TEMB: Tool for End-to-End Measurement of Available Bandwidth

Tricha Anjali and Caterina Scoglio *

Broadband and Wireless Networking Laboratory
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
Email: {tricha, caterina}@ece.gatech.edu

Abstract

The explosive growth of the Internet has induced a need for developing tools to understand the composition and dynamics of the Internet traffic. Measurements of the various characteristics of a network provide insight into the state and performance of the network. The end-to-end available bandwidth metric can be used by various applications for various purposes like path selection, dynamic routing etc. In this paper, a measurement tool for end-to-end available bandwidth is presented. The tool combines the advantages of both active and passive measurement methodologies to obtain accurate, reliable measurements of the available bandwidth along a path.

Keywords: *Measurement, End-to-end available bandwidth, MRTG, Active measurement*

1 Introduction

Measurement is necessary for a network. A user would like to monitor the performance of his applications, check if level of service meets the agreement, etc. A service provider would like to monitor the current level of activity, enforce service level agreements (SLAs), plan for future etc. Measurements can be obtained either in the core of the network or at the edges. Some have local significance at each router while others are end-to-end metrics. They can be obtained by measurements from the various network elements. Common users can only measure the end-to-end metrics. The metrics with local significance at each router can only be measured by the network operators. The approaches to monitor a network are *active* or *passive*. First gives a measure of the performance of the network whereas the latter of the workload on the network. Both have their merits and should be regarded as complementary. The active approach relies on the capability to inject packets into the network and then measure the services obtained from the network. Passive measurements are carried out by observing normal network traffic, without the extra load.

Available bandwidth (together with other metrics like latency, loss etc.) can predict the performance of the network. The *available bandwidth* of a link is the maximum throughput provided to a flow despite the current cross-traffic, when contrasted with the *capacity* which

---

*This work was supported by NASA Goddard and Swales Aerospace under contract number S11201 (NAS5-01090)

is the maximum throughput provided to a flow in absence of cross-traffic. Based on the bandwidth available in various segments of the network, the network operator can obtain information about the congestion in the network, perform the admission control, routing, capacity provisioning etc. The available bandwidth can be measured for individual links of the network. The end-to-end available bandwidth information can be obtained by a concatenation of the available bandwidth measurements of the individual links comprising the path. However, this approach can be very inefficient as the amount of data collected grows as the path size increases and a central data analysis station will be required. Thus, tools have to be devised that can measure the end-to-end available bandwidth directly and accurately from the path. The end-to-end available bandwidth information can be used for selection of alternative paths, selection of web servers etc.

The rest of the paper is organized as follows. In Section 2, we present a description of various bandwidth measurement techniques. In Section 3, we propose TEMB, our Tool for End-to-end Measurement of the available Bandwidth, followed by description of details of the implementation of the MRTG-based tool in Section 4. In Section 5, the results of the experiments for performance evaluation of the proposed tool are presented. Finally, we conclude in Section 6.

2 Related Work

The available bandwidth of a link is indicative of the amount of load that can be routed on the link. Obtaining an accurate measurement of the available bandwidth can be crucial to effective deployment of QoS services in a network. Available bandwidth can be measured using both active and passive approaches. Various tools and products are available that can be used to measure bandwidth of a path in the network. In [1], the authors have described a few bandwidth estimation algorithms. They can be split into two families: those based on pathchar algorithm and those based on Packet Pair algorithm. In the pathchar approach, packets of varying sizes are sent with increasing values of the Time-To-Live (TTL). The packet pair algorithm measures the bandwidth of the narrow link of a route. It operates by sending two packets which get queued along the narrow link of the path and their time-spacing provides estimate of the narrow link bandwidth. In [2], the authors have proposed another tool to measure narrow link bandwidth based on packet pair technique. Some other tools based on the same technique for measuring bottleneck bandwidth (of narrow link) of a route have been proposed in [3, 4].

In [5], the authors have proposed a tool to measure the available bandwidth of a route which is the minimum available bandwidth along all links of the path. It is an active approach based on transmission of self-loading periodic measurement streams. This scheme sends traffic at increasing rates from the source to the destination until the rate finally reaches the available bandwidth of the tight link after which the packets start experiencing increasing amounts of delay. Thus this scheme can be highly intrusive even though momentarily. MRTG is a tool, based on SNMP, that gives periodic measurements of the utilization of a particular link along the path. To obtain statistics of a link via MRTG, the SNMP query needs access to the router. Also, MRTG obtains available bandwidth estimates over periods of length 5 minutes.

Most of the tools/approaches described above obtain estimates of the capacity of the path, rather than the available bandwidth. Even the ones that do measure available bandwidth oper-

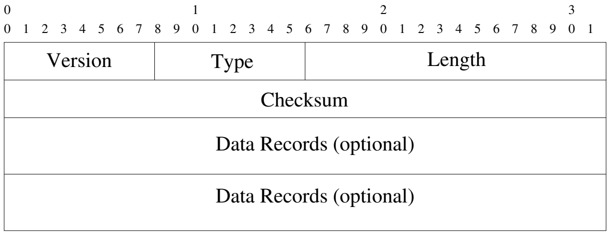


Figure 1: Measurement Packet Format.

ate under a lot of assumptions about the packet pair and their queuing along the path. In the following sections, we propose our own tool, TEMB, for estimation of available bandwidth along a path which is accurate, scalable and flexible.

3 Tool for End-to-End Measurement of Available Bandwidth

In this section, we describe the proposed tool, TEMB, for measuring end-to-end available bandwidth over a path that can possibly span across multiple domains. The tool is needed to answer the question “Where in the path between the two endpoints is the least bandwidth available to a flow and how much is it?”. Currently this is hard to do because the available bandwidth, even for a single link, shows large variations with time, the path may change during the measurement, etc.

The need for an active network measurement tool is well established due to the accuracy requirements. TEMB is a measurement tool that is efficient, easy to implement, and a combination of active and passive approaches. This way, it derives the benefits of both the measurement approaches. The tool is designed such that the measurement packets are processed with about the same computation level as IP forwarding.

TEMB utilizes the interface information from the Management Information Bases (MIBs) in the routers along the path. The functionality of TEMB is distributed between both the source and destination of the path whose measurement is desired. The source sends measurement packets that collect information along the path and are returned back by the destination to the source.

3.1 Measurement Packets

The TEMB tool is based on the use of measurement packets to probe the available bandwidth along the path. The format of the measurement packet is shown in Fig. 1. In the packet:
- Version: Set to 0,
- Type: Set to 0 if the packet is sent from the source to the destination and is routed hop-by-hop by the network, 1 if the path from the source to the destination is already pinned and encoded into the packet, and 2 if the packet is being returned from the destination, as explained later
- Length: Total length of the TEMB packet in bytes,
- Checksum: CRC for the whole packet,
- Data Records: Modified by each hop, as explained later.

Assuming that the TEMB packets can encounter links with the smallest MTU (576 bytes), we design TEMB such that the measurement packets can not exceed a size of 556 bytes,

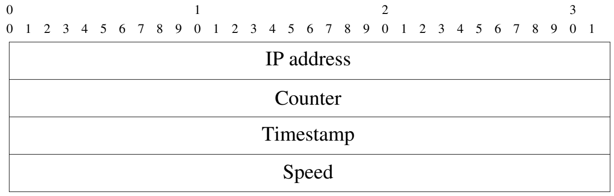


Figure 2: Measurement Packet Data Record.

accounting for the 20 byte IP overhead. The 576 byte limit is imposed as longer packets might get fragmented and eventually discarded. The 556 byte payload implies that a maximum of 34 data records (16 bytes each) can be gathered by a TEMB packet, which is a reasonable limit to the number of hops encountered between any source-destination pair across the world. The TEMB measurement packets are encapsulated into IP packets as explained later.

3.2 Destination Functionality

When a measurement packet finally reaches its destination, it has gathered information from TEMB-compatible hops in the path from the source to the destination, in the form of the data records in the packet. Since the information is along the path and is unidirectional, the source of the path is better suited to analyze the information from the measurements. Thus, the measurement packet has to be returned to the source. Towards this end, the destination interchanges the source and destination fields of the IP header of the measurement packet, changes the type field to 2 to indicate a packet back from the destination to the source and sends the packet to its queues for transmission.

3.3 Hop Functionality

Each hop on the path from the source to the destination appends its information to the measurement packets, if it is TEMB-compatible. The information is in the form of data records which are included at the end of the measurement packet. The structure of the data record is shown in Fig. 2. Each data record contains the IP address of the out-bound interface of the router, the counter for the number of octets that have passed that interface at the processing time, the time-stamp at which the packet was processed by the router, and the speed of the outgoing interface. The value of the counter can be obtained by looking at the ifOutOctets object in the interfaces group of the MIB-II in the router. As the router has modified the measurement packet, it has to recompute the length of the packet and the CRC checksum for the modified packet. These values then have to be substituted into the packet and the packet then queued for transmission downstream.

3.4 Source Functionality

In the design of TEMB, most of the operational burden is given to the source router of the path. This is because it is closest to the user who demands the bandwidth and the QoS and can inform the user about the path conditions. The source of the path has to assemble the initial measurement packet with the initial data records and correct packet length and checksum. Then, the measurement packet is encapsulated into an IP packet and the appropriate link layer packet. The source makes multiple copies of this packet and sends them over the path to the destination. Multiple packets are sent to obtain a correct estimate of the identity of the tight link and its available bandwidth. Also the source node has to analyze the incoming packets and take further measures to obtain more refined tight link available bandwidth measurements. The detailed operation of the TEMB tool is described next.

3.5 Overall Operation

TEMB is a tool designed to measure the available bandwidth along a path between a source and a destination. The operation of TEMB can be split into two parts. The first obtains crude estimates of available bandwidth along all the links of the path and determines the least amount among them as a method to identify the tight link along the path. The second part obtains more accurate measurement of the available bandwidth on the identified tight link. The tool operates by sending the measurement packets from the source to the destination. TEMB is designed to initially transmit 10 measurement packets during an interval of 1 sec. The number 10 was chosen because it gives a reasonable approximation of the bandwidths along the path without being highly intrusive. These packets gather information along the path and the destination sends the packets back to the source. If the traffic profile along the path is highly variable, TEMB is designed to dispatch another set of 10 packets in one second to obtain better identification of the tight link of the path.

Suppose N out of the initial 10 packets are finally back at the source for analysis. Let T denote the set of all the time-stamps gathered by the packets, *i.e.*, $T = \{t_1, t_2, \ldots, t_N\}$, where the elements have been arranged in increasing order, *i.e.*, $t_1 < t_2 < \ldots < t_N$. Let P denote the list of successive interfaces encountered by the measurement packets, *i.e.*, $P = \{I_1, I_2, \ldots, I_H\}$ where H is the number of hops in the path. Let C_I denote the set of counters for interface I in the path, *i.e.*, $C_I = \{c_{I1}, c_{I2}, \ldots, c_{IN}\}$, where c_{Ik} denotes the counter for interface I along the path at time t_k. Since the set T is arranged in an increasing order, the elements of C_I are also monotonous non-decreasing as the number of packets crossing an interface is always non-negative. Let S_I be the speed of the interface, as gathered by the measurement packets. Then, the utilization of the interface is calculated from the $k-th$ sample as

$$U_{Ik} = \frac{c_{Ik} - c_{I(k-1)}}{t_k - t_{(k-1)}} \quad \text{for } k = 2, 3, \ldots, N$$

and the available bandwidth is $A_{Ik} = S_I - U_{Ik}$. Once these $(N-1)$ estimates are obtained, TEMB tries to identify the tight link. If all the estimates agree on a certain interface being the one with the minimum available bandwidth and the estimated values are similar, TEMB identifies the tight link. On the other hand, if the estimates disagree on either the interface or its available bandwidth, TEMB sends the next batch of measurement packets. The agreement about the identity of the tight link is reached if at least a certain percentage ($agree_{link}$) of the estimates concur. The agreement about the estimated value of the available bandwidth is reached if all the $(N-1)$ estimates for the interface I are less than a a certain percentage ($agree_{avail}$) of the minimum estimate. Note that the values of both $agree_{link}$ and $agree_{avail}$ should be very close to 100 but the former should be less than 100 and the latter greater than 100.

Finally the average available bandwidth of the selected interface I is

$$A_I = \frac{1}{n * (N-1)} \sum_{k=1}^{p*(N-1)} A_{Ik} \qquad (1)$$

where n is the number of attempts that TEMB made during the first step. The identified link is then chosen for further investigation. Also, if the estimated available bandwidth for any other interface falls within 150% of the lowest value, that interface is also marked critical and qualifies for further investigation. This is done due to the non-stationary nature of cross-traffic.

The tool is designed to operate in two cases: a. When the path between the source and destination is the min-hop path, b. When the available bandwidth measurement is required for a non-min-hop path between the source and destination. In the first case, the measurement packets formed at the source have empty data records, the type is set to 0, and the measurement packets are IP encapsulated. As they move down the path, each hop adds its data record and forwards the packets to the destination by utilizing the pre-existing IP lookup tables. In the second case, the data records are already included in the measurement packets at the source. The data records contain the IP address of the hops along the path. In the measurement packets, the type value is set to 1. The hops of the path modify their data records by including the interface information and then queue the packets for transmission towards the next hop, as recorded in the next data record. In this way, the available bandwidth measurements can be obtained for predetermined paths between two endpoints.

Once the identification of the tight links of the path is done, a more accurate estimation of its available bandwidth is desired. This is done by utilizing an MRTG based approach that is passive in nature and has been monitoring the interface over time. This approach is described in the next section.

4 MRTG Based Measurement Approach

The first step of the TEMB tool identifies the tight link(s) of the path from the source to the destination. The next step involves obtaining more accurate estimates of the available bandwidth along that link by utilizing an MRTG based approach. The approach is based on past router statistics collected in a periodic manner.

MRTG, as mentioned before, has a limitation that the utilization measurements are available with a periodicity of 5 minutes. This maybe too inaccurate for certain applications. Thus, we have developed MRTG++ [6], which is a modified version of MRTG with averaging intervals of 10 sec. This is done by modifying the database structure and querying the routers more often.

Our approach is based on periodic collection of router statistics. Let us define, for a link:

- C: Capacity of link in bits per sec,
- $A(t)$: Available capacity at time *t* in bits per sec ,
- $L(t)$: Traffic load at time *t* in bits per sec,
- τ: Length of the averaging interval of MRTG,
- $L_\tau[k]$, $k \in \mathbf{N}$: Average load in $[(k-1)\tau, k\tau]$.

The available capacity can be obtained as $A(t) = C - L(t)$. So, it would be sufficient to measure the load on a link to obtain available bandwidth. We also define

- p is the number of past measurements in prediction,
- h is the number of future samples reliably predicted,
- $A_h[k]$: the estimate at $k\tau$ valid in $[(k+1)\tau,\ (k+h)\tau]$.

Our problem can be formulated as linear prediction:

$$L_\tau[k+a] = \sum_{n=0}^{p-1} L_\tau[k-n]\, w_a[n] \qquad \text{for } a \in [1, h] \tag{2}$$

where on the right side are the past samples and the prediction coefficients $w_a[n]$ and on the left side, the predicted values. The problem can be solved in an optimal manner using covariance method. We propose to dynamically change the values of p and h based on the traffic dynamics.

So, the steps of the approach involve solution of the prediction equations to determine the prediction coefficients, find the predicted utilization, find the prediction error, and finally adjust the values of p and h based on the prediction error.

New values for p and h have to be determined to adapt the prediction process to the traffic dynamics. This is done based on the observed prediction error $e_\tau[k+a]$ given by $e_\tau[k+a] = \Big(L_\tau[k+a] - \widehat{L}_\tau[k+a]\Big)^2$ for $a = 1, \dots, h$. The details of the algorithm can be found in [6]. We limit the values of h and p by h_{min} and p_{max} because small values of h imply frequent re-computation of the regression coefficients and large values of p increase the computational cost of the regression. Also, we have introduced the thresholds Th_1 to Th_4 to decide when to change the values of the parameters p and h. They are determined based on the traffic characteristics and the conservatism requirements of the network domain.

5 Performance Evaluation

In this section, we present the results of our experiments and simulations to verify the operation of the proposed tool TEMB. Our simulations are divided into two categories to verify the two parts of the functionality of TEMB. First, we will present some implementation details about TEMB.

5.1 Implementation Details

Our proposed tool TEMB for identification of the tight link along a path and subsequent available bandwidth measurement along the tight link is designed to be as much non-intrusive as possible. Unlike some other schemes that transmit packets at speed higher than the the tight link available bandwidth to get an estimate, our tool sends 10 packets in a second per path for each measurement required, which may be repeated a couple of times. Thus, we combine the advantages of both the active and passive approaches of measurement by using a hybrid tool.

One advantage of using TEMB is that the router time-stamps saved in the measurement packets have only local significance, *i.e.*, they are not correlated with other routers. The routers do not need information about the real time reference of their time-stamp. Only the difference in the consecutive time-stamps is used to calculate the utilization for that interval. Also, the ordering of the measurement packets does not matter when they reach the destination.

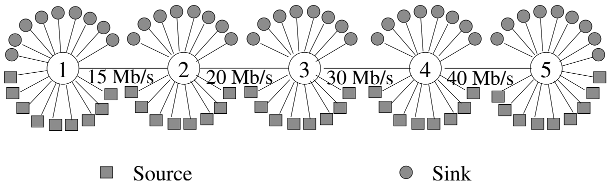


Figure 3: Simulation topology.

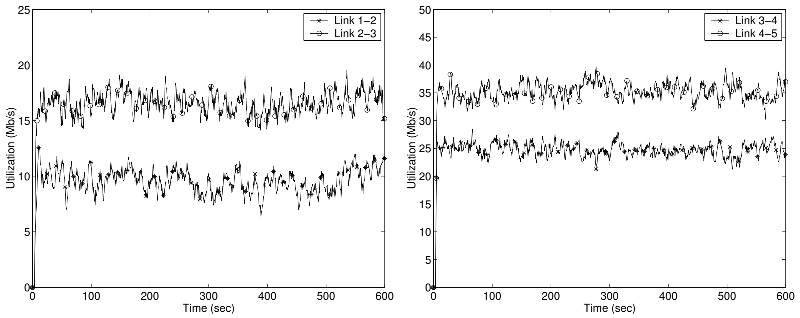
5.2 Simulator Description

We use the well-known network simulator ns to simulate the topology of Figure 3. Traffic is generated by utilizing various UDP and TCP sources which are attached to different nodes in the path. The UDP senders are ON/OFF sources. The durations of the ON and OFF periods are selected from a Pareto distribution with parameter β. During the ON period, the transmission rate of the sender is a constant configured rate. In our simulations we have used three different UDP sender profiles. A combination of Pareto-based ON/OFF sources is used as it leads to long-range dependence in the multiplexed traffic. Sources send traffic to sinks located at all the nodes in the network. As we are using a combination of ON/OFF and TCP sources with different source/sink pairs, average sending rates, the resultant traffic on the links of our simple simulation topology is not correlated and is a good representative for current Internet traffic. For verification of the MRTG-based tool, we obtain traffic measurements from a real Internet backbone.

5.3 Path Probing Results

Our goal here is to show how well the path probing tool of TEMB works over time when presented with different traffic patterns on links of the path and how the parameter tuning affects the performance. Our simulation topology consists of 5 nodes arranged in a totem-pole (Fig. 3). While this topology does not cover the full heterogeneity of the Internet, it is sufficient for our purposes as it provides different, uncorrelated traffic patterns on the different links. The end-to-end available bandwidth measurement is desired from node 1 to node 5. A combination of TCP and UDP sources are attached to the nodes. The value of $agree_{link}$ is set at 85% and $agree_{avail}$ at 120%. Thus, at least 85% of the measurements have to point to a certain link for it to be identified as the tight link of the path and the maximum measurement obtained for that interface should not exceed 120% of the minimum. This constraint is applied to ensure that the traffic profile does not vary a lot during the measurement to guarantee a good measurement estimate for the available bandwidth of the tight link. Fig. 4 shows a sample utilization profile for the four links, obtained from one simulation run. If the path probing mechanism of Sec. 3 is applied to probe the available bandwidth at time instant 360 sec, we obtain with 77% agreement that the least available bandwidth is along link 1-2. This is because 7 of the 9 measurements pointed towards link 1-2. If the application that desires the measurement demands a higher value of $agree_{link}$, then the next batch of measurement packets is sent at time 362 sec. From this batch, all 9 measurements point towards link 1-2. Thus the confidence in identification of link 1-2 as the tight link becomes 89% which is higher than the threshold $agree_{link}$. We also observe that the maximum among the measurements obtained for the link 1-2 is about 113% of the minimum, which is below the limit set by $agree_{avail}$. Thus the tight link is identified as link 1-2 and its available bandwidth is the minimum measurement obtained, *i.e.* 2.6Mbps. Also, link 4-5 is marked as critical as its available bandwidth measurements fall below the 150% mark of the link 1-2 minimum 2.6Mbps. If, on the other hand, the path probing mechanism was applied at time 390 sec, we obtained that 7 of the 9 measurements pointed towards link 2-3. As this agreement (77%) is below $agree_{link}$, the next 10 measurement packets are sent at time 393 sec. Among them, 88% point towards link 4-5, which necessitates another attempt at tight link identification. The third attempt at time 395 sec returns link 2-3 with an agreement of 88% which leads to an overall agreement of 55% for link 2-3. This leads to 6 further attempts which all point to link 2-3 with a 100% agreement before the link 2-3 is finally chosen as the tight link. These two scenarios illustrate that the computational effort and the intrusiveness of the scheme is highly dependent on the parameters $agree_{link}$ and $agree_{avail}$ specified by the application. If the application needs a high level of agreement before identifying the tight link, multiple attempts may be necessary to achieve the same.

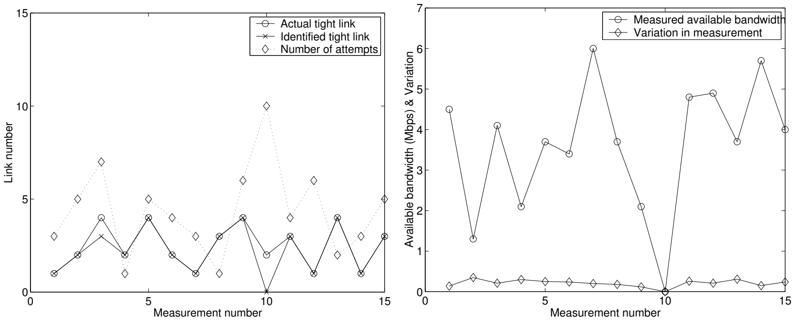
In Fig. 5(a), we show the results for 15 independent runs of the path probing mechanism for the topology and setup shown in Fig. 3. The link 1 is the link between nodes 1 and 2, link 2 between nodes 2 and 3, and so on. The values of $agree_{link}$ and $agree_{avail}$ were set to 80% and 140%, respectively. As can be seen, the mechanism only falters in one case (measurement 3). In the case of measurement experiment 10, 10 attempts were made unsuccessfully to determine the tight link, whereupon the particular experiment was deserted. Also shown in the figure with the dotted line is the number of attempts that the path probing tool made at the measurement before arriving at the final result. As can be seen, the number of attempts is not very high, demonstrating that the scheme is not highly-intrusive (as each attempt includes transmitting 10 packets in 1 sec). In Fig. 5(b), we show the variation of the measured available bandwidth of the identified tight link for the same 15 experiments. The variation is defined as the ratio of the difference of the maximum measurement and the minimum measurement to the minimum measurement. Also shown is the final estimate of the tight link available bandwidth. The figure gives a slight hint that the variation in the available bandwidth increases as the tight link utilization increases.



(a) Link 1-2 and 2-3

(b) Link 3-4 and 4-5

Figure 4: Utilization of the four links.



(a) Verification of path probing mechanism.

(b) Variation of available bandwidth.

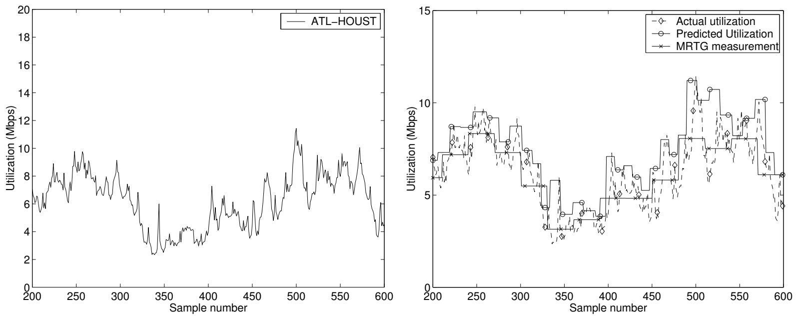
Figure 5: TEMB performance

5.4 MRTG-Based Tool Results

Our MRTG-based tool is designed to find, for the link identified by the path probing mechanism, a more accurate estimate of the available bandwidth. It operates, independent of any assumptions about the traffic models, based on the actual measurements obtained from the link. To validate the performance of the MRTG-based tool, we obtain traffic traces from measurement available from real Internet backbones. This is done because it gives insight into the performance of the tool for observed real traffic traces.

The choice of the parameters used in the tool (Th_1, Th_2, etc. and h_{min}, p_{max}) for updating the values of p and h has to be made by the network operator depending on the conservativeness requirements of the network operation. We have obtained the following results by choosing $Th_1 = 1.1$, $Th_2 = 0.9$, $Th_3 = 0.7$, $Th_4 = 0.5$ and $h_{min} = 10$, $p_{max} = 50$. Also, the representative utilization for an interval is fixed at the maximum predicted utilization, in order to provide a very conservative estimate for the link available bandwidth. In the following, the traffic traces have been obtained from Abilene, the advanced backbone network of the Internet2 community of universities, on July 1, 2002. The performance was checked for traffic traces obtained from various links of the network, but in the following we present the results for the traffic trace between Atlanta and Houston routers, for the outgoing traffic from the Atlanta router (shown in Fig. 6(a)). In the figure, the samples are collected with a time granularity of 10 seconds.

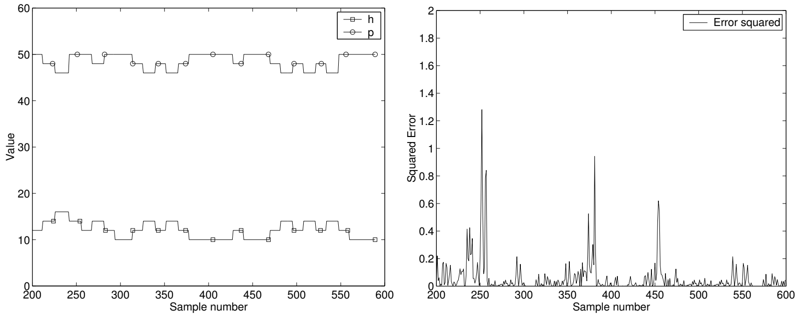
When the tool is applied to this traffic trace, the predicted utilization is shown in Fig. 6(b). Also shown in the figure is the utilization profile that would be observed if MRTG was applied to the trace, with its 5 minute averaging. As we can see, our tool performs much better than MRTG as it gives a very conservative utilization estimation. Also combined with our tool is the capability to predict, for a future small interval, the utilization with a high degree of confidence. In this figure, we have not shown the available bandwidth profile as it can be obtained simply by subtracting the utilization from the total capacity of the link. Also, the predicted utilization shows more relevant information when placed against the actual measurements. In the figure, the results are shown from sample number 200 onwards as the initial samples are used to stabilize the tool. In Fig. 7(a), we show the values that were assigned



(a) Output traffic from ATL on ATL-HOUST.

(b) Predicted Utilization.

Figure 6: Performace for real traffic trace



(a) Variation in the parameters h and p.

(b) Error in prediction.

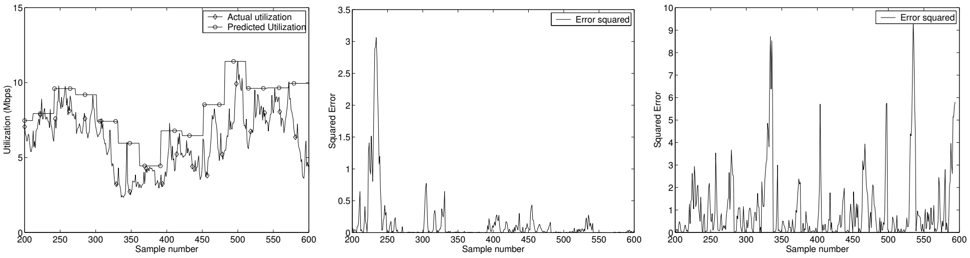
Figure 7: Prediction performance

to the forecast parameters p and h during the experiment.

As can be seen, the values of h and p are not always the limits set by h_{min} and p_{max}. This shows that the scheme was able to gain confidence in its prediction for certain intervals. Next, in Fig. 7(b), we show the squared error of the prediction. As can be noticed from comparing the figure with Fig. 6(b), the error does not exceed 10% of the actual utilization.

Next, in Fig. 8(a), we have increased the value of h_{min} to 30 forcing the tool to predict for longer intervals even though the confidence in the prediction may not be so high. As can be seen, the prediction error increases (as expected) but the prediction is still very conservative.

To validate the operation of the proposed tool, we present in Figs. 8(b) and 8(c) the scenario where the parameters h and p are fixed, respectively. In other words, for the case in Fig. 8(b), h was fixed at 30 to obtain results for periods equal to MRTG intervals of 5 min. For the case in Fig. 8(c), p was fixed at 30 to obtain the results. This value was picked as it is the average of the range allowed for p in our simulation. If a larger value is chosen for p, it encompasses more computational effort during the covariance normal equation solution. In both of these cases, the algorithm for h amd p modification is simplified as the values of either h or p need not be calculated. Upon comparing the error obtained in Figs. 8(b) and 8(c) with the error



(a) Increased h_{min} (b) Constant h (c) Constant p

Figure 8: Effect of parameter variation

profile in Fig. 7(b) for our unmodified experiment, we see that the latter is less than the former two. This fact demonstrates that the adaptation of the parameters h and p has indeed reduced the error in the prediction.

In summary, the presented tool TEMB is an efficient method to calculate the end-to-end available bandwidth between two network points, either on a given path or on the min-hop path between the two.

6 Conclusions

In this paper, we have developed and presented a tool to measure the available bandwidth on a path between two specified points in a network. The path can be pre-specified or determined hop-by-hop. The proposed tool TEMB is very accurate, reliable, scalable and non-intrusive. The tool first identifies the tight link along the path, with the least available bandwidth. Then it proceeds to find more accurate measurements by obtaining statistics from the involved routers directly. This procedure also predicts the available bandwidth and tells the duration for which the prediction is valid with a high confidence. The tool dynamically adapts the prediction process to the traffic variations. Various applications, like TCP, streaming applications, etc., can benefit from the accurate measurement of end-to-end available bandwidth. The values of the various parameters used in the specification of the tool like $agree_{link}$, $agree_{avail}$, Th_1 to Th_4, p_{min}, p_{max}, h_{min} have to be negotiated between the application and the network operator. They define the performance of the tool in terms of its efficiency, conservativeness etc. An optimal method to determine these parameters for given application specifications can be the next step in the research.

Acknowledgements

We would like to thank Dr. Ian F. Akyildiz for his invaluable support and guidance during the course of our research. We would also like to thank George Uhl of NASA Goddard Space Flight Center for his comments and feedback.

References

[1] K. Lai and M. Baker, "Measuring Bandwidth," in *Proceedings of IEEE INFOCOM'99*, New York, USA, March 1999, pp. 235–245.

[2] K. Lai and M. Baker, "Nettimer: A Tool for Measuring Bottleneck Link Bandwidth," in *Proceedings of 3rd USENIX Symposium on Internet Technologies and Systems*, San Francisco, USA, March 2001, pp. 184–193.

[3] R. L. Carter and M. E. Crovella, "Measuring Bottleneck Link Speeds in Packet-Switched Networks," *Boston University Technical Report BU-CS-96-006*, 1996.

[4] V. Paxson, "End-to-end Internet Packet Dynamics," in *Proceedings of ACM SIGCOMM'97*, Cannes, France, September 1997, pp. 139–152.

[5] M. Jain and C. Dovrolis, "End-to-end Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput," in *Proceedings of ACM SIGCOMM'02*, Pittsburgh, USA, August 2002, pp. 295–308.

[6] T. Anjali, C. Scoglio, L. Chen, I. F. Akyildiz, and G. Uhl, "ABEst: An Available Bandwidth Estimator within an Autonomous System," in *Proceedings of IEEE GLOBECOM'02*, Taipei, Taiwan, November 2002, pp. 2360–2364.