constant and aggressive cross traffic at the bottleneck link. In this case, CapProbe will spend an extremely long time searching for C-indicators, and Pathrate and Nettimer will deliver incorrect estimations. Equation (3) introduces some important prospects, including ways to overcome the above problem:

– We can calculate the capacity bandwidth without the existence of C-indicators, assuming that the available bandwidth value is known.
– The measurement does not require δ as the smallest value that the sender can create. Any two TCP data packets that are sent in an appropriately small interval can be exploited for the calculation. This is a very important advantage because more data can be collected for the capacity search.
– We can discuss the statistical confidence of the measurement results based on the value of the variance of the calculated data.

We present a simulation of packet pair measurement as an example explaining Equation (3). We perform a simulation of packet pair measurements over 50 seconds on a

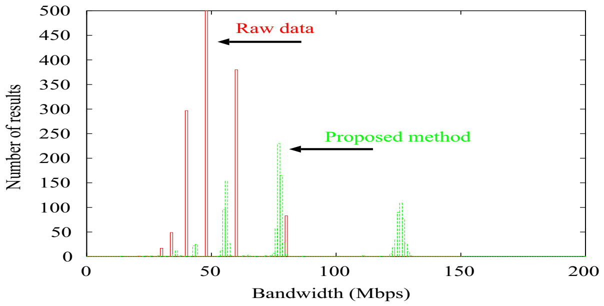

Fig. 1. Results calculated using the proposed Equation

network path for which the available bandwidth is 15 Mbps during the time. The background traffic is made up of an UDP packet flow. The UDP packet size is 500 KB. The correct capacity of the path is 80 Mbps. In Figure 1, the "Raw data" line shows the measurement results calculated using Equation (1), and the "Proposed method" line shows the results obtained by using Equation (3). We can conclude that Equation (3) provides a better result for capacity because the calculated data concentrate at the correct value of capacity (80 Mbps).

References

1. Cao Man, Go Hasegawa and Masayuki Murata, "Available bandwidth measurement via TCP connection," in *Proceeding of the 2nd Workshop on End-to-End Monitoring Techniques and Services E2EMON*, Oct. 2004.
2. M.Gerla, B.Ng, M.Sanadidi, M.Valla, R.Wang, "TCP Westwood with adaptive bandwidth estimation to improve efficiency/friendliness tradeoffs," *To appear in Computer Communication Journal*.
3. M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," in *Proceedings of ACM SIGCOMM 2002*, Aug. 2002.
4. Cao Man, Go Hasegawa and Masayuki Murata, "A new available bandwidth measurement technique for service overlay networks," in *Proceeding of 6th IFIP/IEEE International Conference on Management of Multimedia Networks and Services Conference, MMNS2003*, pp. 436–448, Sept. 2003.
5. C. Dovrolis and D. Moore, "What do packet dispersion techniques measure?," in *Proceedings of IEEE INFOCOM 2001*, pp. 22–26, Apr. 2001.
6. K. Lai and M. Baker, "Nettimer: A tool for measuring bottleneck link bandwidth," in *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, Mar. 2001.
7. R. Kapoor, L. Chen, L. Lao, M. Gerla and M. Sanadidi, "Capprobe: a simple and accurate capacity estimation technique," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2004.