Network Independent Available Bandwidth Sampling and Measurement

Manthos Kazantzidis, Dario Maggiorini, and Mario Gerla

Computer Science Department, University of California, Los Angeles, CA 90095
{kazantz,dario,gerla}@cs.ucla.edu

Abstract. Available bandwidth knowledge is very useful to network protocols. Unfortunately, available bandwidth is also very difficult to measure in packet networks, where methods to guarantee and keep track of the bandwidth (eg, weighted fair queuing scheduling) do not work well, for example the Internet. In this paper we are dealing with an available bandwidth sampling technique based on the observation of packet time dispersion in a packet train or pair. In standard techniques the available bandwidth is sampled by using a "bytes divided by dispersion" (or "bytes over time", BoT) calculation and then filtered. This method of calculating available bandwidth samples has been used in all packet dispersion related work. We propose a new sampling method of available bandwidth called ab-probe. The ab-probe method uses an intuitive model that helps understand and correct the error introduced by the BoT sample calculation. We theoretically compare the new model with the previous one, exploring their differences, observability and robustness. We argue that the model may significantly improve protocols that can use an available bandwidth measurement, in particular transport-level protocols that currently use the BoT calculation.

1 Introduction

Let us define the available bandwidth over one link as the bottleneck link bandwidth minus the used bandwidth, i.e. the un-utilized bandwidth. A path's available bandwidth is the minimum available bandwidth across all links in the end-to-end path. If the latter could be known to end hosts, it could be used to apply efficient network and flow control, for example, by means of multimedia content adaptation [8], application/web routing and dynamic server selection [1], congestion control transports [6]. At the same time it is hard to measure available bandwidth. It is a highly variable quantity and constrained to an end-to-end observation, as the Internet Protocol scalable architecture dictates. Another restriction, which we will attempt to relax, is that current "available bandwidth" techniques assume that the network is performing weighted fair queuing on its flows [7]. This is the only case where the rate observed at the receivers may be the path available bandwidth and can be calculated as the bytes over the time dispersion of two or more successive packets (from now on we will refer to this scheme as Bytes over Time or "BoT approach"). The above assumption holds in

M. Ajmone Marsan et al. (Eds.): QoS-IP 2003, LNCS 2601, pp. 117–130, 2003.

a Diff Serv environment, with well defined traffic and service classes, and with WFQ routers. The Internet today, however, cannot classify/distinguish flows, may employ a variety of queuing disciplines and currently has pre-dominantly FIFO routers. It is known that the WFQ model assumption often leads to relatively high bandwidth estimation errors especially at high loads. Unfortunately, this is exactly when available bandwidth information is most essential. We show an alternative sampling method that assumes no specific queuing discipline and therefore does not contain the error caused by the above assumption.

An "active" heuristic technique called cprobe [1] is widely used to measure available bandwidth. It was developed for server selection purposes, and it is the simplest example of active BoT sample calculation. "Passive" measurements of available bandwidth using packet dispersion techniques are also found in transport level. For example, the rate estimation proposed in TCP Westwood [6] reminds us of the packet pair bandwidth estimation technique. Likewise, Microsoft's video streaming protocol MSTFP employs bandwidth estimation concepts [8].

Packet pair techniques can measure the bottleneck link bandwidth in FIFO queuing networks by using the BoT model and a combination of techniques such as kernel density estimation and Potential Bandwidth Filtering, as implemented in nettimer and proposed in [2]. In this work a kernel density estimator is used to find the highest density point by effectively dealing with well-known problems arising from constant bin size histograms. The basic assumption, under which this estimation is correct, is that packet pair results tend to aggregate around the bottleneck link bandwidth . This work has progressed from [3] and has also produced packet tailgating [4]. The latter is a technique that uses a train of packets to measure multiple bandwidths along a path. Unfortunately, few applications are directly interested in bottleneck link bandwidth. A bottleneck link bandwidth measurement is needed however for all current available bandwidth estimation techniques.

In [5] it is shown that packet dispersion techniques converge to an Asymptotic Dispersion Rate that is lower than the capacity but is not quite the available bandwidth in the path. The method in [5] investigates the effects of cross traffic on packet dispersion in the Internet. Our work is in agreement with that study, It proposes similar models, but it uses a sample calculation method instead of the BoT method that was the basis of that work as well.

In essence, all of these schemes have in common the same probing technique, ie packet pair or train. The difference is in the filtering method. For example, when a packet dispersion based scheme is used for available bandwidth estimation, the filter may average the samples [1], discover modes [5] or discover density points [3]. In all mechanisms used so far, the samples are computed using the simple BoT formula. The main contribution of this paper is to use a different equation for computing bandwidth samples. Our equation is independent of a specific queuing discipline. It is slightly more complex than simple division, but it produces much more accurate samples, as we will show in the sequel.

In this paper, after presenting our model, we theoretically explore its robustness and explain the way it differs from the BoT model. We then develop an "active" measurement technique that uses our sample calculation in packet pairs and then in trains. Next to it, we also develop a "passive" measurement technique, again using both packet pairs and trains. We consider the passive measurement especially important as it may be used in end-to-end transports to aid in flow control. We use real MPEG and H263 video traces to provide the packets for the passive measurement. We extensively test and compare our model to the BoT model in various Internet scenarios. All the experimental results support our theoretical claims and show that our sampling technique improves upon the BoT method, especially when the network load is high.

In the next section we analyze the packet pair behavior link by link. In Section 3 we discuss the need for a more sophisticated sampling technique, present our general model for the packet train behavior related to the available bandwidth, use our model to study the BoT problems and finally develop an active and a passive measurement. In 4 we present the real network experiments, which are in keeping with the theoretical models and provide insight on how effective is the new measurement based on our model. We conclude the paper in Section 5.

2 Packet Pair Background

In this section, we qualitatively analyze the interference that an end-to-end packet pair based measurement may experience from cross traffic. Our goal is to understand the source of noise and errors in a packet pair measurement, especially on an available bandwidth measurement that we are dealing with here.

Let us focus separately on three segments of the path: 1) the path consisting of links from the source up to the bottleneck link, 2) the bottleneck link, 3) the path from after the bottleneck link to the destination. When we refer to a link we refer to the outgoing queue. Let us assume:

- $N-1$ links on the path, with the bottleneck at link k
- Pb_i, the potential bandwidth upon entering the link i, determined by the pair dispersion after link i ; Pb_0 is then the pair potential bandwidth at the sender, defined as the payload bytes over the initial time separation.
- B_i, the actual bandwidth of link i
- $Pb_0 > B_k$

Then we can analyze in details the behavior of the three segments.

The first segment of the path is interesting because it ultimately defines the potential bandwidth (Pb_k) with which the pair will enter the bottleneck link k. In order to make sure that we can measure the bottleneck link, Pb_k must be greater than the bottleneck link bandwidth. After each link the pair may decrease its potential bandwidth when $Bi < Pb_{i-1}$; the same can happen when cross traffic arrives in the time between the pair arrivals (Note that the probability of this event occurring is inversely proportional to Pb_{i-1} for a given network load). When enough cross traffic is queued already when the packet pair arrives then we may get useful time compression. It is called time compression because the time separation of the packet pair is potentially compressed (it becomes the payload of one packet over the link bandwidth). We call it "useful" because it increases the pair's potential to correctly measure large bottleneck bandwidths. Note that in the case of available bandwidth measurement this compression in not useful.

The packet pair arrives to the bottleneck link with a potential bandwidth of Pb_{k-1} and exits with Pb_k, hopefully equal to B. If Pb_{k-1} is less than B, then we already have an underestimate of B. This underestimate may be further increased by intercepting cross traffic. If Pb_{k-1} is more than B, then the exiting potential bandwidth is either B or some overestimation due to compression because of preexisting traffic.

Other events may occur after the bottleneck link. As the pair is traveling after the bottleneck link it should sustain its time separation. However, enough intercepting cross traffic (i.e. traffic that enters the node and gets queued in the middle of the packet pair interval) may cause further underestimation over a link. Traffic queued in front of it may cause time compression (if service/transmission time for the packets queued in front is more than the packet pair time separation). This is an adverse time compression that causes over-estimation of the bottleneck link bandwidth.

3 Extension of Packet Pair/Train for Available Bandwidth Sampling

3.1 Why Do We Need a New Method to Calculate the Samples?

Let us attempt to calculate what the available bandwidth over a single link should be and how this is different than the BoT calculation. We first illustrate the BoT method. Consider two packets, Packet 1 and Packet 2, each of size S, entering the bottleneck link with a potential bandwidth of Pb_{k-1} which is close to B, the bottleneck link bandwidth. The time that Packet 1 exits from the link, ie, exit from service, marks the beginning of time interval and Packet 2's exit from service marks the end of the interval, denoted as time separation d. The BoT method calculates the available bandwidth A at S/d.

Now assume that A is indeed the available bandwidth seen by those packets. Then, the consumed bandwidth is obviously $B - A$. Let us call the interval over which the available bandwidth A is measured the sampled interval. This interval can be defined, for example, by the separation of the packets, from the time the Packet 1 enters the queue until Packet 2 enters the queue, which is by definition S/Pb_{k-1}.

Since during that time the bandwidth consumed (by other connections) is $B-A$ then the number of bits that enter the link queue during the sampled interval is $(B - A) \cdot S/Pb_{k-1}$. The observed separation at the link exit, as defined in the previous paragraph, will be the transmission/service time of the intervening bits that have entered the queue after Packet 1, and the transmission/service of

Packet 2. Therefore $d = (B - A) \cdot (S/Pb_{k-1})/B + S/B$. It is obvious then, by solving for the available bandwidth A, that A is not generally S/d as the BoT calculation dictates. We elaborate on their difference in the rest of the paper and attempt to develop a new sampling method based on it.

3.2 The Ab-probe Model

In this paragraph we use an approach similar to that described in 3.1. to calculate the available bandwidth samples from a packet train of N equally sized and spaced in time packets. Assume N $(N \geq 2)$ equally spaced packets each of size S bits. Assume the packets reach the bottleneck link with a potential bandwidth of Pb, that the bottleneck link bandwidth is B and the actual available bandwidth (during the train transit) at the bottleneck link is A, as before. We are calculating the available bandwidth from the perspective of an application, without counting the probing packet overhead. The sampled interval (see 3.1) is:

$$Sampled\ Interval = \frac{(N-1)S}{Pb} \tag{1}$$

If the available bandwidth during this interval is A then, using (1), the traffic (in bits) the queue should be receiving from other flows during this interval is:

$$Cross\ Traffic\ bits = \frac{(B-A)(N-1)S}{Pb} \tag{2}$$

Then the observed time dispersion (separation) between Packet 1 exiting service and Packet N exiting this link's service is the transmission time of the packet train bits and the intervening traffic bits through the link.

$$\begin{aligned} T = T_N - T_1 &= \tfrac{(N-1)S}{B} + \tfrac{(B-A)(N-1)S}{Pb \cdot B} \\ &= \tfrac{(N-1)S}{B}\left(1 + \tfrac{B-A}{Pb}\right) \end{aligned} \tag{3}$$

Solving for the available bandwidth A we get:

$$A = B - \frac{B \cdot T - (N-1)\,S}{(N-1)\,\frac{S}{Pb}} \tag{4}$$

Note that an available bandwidth sample, averaged over the interval sampled by the packet train, may be negative. This is intuitive and is consistent with available bandwidth definition. It may be negative when, during a transient, period more traffic enters the link than the total bandwidth can support. (More traffic than the link's capacity may enter a link for a short interval because of the queue buffer space). Averaged over longer period of times the available bandwidth with samples given by these equations will turn out to be positive. The above model can be used (i) to help understand the error when sampling available bandwidth using the BoT model in non-WFQ networks. (ii) to develop a network independent sampling method. We study these issues further.

3.3 The BoT Sampling

In this paragraph we look at the model used so far in measuring available bandwidth that uses the BoT sample calculation. Equation 4 gives the correct available bandwidth averaged over a period of time (defined by a train of packets). This equation is to be compared with the BoT calculation $(N-1) \cdot S/T$.

First, we notice that following the above logic, the BoT calculation gives a different available bandwidth result for different size trains. This effect has been in fact noticed in real testbed experiments with cprobe [1]. Figure 1 shows that applying this equation has the exact error effect shown experimentally in [1]. A train of 4.1 Mbps potential bandwidth (i.e. the potential (sender) bandwidth of the packet pairs in the train) is used over a 4 Mbps capacity link and 2 Mbps available bandwidth. The equations of the model are applied to calculate the actual available bandwidth using different values for N, the number of packets in the packet train.

Figure 2 and Fig. 3 show the relative error for the BoT calculation used over a 4 Mbps link and trains of 10 packets and 160 bits each packet. Note that the BoT error becomes exponentially larger as the network gets congested. Specifically it reaches a 100% (relative error) when the 4 Mbps link has 1.28 Mbps available bandwidth. When the network is in a very congested state, and the link has only 100 Kbps available bandwidth, out of its 4 Mbps, we have a 2000% relative error.

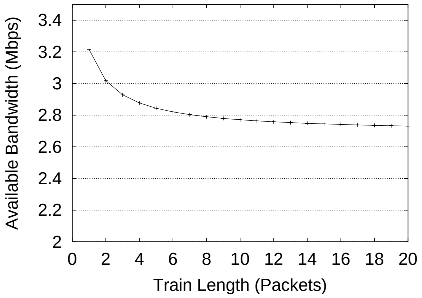

Fig. 1. BoT available bandwidth calculation varies with train size under same network conditions.

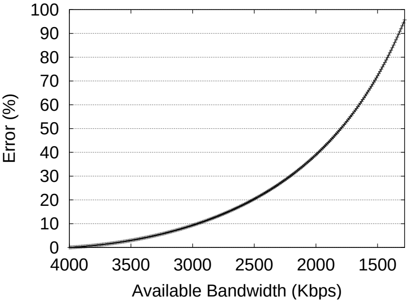

Fig. 2. The BoT relative error for a 4 Mbps link with 4 Mbps up to 1.280 Mpbs available.

3.4 Observability and Robustness of Ab-probe

The ab-probe (available bandwidth probe) model just described becomes a viable measurement method, if B and Pb can be accurately estimated. The measurement can be implemented both as an end-to-end measurement and as a network

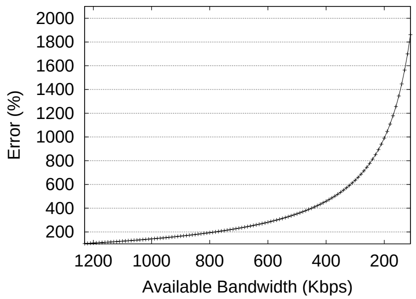

Fig. 3. The BoT relative error for a 4 Mbps link with 1.230 Mbps to 100 Kbps available.

layer measurement in network feedback architectures, in both active and passive forms.

In the network layer case, the measurement can be done link by link, thus *Pb* is known and *B* can be very easily estimated over each link [3].

In an end-to-end measurement, the bottleneck bandwidth (as opposed to available bandwidth) can be estimated accurately by proper filtering of the BoT samples as shown in the nettimer method [3]. The above method relies on the fact that with non zero probability the two packets in the pair are transmitted one immediately after the other by the bottleneck link. Thus, the kernel density estimator can identify the bottleneck link "mode" of the samples and compute the corresponding value. The "potential bandwidth" (*Pb*) at the entry of the bottleneck link however is only known to be between the sender bandwidth (assumed larger than the bottleneck bandwidth) and the bottleneck link bandwidth in an end-to-end measurement. This is because, when distorting events do not occur, specifically the time compression before the bottleneck link, then the sender bandwidth cannot increase. It may only decrease by entering through smaller capacity links. If we sent at a sender (potential) bandwidth of the bottleneck link capacity then with no distortions by definition the *Pb* will be *B*. In Fig. 4, we see how possible distortions in the *Pb* estimation affect the available bandwidth measurement. The graph is based on the equations presented by introducing errors in the *Pb* estimation. It shows that when the actual *Pb* is lower than the estimation, the available bandwidth estimation becomes negative, a 50% underestimation in *AB* will result from a 35% underestimation in *Pb*. When *Pb* is estimated higher than the actual *Pb* then the error in available bandwidth is practically bounded by a 50% overestimation. Since it is preferable to have an overestimation we sent at a sender bandwidth of slightly more than *B* as mentioned before. Very negative available bandwidth samples can then be filtered out and not be accounted for in the measurement. Next we show how the above can be successfully put into practice.

As the packet train travels towards its destination it undergoes the distortions we described in the packet pair section and appropriate filtering of the resulting samples should be used.

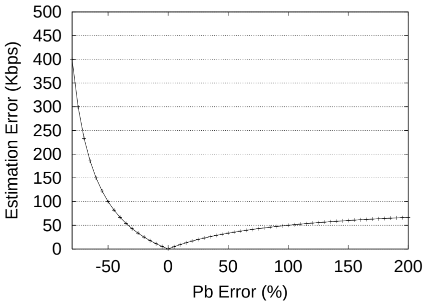

Fig. 4. How an error in *Pb* estimation affects the available bandwidth sampled with ab-probe.

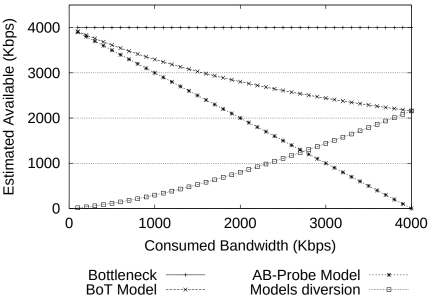

Fig. 5. Analytical Difference between BoT and ab-probe measurement.

4 Experiments

We have run real experiments using long range Internet connections and short range campus Internet connections. The experiments are performed using the active and passive measurement and testing both options of packet pairs or trains. In all cases we have found that practice agrees with theory.

The short range Internet topology is shown in Fig. 6 The two LANs are connected by a Cisco 2600 router in two 10 Mbps interfaces. On each LAN one host is performing the "probing" and another is used for injecting extra cross traffic. Normal traffic exists in the network as usual, but we inject the extra traffic so that we may observe how the measurements react to it. The long range Internet topology (Fig. 7) has the two source and destination LANs more than 20 hops apart. The sending hosts (probing and cross traffic) are switched through a Cisco Catalyst switch series 6509, located in California. The two receiving hosts are connected to a Cabletron SmartSwitch 2100, located in Italy.

The probing connection is either active or passive and uses the appropriate time-stamping in the packet headers as required for the measurement. We use application level timestamps as a measurement application or an adaptive multimedia streaming application would have to. We use the nettimer tool to compute the bottleneck link bandwidth just prior to the experiment. The active measurement uses 128 byte packets changing the inter-packet delay to achieve the required sender bandwidth. In the case of packet pairs, 40 packet pairs (80 packets) are sent at the required bandwidth following a one second idle interval. When packet trains are tested we sent 10 trains of 8 packets following a one

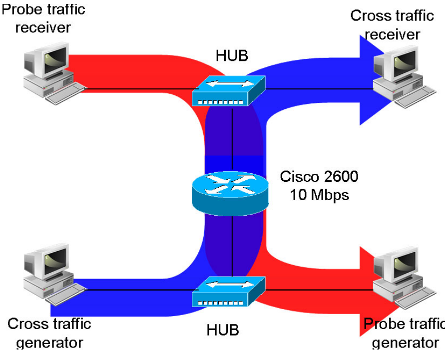

Fig. 6. The short range campus Internet experiment topology.

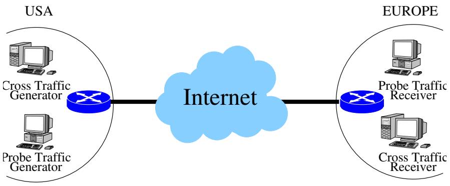

Fig. 7. The long range campus Internet experiment topology.

second idle interval. We manage to experiment with high potential bandwidth packet pairs by performing busy wait when necessary, dealing effectively with the non-real time Linux (Redhat 6.2) kernel 10 msec timer granularity problem. The passive measurement uses a real source of MPEG video. Traces are pre-captured from a Star Wars trailer clip [1]. It is simply smoothed to 200 byte packets sent uniformly over the frame time. The cross traffic is simply 512 bytes payload CBR of the reported rate in all cases. As mentioned the bottleneck link bandwidth is determined using the nettimer tool.

4.1 Active Measurement

In this paragraph we present the active measurements experiments. We perform one experiment per measurement method using the same probing traffic. All experiments are performed one after the other. These experiments prove that our model is correct in practice and it is in fact a significant improvement over BoT.

The active measurements for the short range experiment are summarized in Fig. 8. The figure depicts a graph similar to that in Fig. 5 but using real experiment results instead of equations. In this one graph we summarize results for all relevant active measurement experiments, i.e. using pairs and trains and with the BoT and the ab-probe model sample calculation. The link capacity was measured to be 9 Mbps using the Nettimer method. This is a fairly accurate measure, given that the network is a 10 Mbps E-net. The x axis is the rate of the "additional" injected cross traffic. Note that prior to our experiment there was already some traffic present in the network, over which we have no control. That is why the curves do not start at 9 Mbps at zero cross traffic. If the network had been unloaded initially, all the curves would have started from 9 Mbps at zero crossload, as clearly indicated in Fig. 5. Using the BoT sample calculation we get

---

[1] Star Wars trailer clip: http://www.trailersworld.com/movie.asp?movie_id=692

6.1 Mbps available bandwidth over a 9 Mbps link, while the ab-probe calculation gives approximately 4.5 Mbps. After getting the initial point for the two methods (at cross traffic equal to 0) we draw a lighter “expected available bandwidth” line. It represents the available bandwidth after injecting from 1 Mbps to 3 Mbps. For example, we would expect to measure 5.1 Mbps available bandwidth after injecting a 1 Mbps rate cross traffic (6.1 Mbps minus 1 Mbps). We clearly see however that BoT. does not react at all to the fact that another 1 Mbps is being injected in the path. Even 2 or 3 Mbps cross traffic does not have an impact on the BoT. measurement. On the other hand, ab-probe measures correctly, following the expected available bandwidth line. Comparing this graph with our model graph at Fig. 5 we see that the above observations are exactly seen in our model equations as well. Our theoretical model is very closely validated in the experiments. The ab-probe available bandwidth improvement is visible in practice as well.

In Fig. 9 we see the same graph regarding the active measurement in the long range Internet scenario. In this case the bottleneck link is measured at 4 Mbps. The available bandwidth using the BoT samples in our averaging filter is measured at approximately 2.5 Mbps when no other cross-traffic is injected intentionally. In this case too, when we inject cross traffic of 1 Mbps and 2 Mbps the BoT based measurement fails to drop appropriately, and seems only slightly affected by the extra traffic. The ab-probe method initially measures a 1.5 Mbps available bandwidth (when no cross traffic is injected intentionally). When a 1 Mbps stream is injected the ab-probe reacts to the available bandwidth drop as predicted by theory, measuring around 200 Kbps available. When a 2 Mbps stream is injected the ab-probe measures -1 Mbps. Note that as we mentioned before the available bandwidth may become negative for some short intervals. Looking at the connection loss rates for this case, we see that the probing traffic lost 80% of its packets and the cross traffic lost 33% of its packets.

Even in this highly congested, unstable condition the ab-probe manages to track the available bandwidth changes while the BoT is unable to measure or even detect the oncoming of congestion. Figure 10 shows more details regarding the long distance experiment. This graph shows the packet pair bandwidth estimation points in the above experiment in the ab-probe case on the left and in the bot case on the right. We note the many different distinct concentration points (modes) of the samples in both sample calculation methods, as has been noted by previous work.

In all, the ab-probe active measurement managed to successfully measure the available bandwidth in all cases, even in long distance (more than 20 hops) end-to-end observation, while the BoT. samples have failed.

4.2 Passive Measurement

In the passive measurement, as mentioned before, we probe bandwidth using a VBR real video source. This case is especially interesting because it shows that our measurement can be successfully piggybacked on a rate-based transport. We

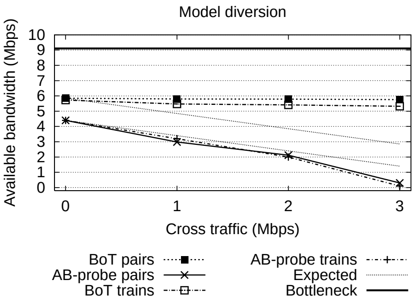

Fig. 8. Active Measurement using BOT and AB-probe in the short range Internet topology.

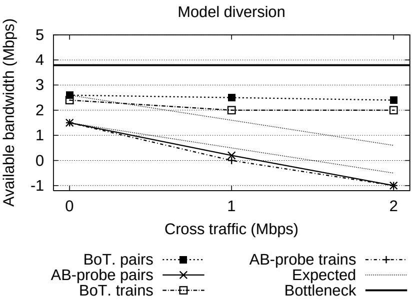

Fig. 9. Active Measurement using BOT and AB-probe in the long range Internet topology.

take advantage of the fluctuating nature of the VBR video source to probe higher bandwidths than the stream average bandwidth.

The experiments show that ab-probe is less accurate than in the active case, as expected, due to the lower and uncontrollable rate of the probing packet pairs. Figure 11 shows our usual model diversion graph for the short range Internet case where a 4 Mbps average video trace is used over a 9 Mbps reported bottleneck link bandwidth. Ab-probe is somewhat successful in reporting the extra cross traffic injected in the network whereas the BoT is practically unaffected by the additional traffic.

We used the 4 Mbps average video for the long range case. The bottleneck link reported was 4 Mbps and the loss rates of the video traffic were: 1%, 1.5% and 50%, for (*a*) no additional cross traffic, (*b*) 1 Mbps and (*c*) 2 Mbps cross traffic respectively. In this case we note: (*i*) Our probing (video) traffic consumes the bottleneck link. The 1 Mbps cross traffic connection suffered over 40% loss therefore leaving the measurements almost unaffected. (*ii*) There is significant queuing of other uncontrolled traffic after of the bottleneck link as evidenced by the fact that the BoT pairs average at higher than bottleneck point rates (many time-compressed samples). Ab-probe is more effective in dealing with compressed packets since it can average them out with the negative samples. (*iii*) Both BoT and ab-probe were able to sense the 2Mbps cross traffic. Note that this is the only satisfactory result regarding the BoT model and is reported when there is significant loss in probing and competing traffic. The BoT is able to capture this effect due to the lost packets (the size of the packets is the nominator). (*iv*) The trains fail to "sense" the 2 Mbps injected cross traffic because the lost packets cannot be accounted for correctly. Specifically, trains may become smaller when first and last packets are lost leaving large intervals without having samples.

In all, a simple averaging filtering was being used, ab-probe showed measurements close to the expected at all times while the BoT showed deficiencies similar to those found in the active measurement case.

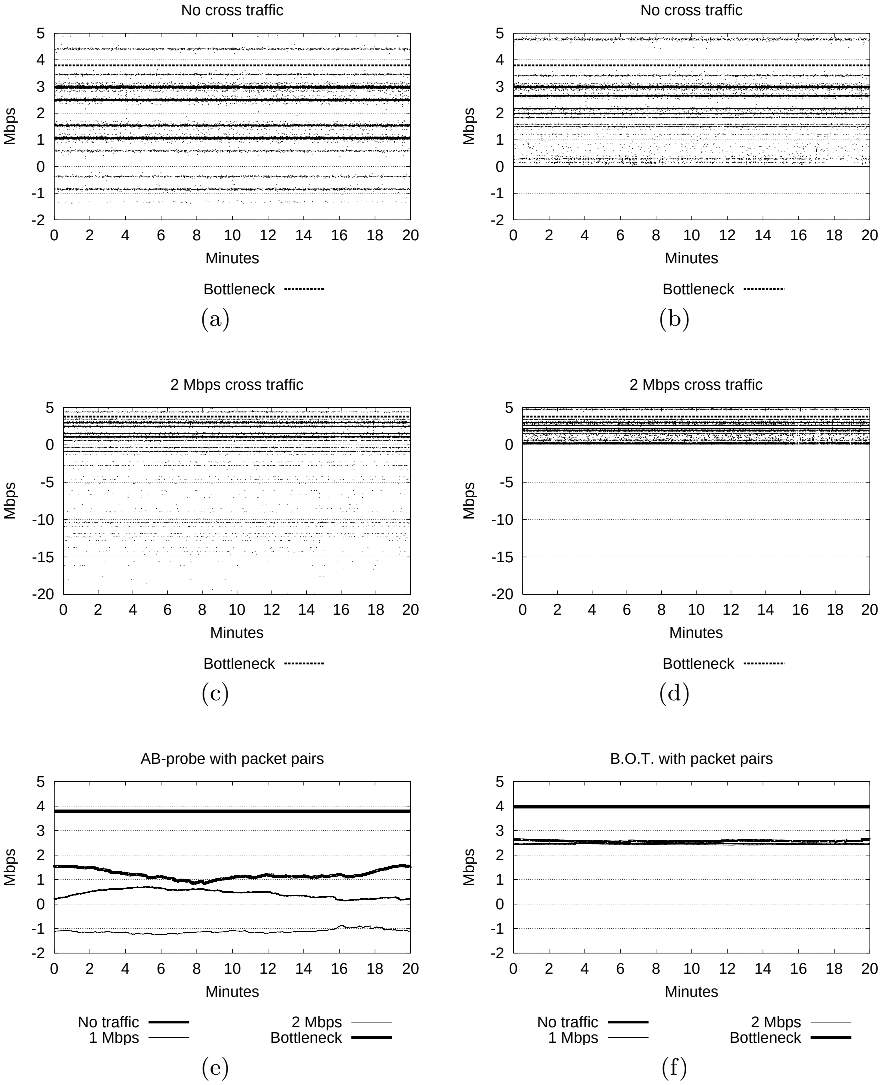

Fig. 10. Graphs from the long distance active measurement experiment. a) Ab-probe short distance packet pair samples when injected traffic is 0 Mbps. b) BoT packet pair samples when injected cross traffic is 0 Mbps. c) Ab-probe short distance packet pair samples when injected cross traffic is 3 Mbps. d) BoT packet pair samples when injected cross traffic is 3 Mbps. e) Ab-probe packet pair samples filtered. f) BoT packet pair samples filtered. The measurement is not affected by the injected traffic in all cases.

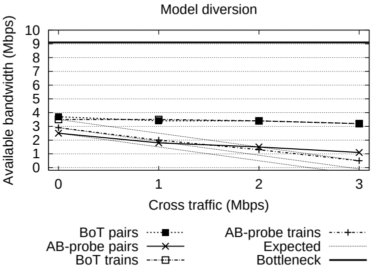

Fig. 11. Model diversion graph for pairs and trains for the short distance case.

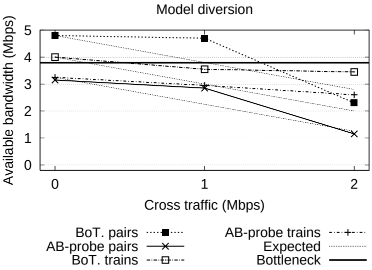

Fig. 12. Model diversion graph for pairs and trains for the long distance case.

5 Conclusions

We have presented ab-probe, a new model and measurement technique to estimate the path available bandwidth from end-to-end observation of packet dispersion. The model directs us towards a new sample calculation that is independent of the network queuing discipline and therefore can be used successfully on the Internet. Two simple measurement methods have been implemented, one active and one passive, mainly with the purpose to show that any measurement implementation based on the new model is reasonably accurate. This work can improve the performance of user sessions, or more generally any network protocol that may need available bandwidth measurement; a very useful quantity, especially for server selection applications, routing, end-to-end transports, adaptive multimedia etc.

Our motivation for this research has been the unintuitive, currently widespread method for calculating available bandwidth samples from packet dispersion used in protocols. The simple BoT calculation is theoretically correct only to the degree that the network approximates a perfect fair queuing for flows. We found that its use generally causes significant errors in non-conforming networks like the Internet, especially at high loads. We developed a model that quantifies this error and used it to produce a new sample calculation that is correct without assumptions for queuing disciplines. The model also captured and explained previously reported c-probe behavior as seen by its researchers in measurements over a real network that has been to this day unexplained. Long range Internet and short range campus experiments showed that ab-probe is accurate in practice as well and a significant improvement over BoT . The improved accuracy in available bandwidth measurements will be very beneficial to adaptive multimedia applications. It will improve network stability and will provide better congestion protection.

References

1. R.L. Carter, M.E. Crovella: Server selection using dynamic path characterization in wide-area networks. Proceedings of IEEE INFOCOM '97, Kobe, Japan, April 1997
2. K. Lai, M. Baker: Nettimer: A tool for measuring bottleneck link bandwidth. Proceedings of the USENIX Symposium on Internet Technologies and Systems, Boston, MA, USA, March 2001.
3. K. Lai, M. Baker: Measuring bandwidth. Proceedings of IEEE INFOCOM '99, New York, NY, USA, March 1999.
4. K. Lai, M. Baker: Measuring link bandwidths using a deterministic model of packet delay. Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, September 2000.
5. C. Dovrolis, P. Ramanathan, D. Moore: What do packet dispersion techniques measure? In Proceedings of IEEE INFOCOM 2001, Anchorage, AK, USA, April 2001.
6. C. Casetti, M. Gerla, S.S. Lee, S. Mascolo, M. Sanadidi: TCP with Faster Recovery. Proceedings of IEEE Military Communications Conference MILCOM 2000, Los Angeles,CA, USA, October 2000.
7. S. Keshav: Packet-pair flow control. http://www.cs.cornell.edu/skeshav/papers.html, 1994
8. Q. Zhang, Y.Q. Zhang, W. Zhu: Resource Allocation for Audio and Video Streaming over the Internet. Proceedings of International Symposium on Circuits and Systems ISCAS 2000, Geneva, Switzerland, May 2000.
9. D. Poulton, J. Oksman: Digital filters for non uniformly sampled signals. Proceedings of Nordic Signal Processing Symposium NORSIG 2000, Vildmarkshotellet Kolmarden, Sweden, June 2000.
10. V. Paxson: Measurements and Analysis of End-to-End Internet Dynamics Ph.D. Thesis, University of California, Berkeley, 1997.
11. S. Floyd, K. Fall: Promoting the use of end-to-end congestion control in the Internet. IEEE/ACM Transactions on Networking, vol. 7, no. 4, August 1999.
12. D. Bansal, H. Balakrishnan: Binomial congestion control algorithms. Proceedings of IEEE INFOCOM 2001, Anchorage, Alaska, April 2001.