Impact of the Ethernet Capture Effect on Bandwidth Measurements *

Mats Björkman and Bob Melander

Uppsala University, Information Technology, Dept. of Computer Systems,
Box 325, SE-751 05 Uppsala, Sweden.
{Mats.Bjorkman,Bob.Melander}@docs.uu.se

Abstract. In this paper we present impacts of the Ethernet capture effect on bandwidth measurements. We show that the unfairness caused by the Ethernet capture effect can have a severe impact on the ability to estimate available bandwidth. We define two metrics, surplus bandwidth and fair share bandwidth, that are protocol and application independent available bandwidth metrics. We propose a measurement method, the TOPP method, that measures link load with low risk of triggering an Ethernet capture effect. From TOPP measurements, we can estimate available bandwidth.

1 Introduction

Knowledge about the network bandwidth along a path from a source to a destination can be valuable for many applications that are able to adapt to changing network conditions. The bandwidth available from a source to a destination in a best effort network depends on the characteristics of the network path that the traffic travels. Not only static properties, such as the link speeds of the links forming the path, but also dynamic properties, such as competition with other traffic and changes in the routing topology are important. The dynamic properties can make the available bandwidth change dramatically over short periods of time.

In a network, we are likely to find shared-medium links such as Ethernet/802.3 links close to or local to the sender and receiver respectively. However, some of these links (notably the Ethernet/802.3), have an access method that is not fair in contention situations. This unfairness has been called the *Ethernet Capture Effect* [11]. The Ethernet capture effect makes estimation of available bandwidth hard.

In this paper we present results from experiments with an extension to the packet pair probing technique [6]. We call our probing method the *TOPP* method. The aim is to measure available bandwidth also on links with the Ethernet capture effect.

The contributions made are the following: We describe why measurement of available bandwidth is hard in Ethernet capture effect situations. We show how

* This work is supported in part by the Ericsson MARCH project.

G. Pujolle et al. (Eds.): NETWORKING 2000, LNCS 1815, pp. 156–167, 2000.
© Springer-Verlag Berlin Heidelberg 2000

the presence of the Ethernet capture effect can be detected in measurements. We propose a measurement method, TOPP, and two metrics that can be useful when estimating available bandwidth in Ethernet capture effect situations.

The paper is organized as follows. Related work is presented in Section 2 and section 3 presents the bandwidth definitions used in the paper. The Ethernet capture effect is then described in section 4. Section 5 describes our experimental setup. Section 6 presents how the link bandwidth can be found, and section 7 shows how the Ethernet capture effect can be detected. Section 8 presents the TOPP measurement method and how we analyze TOPP measurements to estimate available bandwidth. Finally, in section 9, conclusions are drawn and future work presented.

2 Related Work

Paxson [10] discusses methods for measuring bottleneck bandwidth. He makes a distinction between *bottleneck* [link] *bandwidth*, which is the data rate of the slowest forwarding element of a path, and *available bandwidth*, defined as the data rate at which a connection should transmit to preserve network stability.

The packet-pair method for estimating the bottleneck bandwidth has been studied by a number of researchers [2,6]. Carter and Crovella [3] propose two packet-pair based techniques, B-probe and C-probe, to measure bottleneck link bandwidth and available bandwidth, respectively. Both techniques rely on sending trains of ICMP echo packets and applying filtering methods to the measured inter-arrival times of the reply packets.

Weaknesses of the packet pair method, such as the problem with multichannel links, limitations due to clock resolution and out of order packet delivery are discussed by Paxson [10]. To deal with these short-comings, he introduces an extension to the packet pair technique called the PBM probing technique. In this, estimates for a range of probe train lengths (or bunch sizes) are formed and multiple bottleneck values are allowed. In [1], Allman and Paxson investigate methods for estimating the bandwidth that will be available to a new TCP connection, so that it immediately can begin sending data at that rate.

Lai and Baker [7] also suggest refinements to the packet-pair method. They propose to use a receiver-only method based on existing traffic together with a 'potential bandwidth filtering' scheme. In order to quickly adapt to bandwidth changes they introduce a packet window. When estimating the bandwidth, only packets that arrive within the time interval given by the window will be used.

Treno [8] is a tool that tries to measure the achievable TCP throughput between a pair of hosts. To achieve this, it simulates the full TCP algorithm including slow start. UDP packets with a suitably low time- to-live (TTL) value are sent to the destination, which will reply by sending back ICMP TTL exceeded packets. These are then interpreted as the equivalent of TCP's acknowledgements. Since Treno simulates the TCP algorithm, it normally needs 10 seconds or more to accurately estimate the available bandwidth.

The 'pathchar' [5] tool estimates the latency and link bandwidth of each hop along a network path by measuring the round trip times of packets sent by a single source. Pathchar uses a TTL-based technique, similar to that of Treno, to determine how many links the probe packets should traverse. A disadvantage of pathchar is that it consumes considerable amounts of bandwidth and that it is quite slow. Downey [4] suggests techniques to improve the accuracy of pathchar's estimates and to reduce the time required to generate them.

The Ethernet capture effect has been described in several papers [9,12,13]. Ramakrishnan and Yang [11] discuss its impact on UDP and TCP traffic and show that the capture effect degrades the performance of TCP. They propose a modified backoff algorithm, Capture Avoidance Binary Exponential Backoff (CABEB), that will overcome the capture effect.

3 Definitions

By *bottleneck link bandwidth* we mean the maximum transmission rate that can be achieved on the slowest link along a network path from sender to receiver. This is a well-established definition in the literature. *Available bandwidth*, on the other hand, is a less well-defined property. In this paper we have identified three metrics that can be used when characterizing the available bandwidth.

Since we do not know the details of the traffic along a network route, for instance whether traffic sources will adapt to accommodate new traffic sources (e.g. TCP), or not (e.g. UDP), we can only estimate upper and lower bounds on the bandwidth available to us. A lower bound (assuming adaptation from our side but not from the other traffic) would be the currently unused portion of the bottleneck link bandwidth. This is what we define as the *surplus available bandwidth* (or surplus bandwidth for short).

A usable upper bound on the available bandwidth would be our fair share of the bottleneck link bandwidth relative to the total traffic load offered. We define this bandwidth as the *fair share available bandwidth* (or for short fair share bandwidth). Naturally, the theoretical upper bound would be the link bandwidth. This bound could, for example, be approached by an aggressive UDP source or by a TCP source not adapting properly to congestion, when the remaining traffic consists of well-behaved TCP sources that all back off.

The actual bandwidth as perceived by a specific protocol and application should lie somewhere between the surplus bandwidth and the fair share bandwidth and this we define as the *protocol dependent available bandwidth*. In the rest of this paper we will refer to this simply as the available bandwidth.

4 The Ethernet Capture Effect

The Ethernet capture effect [12], is a term used to describe an undesirable side effect of the Ethernet CSMA/CD backoff algorithm. It is most apparent in a high load/few stations scenario and it results in transient or short-term unfairness. What this means is essentially that under high load, one station on a LAN can hold on to the channel and transmit several consecutive packets even though other station(s) are contending for access.

4.1 The Ethernet (802.3) Backoff Algorithm

The aim of the Ethernet medium access method is to give all stations fair access to the channel, i.e. there are no prioritized stations or classes of traffic.

Whenever a station has an Ethernet frame to send it checks if the channel is idle and in that case it attempts to transmit the frame. If other stations are trying to transmit at the same time then all stations will detect a collision. To arbitrate between the contending stations the Ethernet CSMA/CD protocol uses a backoff algorithm in which the currently offered load to the channel plays a central role. When the offered load is high, the stations should back off for longer times compared to when the load is low.

To estimate the instantaneous offered load, the stations associate with each frame a collision counter, n. It is initially zero and is then increased by one each time the frame experiences a collision. When a station has detected a collision it uses the collision counter to decide how many slot times, n_s, to wait before attempting to transmit again. This delay, n_s, is chosen as a uniformly chosen random integer in the range $0 \leq n_s \leq 2^k - 1$ where $k = \min(n, 10)$. If the frame experiences 16 consecutive collisions, the retransmission procedure is aborted and the frame is discarded.

The problem with this backoff scheme is that the station which is successful in sending its frame after a collision (i.e., the station that chooses the earliest slot no other station chooses) will start with a new frame having the collision counter set to 0. The other stations involved in the collision, on the other hand, will keep their collision counter values. As a result, if the successful station is involved in a new collision with stations involved in the previous collision, the successful station will choose its random wait time in a more narrow range than the other stations. This will increase its probability of being successful again in the new collision, and that increase of probability leads to the unfairness that is called the capture effect. For every consecutive collision that is won by the same station, the probability for that station to win again will increase and quickly tend towards 1 [11].

As the number of stations that share the channel increases, the capture effect will decrease. This is because the set of stations that are involved in consecutive collisions will vary more. Since every new station in the set will have a collision counter equal to 0 (i.e., the same value as the winning station has), they too will randomize in a narrow range after an initial collision. This will then effectively decrease the probability of the same station winning many consecutive collisions.

4.2 Impact on Active Probing Techniques

Most of the proposed techniques for actively measuring bandwidth are variants of the packet pair technique. There, the principle of the bottleneck spacing effect is used to estimate the bandwidth. That is, when two packets are transmitted back-to-back from sender to receiver they will be be spread out in time as they pass the bottleneck. When they exit the bottleneck this spreading will result in a time separation of the packets. That spacing between the packets is due to the transmission delay of the packets and possibly due to packets from competing traffic on the channel.

In the packet pair technique, one or several pairs of probe packets are sent from the source to the destination to measure the bottleneck or available bandwidth along that path. The bandwidth estimate, $\tilde{b}$, is then calculated as

$$\tilde{b} = \frac{s}{t_2 - t_1} = \frac{s}{\Delta t} \tag{1}$$

where s is the size of the probe packets and t_1 and t_2 are the arrival times of the two packets. Since s and the resolution by which Δt can be measured set a limit on the largest bandwidth that can be measured, it is desirable that s is as large as possible. Furthermore, if the interval at which the probe packets are sent, Δt_s, is larger than the interval corresponding to the bottleneck bandwidth, then the bottleneck spacing effect will not be noticed. For that reason, the probe packets should be sent as close to back-to-back as possible.

When the goal is to measure bottleneck *link* bandwidth, packets from competing traffic can be regarded as noise as these may compress or extend the time separation between the probe packets. To get a good estimate of the link bandwidth, that noise must be filtered out from the time measurements. If instead the goal is to measure the *available* bandwidth, the packets from the competing traffic are not noise since it is the competing traffic that makes the bottleneck available bandwidth deviate from the bottleneck link bandwidth. In order for the packet pair estimate $\tilde{b}$ to accurately estimate the available bandwidth, it is important that packets from the competing traffic interleave with the probe packets in proportion to the channel load. However, this requires that the medium access method is fair.

Since the Ethernet capture effect alters the fairness of access to the channel it also affects the packet pair probing techniques. When measuring bottleneck *link* bandwidth it will work to our advantage since the probe packets are likely to block out packets from the competing traffic. However, when measuring *available* bandwidth the capture effect will affect the packet pair techniques negatively. This is because the packets from the competing traffic will not interleave with the probe packets in proportion to the load on the channel, due to the blocking nature of the capture effect.

C-probe Being a variant of the packet pair probing technique, C-probe [3] will be affected by the capture effect for the same reasons as explained above. In C-probe, the goal is to measure available bandwidth and this is done by sending trains of ICMP echo packets to the destination host. The probe packets (i.e., the ICMP packets) in the train are of equal size and are sent as close to back-to-back as possible.

The available bandwidth calculation is based on equation (1), when t_1 and t_2 are the arrival times of the *first* and the *last* packet in the train. As the probe packets are sent as close to back-to-back as possible, a queue will quickly build up in a contention situation. Now, because of the capture effect there is a high probability that several packets in that queue will traverse the channel without being interleaved with packets from competing traffic. As a consequence, the bandwidth estimate given by equation (1) will be too high.

5 Experimental Setup

All the results we present in this paper are based on experiments done on a 10 Mbps Ethernet LAN, using a number of Sun Sparcstations 4 running Solaris 2.6 and a number of Intel-based PC's running Linux (kernel versions 2.0.35 and 2.2.3). The Sparcstations are all equipped with an Am7990 (Lance) based Ethernet interface whereas the different Linux hosts use a variety of network interfaces (3Com Etherlink III, D-Link DFE-530TX and SMC EtherEZ).

6 Finding the Link Bandwidth

We want to estimate the link bandwidth of the bottleneck link. Packet pair probing is a technique that can be used for this. Instead of using a number of pairs, we extend the ideas of B-probe [3] and send a long train of back-to-back packets through the network. Here, the Ethernet capture effect can actually help us when trying to establish the link bandwidth of the bottleneck link. Should the bottleneck link be an Ethernet, the Ethernet capture effect will affect our bandwidth measurements as has been explained earlier. Using a long train of packets will increase the possibility that we will see a capture effect if there is one. When analyzing packet pair inter-arrival times, times reflecting the link bandwidth will (as explained earlier) be over-represented as compared to a link without any capture effect.

As can be seen in figure 1, packets from a train moving through a saturated Ethernet will exhibit an extremely bursty behavior, with several long bursts at, or near in the case of contention slots passing, link bandwidth speed. Should this Ethernet be the bottleneck link, due to the capture effect the link bandwidth can many times be estimated without need for advanced filtering (in the example in figure 1, a median value would be sufficient).

7 Detecting the Ethernet Capture Effect

On a shared-medium link with (short-term) fair resolution of contention, the probability of bursts of frames at link speed will rapidly decrease with the burst length. Consider two stations competing for the medium. With a fair contention resolution, the probability of a certain station winning one collision would be 0.5, the same station winning two successive collisions would be 0.25, three successive collisions 0.125 etc., yielding a mean expected burst length of 2.

On an Ethernet, however, the capture effect will favor the station already sending when this station has a queue of frames to send. Once a competing station has started losing contention resolutions against a station with a queue of frames, chances are (as explained earlier) that it will continue to lose as long as the winning station has frames queued up for sending.

In fact, on a saturated Ethernet with only two senders, once a station has started losing, the sum of probabilities that it will get to send the frame at all before giving up (after 16 successive collisions) can be calculated to less than 50%. (The sum of probabilities that the loser of collision 1 will win any of the collisions 2, 3, ... up to 16 amounts to between 40% and 50%, dependent on the number of ties at contention resolutions). Thus, on a saturated Ethernet with few sending stations, the probability of seeing long bursts will be enormously increased as compared to a fair medium (above). In the saturated two-station example, once queues have built up, the median burst length (counting bursts of size 1 and longer) is 16, since the probability is less than 50% that the loser will be able to break in during exponential backoff, but once the losing station discards the losing frame, its next frame will have a fresh (zero) collision counter, and the losing station stands a 50% chance to win the 17th collision and thus end the burst from the other station. See also the sample train in figure 1.

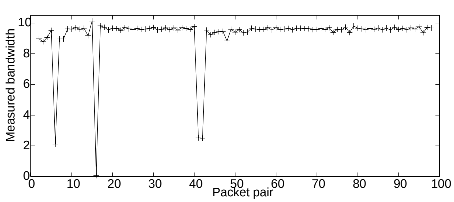

Fig. 1. Packet bursts for packets from the probing station. 10 Mbps link, one station is generating 7 Mbps of CBR cross traffic.

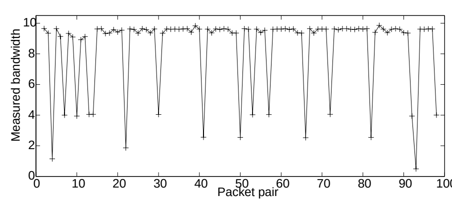

Fig. 2. Packet bursts for packets from the probing station. 10 Mbps link, six stations are together generating 7 Mbps of CBR cross traffic.

Hence, we can detect the Ethernet capture effect by examining the distributions of burst lengths. In figure 1, we saw a typical example of burst lengths when we have two competing stations. As the number of competing stations increase, the capture effect (in the sense that one and the same sender gets to send again and again) will decrease in significance [1]. However, still at seven competing stations (figure 2), we see this over-representation of burst lengths, albeit with shorter burst lengths than in the two-station case shown in figure 1.

The conclusion from this section is the following: as long as we have an Ethernet capture effect significant enough to cause an over-representation of long link bandwidth bursts (effectively destroying our possibility to use packet trains to measure available bandwidth), this over-representation can be detected by examining the burst length distribution. Once we have detected that we indeed

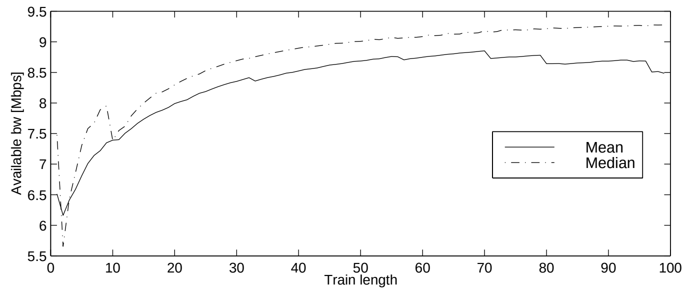

Fig. 3. Measured available bandwidth using packet trains. One station is generating 7 Mbps of CBR cross traffic.

have a capture effect situation, we cannot use a train method, but must use other methods to estimate available bandwidth, see below.

8 Measurements in an Ethernet Capture Effect Situation

As explained above, packet trains will over-estimate available bandwidth in a capture effect situation. Figure 3 shows the available bandwidth as estimated by packet trains on a 10 Mbps Ethernet. The single cross-traffic source offers 7 Mbps, so surplus bandwidth is 3 Mbps. The fair share bandwidth at the 10 Mbps offered by the probe source would be $10/(7+10)*10 = 5.9$ Mbps. As can be seen, the packet train method over-estimates available bandwidth far beyond the fair share bandwidth. Hence, we must instead try to estimate traffic characteristics without triggering any capture effect.

8.1 Measuring Without Triggering the Capture Effect

The long bursts typical for the Ethernet capture effect occur when the sender has a queue of frames to transmit. Hence, we should measure bandwidth without causing queues to build up. One way to avoid this is to measure bandwidth by sending separated packet pairs, where the pairs are so well separated that any disturbances to the network caused by one pair (e.g. queues caused by temporary overloading of the link) have been normalized before the next pair is sent.

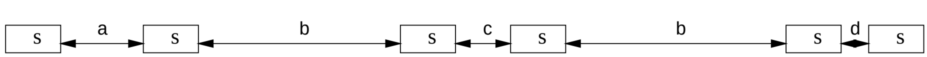

Fig. 4. Parts of a TOPP train. The equally sized probe packets s are sent as pairs with decreasing intra-pair spacing, i.e, $d < c < a$. The spacing between the pairs are chosen randomly in the range $[b-\epsilon, b+\epsilon]$ where ϵ is a small number. For each intra-pair spacing, a, c, etc, n pairs are sent.

Instead of directly trying to measure available bandwidth, we try to find the surplus bandwidth of the bottleneck link. From that and from knowledge of the link bandwidth (measured e.g. using methods described earlier), we can estimate lower and upper bounds of available bandwidth using the methods explained above.

8.2 The TOPP Measurement Method

When trying to estimate the surplus bandwidth, we send a sequence of packet pairs, where the pairs are well separated in order to minimize the Ethernet capture effect. The intra-pair spacing of the packet pairs is decreased in known steps in order to momentarily offer different loads. In the presented measurements we have used steps of 1 Mbps, i.e. the first run of pairs have an intra-pair spacing corresponding to a momentary load of 1 Mbps, the next to 2 Mbps etc. We call this method the *TOPP* method (short for Trains Of Packet Pairs). Figure 4 illustrates this.

Using the TOPP method, we can analyze the effects of a particular offered load without causing long-term disturbances to the network (e.g. without triggering any long-term Ethernet capture effects), disturbances that had been caused had we actually offered that same load for any longer period.

Since the TOPP method is independent of the inter-pair spacing, this spacing can be large for links loaded close to saturation in order to minimize the impact of probe traffic. As an example, assume an inter-pair spacing of 100 ms. For a momentary load of 5 Mbps, we send two frames (each takes 1.1 ms to send) with a 1.1 ms intra-packet space and then wait 100 ms before the next pair. The real load on the link would then be $(1.1+1.1)/(1.1+1.1+1.1+100) = 0.022 = 2.2\%$ of 10 Mbps, i.e. 210 kbps, while we can estimate the effects of a 5 Mbps load.

For the TOPP measurements presented below, the inter-pair spacing has been 20 ms, giving real loads of 670 to 970 kbps while offering momentary loads of 1 to 10 Mbps. Since our experiments have controlled cross-traffic, we know that this spacing will not cause link saturation. For measurements on links with unknown traffic, we recommend longer inter-pair spacing.

By using the TOPP method of “measuring load without loading”, we can find the surplus bandwidth of an Ethernet by analysis of TOPP reception times as the next section describes.

8.3 Analysis

When analyzing TOPP reception times, we have used two interesting metrics.

First, as long as our offered load does not saturate the link, we will see a measured bandwidth that is close to the offered. As the link saturates, i.e. our offered bandwidth has reached the surplus bandwith, the measured bandwidth will start deviating from the offered bandwidth. This point can, in most measurements we have performed, easily be detected. An example is seen in figure 5.

Second, as we cross the link saturation point, the standard deviation in our measurements will increase rapidly, since on a saturated link we will see com-

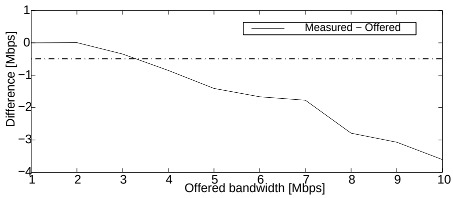

Fig. 5. TOPP measured surplus bandwidth using the *measured - offered* metric. 10 Mbps link, 7 Mbps of CBR cross traffic.

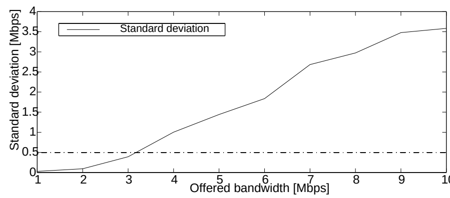

Fig. 6. TOPP measured surplus bandwidth using the *standard deviation* metric. 10 Mbps link, 7 Mbps of CBR cross traffic.

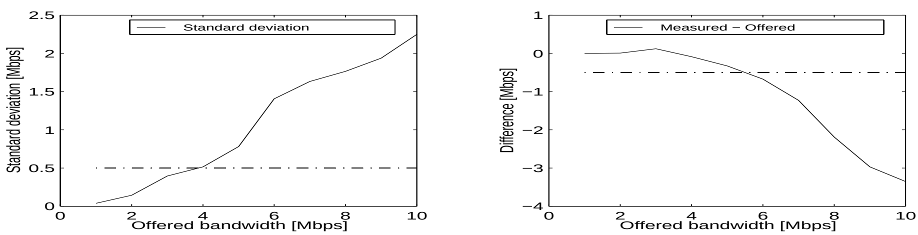

Fig. 7. TOPP measured surplus bandwidth using both metrics. 5 Mbps of VBR cross traffic.

peting traffic at virtually every slot, causing our measured bandwidth to either jump up (should our first frame be delayed and both probe frames then be sent more closely spaced than offered), or jump down (should one or more competing frames be sent in between our probe frames), but seldom would both probe frames cross the link undisturbed and the measured bandwidth be the offered. As seen in the example in figure 6, there is a sharp increase in standard deviation as we reach link saturation.

Below we present figures of the measured and offered bandwidth difference and the bandwidth standard deviation for measurements with varying cross traffic bandwidths and cross packet sizes. We have used a 0.5 Mbps difference as the limit where we consider saturation to have occurred, and a 0.5 Mbps standard deviation as the standard deviation limit. As shown below, the surplus bandwidth can in most of our measurements be estimated quite accurately.

Figure 7 shows results from a measurement with varying cross-traffic frame sizes (as opposed to the fixed size CBR cross-traffic from figures 5 and 6). In this measurement, the offered cross traffic is 5 Mbps. As can be seen, both measures seem to give good estimations of the surplus bandwidth.

Figure 8 shows estimations when having 5 Mbps CBR cross traffic, and figure 9 estimations for 3 Mbps CBR cross traffic. For both these cross-traffic bandwidths, our measures give good estimations of surplus bandwidth.

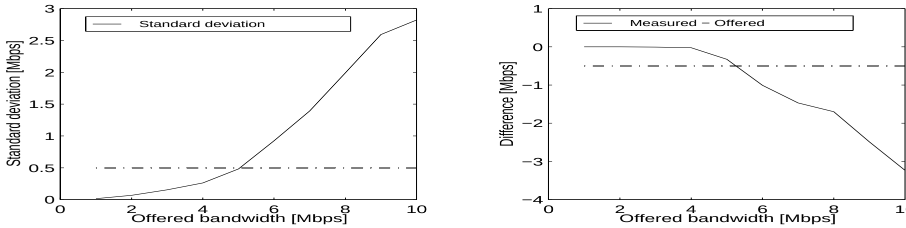

Fig. 8. TOPP measured surplus bandwidth using both metrics. 5 Mbps of CBR cross traffic.

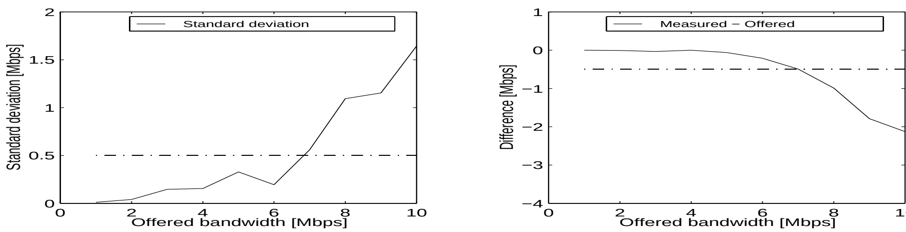

Fig. 9. TOPP measured surplus bandwidth using both metrics. 3 Mbps of CBR cross traffic.

With knowledge of the surplus bandwidth and link bandwidth, we can calculate the cross-traffic bandwidth. Knowing this, we can for a given offered load calculate what our fair share of the link bandwidth would be.

For example, if the surplus bandwidth is 3 Mbps and the link bandwidth is 10 Mbps, we know that cross traffic loads the link with 7 Mbps. Should we offer 7 Mbps ourselves, our fair share of the 10 Mbps link would be $(7/(7+7)) * 10 = 5$ Mbps. Hence, should our application initially offer 7 Mbps to this link, our perceived available bandwidth would likely be not less than the surplus bandwidth of 3 Mbps and not more than the fair share bandwidth of 5 Mbps.

9 Conclusions and Future Work

From the work presented in this paper, we draw several conclusions:

When estimating available bandwidth, two reasonable protocol-independent lower and upper bounds are surplus and fair share bandwidth, respectively.

Trains of back-to-back packets cannot be used to accurately measure available bandwidth in Ethernet capture effect situations. This means that methods like C-probe do not give accurate estimations in these situations, rather will they over-estimate the available bandwidth.

Burst length analysis can detect an Ethernet capture effect situation.

The presented TOPP method of injecting well-separated packet pairs with decreasing intra-pair spacing can be used to measure bandwidth without causing long-term disturbances to the link.

When estimating surplus bandwidth, the difference between offered and measured bandwidth, and the standard deviation of the measured bandwidth, are two metrics that are likely to be useful in the analysis.

Future work include multi-hop measurements and the adaptation of our methods to other shared-media links such as wireless links.

References

1. Mark Allman and Vern Paxson. On Estimating End-to-End Network Path Properties. In *SIGCOMM '99 Conference Proceedings*, pages 263–273, Cambridge, MA, USA, August 31–September 3, 1999. ACM SIGCOMM Computer Communication Review, 29(4).
2. Jean-Chrysostome Bolot. End-to-end packet delay and loss behavior in the internet. In *SIGCOMM '93 Conference Proceedings*, pages 289–298, San Francisco, CA, USA, September 13–17, 1993. ACM SIGCOMM Computer Communication Review, 23(4).
3. Robert L. Carter and Mark E. Crovella. Measuring bottleneck link speed in packet-switched networks. Technical Report TR-96-006, Boston University Computer Science Department, Boston, MA, USA, March 1996.
4. Allen B. Downey. Using pathchar to estimate Internet link characteristics. In *SIGCOMM '99 Conference Proceedings*, pages 241–250, Cambridge, MA, USA, August 31–September 3, 1999. ACM SIGCOMM Computer Communication Review, 29(4).
5. Van Jacobson. Pathchar - a tool to infer characteristics of Internet paths. Presented at the Mathematical Sciences Research Institute (MSRI); Slides available from ftp://ftp.ee.lbl.gov/pathchar/, April 1997.
6. Srinivasan Keshav. A control-theoretic approach to flow control. In *SIGCOMM '91 Conference Proceedings*, pages 3–15, Zürich, Switzerland, September 3–6, 1991. ACM SIGCOMM Computer Communication Review, 21(4).
7. Kevin Lai and Mary Baker. Measuring bandwidth. In *Proceedings of IEEE INFOCOM '99*, New York, USA, March 21-25, 1999.
8. Matt Mathis and Jamshid Mahdav. Diagnosing internet congestion with a transport layer performance tool. In *INET '96*, Montreal, Canada, June 24–28, 1996.
9. Mart L. Molle. A New Binary Logarithmic Arbitration Method for Ethernet. Technical Report CSRI-298, Computer Systems Research Institute, University of Toronto, Toronto, Canada, April 1994.
10. Vern Paxson. End-to-end internet packet dynamics. In *SIGCOMM '97 Conference Proceedings*, pages 139–152, Cannes, France, September 14–18, 1997. ACM SIGCOMM Computer Communication Review, 27(4).
11. K. K. Ramakrishnan and Henry Yang. The Ethernet Capture Effect: Analysis and Solution. In *Proceedings of IEEE 19th Conference on Local Computer Networks*, MN, USA, October, 1994.
12. Rich Seifert. The Effect of Ethernet Behaviour on Networks using High-Performance Workstationss and Servers. Technical Report, Networks and Communications Consulting, Cupertino, CA, USA, March 1995.
13. Brian Whetten, Stephen Steinberg and Domenico Ferrari. The Packet Starvation Effect in CSMA/CD LANs and a Solution. University of California at Berkeley.