MABE: A NEW METHOD FOR AVAILABLE BANDWIDTH ESTIMATION IN AN MPLS NETWORK

T. ANJALI, C. SCOGLIO AND I. F. AKYILDIZ
Broadband and Wireless Networking Lab,
School of Electrical and Computer Engineering,
Georgia Institute of Technology, Atlanta, GA, 30332 USA
E-mail: {tricha, caterina, ian}@ece.gatech.edu

J. A. SMITH AND A. SCIUTO
NASA Goddard Space Flight Center
Greenbelt, MD 20771 USA
E-mail: {jsmith, asciuto}@rattler-e.gsfc.nasa.gov

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 whether it is behaving as expected and whether changes in the network have improved or degraded its performance. Available Bandwidth on the links of a network is an important metric which can predict the performance of the network. In this paper, an estimation algorithm for the available bandwidth on a link is presented. The algorithm estimates the available bandwidth and tells the duration for which the estimate is valid with a high degree of confidence. The algorithm dynamically changes the frequency of obtaining the past samples that are used for prediction and thus the duration for which the prediction holds.

1 Introduction

In recent years, the growth of Internet has been beyond imagination. The traffic patterns are changing day-to-day and the network devices have to be configured to adjust to the evolving traffic. Thus network management is becoming increasingly difficult because of the network size, large traffic volumes, and distributed administration. An important tool in the hands of a domain administrator is a measurement tool that provides measurements of the various characteristics of the network. These measurements provide insight into the state and performance of the network whether it is behaving as expected and whether changes in the network have improved or degraded its performance. Network measurement is necessary both from a user's and network manager's point of view. 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. Some QoS metrics have been defined by IETF for these purposes. The network measurements can be obtained either in the core of the network and or at the edges. Some have local significance at each hop while others are end-to-end metrics. To obtain measured statistics from individual network elements is not possible for users due to security reasons. They can only measure the end-to-end metrics. On the other hand, only the network operators can measure and make available the metrics with local significance at each router.

A network can be monitored either *actively* or *passively.* 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. It introduces extra traffic into the network. But the active approach has the advantage of measuring the desired quantities at the desired time. Passive measurements are carried out by observing normal network traffic, without the extra load. The passive approach measures the real traffic. But the amount of data accumulated can be substantial because the network will be polled often for information.

There are various quantities of interest that can be insightful about the state of the network. Available bandwidth (together with other metrics like latency, loss etc.) can predict the performance of the network. Based on the bandwidth available, the network operator can obtain information about the congestion in the network, decide the admission control, perform routing etc. For MPLS networks, the available bandwidth information can be used to decide about the LSP setup, [1] routing (Shortest Widest Path, [2] Widest Shortest Path [3]), LSP preemption, [4] etc. Each of these processes needs available bandwidth information at a suitable time-scale. It is desirable to obtain the available bandwidth information by measurements from the actual LSPs because they give more realistic information about the available bandwidth. The available bandwidth information can also be obtained by subtracting the nominal reservation for the tunnels from the link capacity which gives a lower bound.

The rest of the paper is organized as follows. In Section 2, we present a description of various bandwidth measurement techniques and motivate for the new algorithm. In Section 3, we propose our algorithm MABE, the method for available bandwidth estimation in an MPLS network. In Section 4, we describe the details of the implementation of the algorithm. In Section 5, the results of the experiments are presented. Finally, we conclude in Section 6 with our overall observations about the measurement algorithm.

2 Measuring Available Bandwidth

The available bandwidth on 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. The available bandwidth can be measured either on a single link of the network or on a route in the network. The link information can be used for congestion avoidance, routing etc. The route information corresponds to the available bandwidth measurement of the most congested link on the route. Various tools and products are available that can be used to measure available bandwidth of a link in the network. A few bottleneck bandwidth algorithms are described in. [5] They can be split into two families: those based on pathchar [6] algorithm and those based on Packet Pair [7] algorithm. The pathchar algorithm is an active approach which leads to the associated disadvantages of consumption of significant amount of network bandwidth etc. The packet pair algorithm measures the bottleneck bandwidth of a route. It can have both active and passive implementations. Active implementations have bandwidth consumption whereas passive implementations may not give correct measurement. Another tool to measure bottleneck link bandwidth based on packet pair technique is proposed in. [8] Some other tools based on the same technique for measuring bottleneck bandwidth of a route have been proposed in. [9,10] None of them measures the available bandwidth or utilization of a desired link of a network. A tool to measure the available bandwidth of a route as the minimum available bandwidth along all links of the path is proposed in. [11] It is an active approach based on transmission of self-loading periodic measurement streams. Another active approach to measure the throughput of a path is Iperf [12] from NLANR that sends streams of TCP/UDP flows. Cisco has introduced the NetFlow [13] technology that provides IP flow information for a network. NetFlow provides detailed data collection with minimal impact on the performance on the routing device and no external probing device. But in a DiffServ environment, the core of a network is interested in aggregate rather than per-flow statistics, due to the scalability issues. All the tools, except NetFlow, give path measurements based on an active approach. A network operator, on the other hand, would be interested in finding the available bandwidth on the links of the network to get a complete picture. He has access to the routers/switches of the network and can measure available bandwidth from the routers directly without injecting pseudo-traffic. Thus, he does not need the end-to-end tools that utilize the active approach of measurement. One approach is to use Simple

Network Management Protocol (SNMP) [14] to communicate with the routers. SNMP is a short-term protocol to manage nodes in the network. An SNMP-managed network consists of three key components: managed devices, agents, and network-management systems (NMSs). A managed device is a network node that contains an SNMP agent and that resides on a managed network. Managed devices collect and store management information in Management Information Bases (MIBs) [15] and make this information available to NMSs using SNMP. Managed devices, sometimes called network elements, can be routers and access servers, switches and bridges, hubs, computer hosts, or printers. An agent is a network-management software module that resides in a managed device. An agent has local knowledge of management information and translates that information into a form compatible with SNMP. An NMS executes applications that monitor and control managed devices. NMSs provide the bulk of the processing and memory resources required for network management. Thus SNMP can be used as a passive technique to monitor a specific device. MRTG [16] is a tool based on SNMP to monitor the network links for their traffic utilization. It has a highly portable SNMP implementation and can run on most operating systems.

Thus, the network operator requires a tool for measuring the available bandwidth on a certain link of the network in a passive manner whenever he desires. Since the operator has access to the router, he can use MRTG. [16] But MRTG has the limitation that it gives only 5 minute averages of link utilization. For applications like routing, this large interval averaging may not be enough. MRTG can be enhanced to decrease the averaging interval down to 1 minute. This may still be large for some applications. Thus, we have modified MRTG to MRTG++, to obtain averages over 10 second durations. This gives us the flexibility to obtain very fine measurements of link utilization. Even though the operator can have these measurements, he may not desire each measurement and also this will increase the load on the routers. So, we propose MABE, a new method for available bandwidth estimation, that is a gradient adaptive lattice based linear regression algorithm to predict the utilization of a link. The algorithm is adaptive because the frequency of sample collection is varied depending on the traffic profile. Using MABE, we predict the utilization of a link simply and efficiently by varying the sample collection times.

3 MABE: Method for Available Bandwidth Estimation in an MPLS Network

In our approach, we concentrate on an MPLS domain and its management. We propose a centralized NMS (at domain level) that will determine the available bandwidth for all the links in the domain. The approach is based on the use of MRTG where the manager will enquire each router in the domain through SNMP and obtain the information about the available bandwidth on each of its interfaces. The most accurate approach will be to collect information from all possible sources at the highest possible frequency allowed by the MIB update interval constraints. However, this approach can be very expensive in terms of signaling and data storage. Furthermore, it can be redundant to have so much information. Thus, we propose a method to predict the utilization based on a few samples to avoid redundancy ad lower the processing/archiving efforts.

We can set the measurement interval of MRTG++ and measure the average link utilization statistic for that interval. We define for a link between two nodes i and j:

- 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_{min}: Length of the minimum sampling interval of MRTG,
- τ: Length of the current sampling interval of MRTG, (integral multiple of τ),
- η, $\eta \in \mathbf{N}$: τ/τ_{min}, the number of τ_{min} in current τ,
- $L_\tau[k]$, $k \in \mathbf{N}$: Average load in $[(k-\eta)\tau_{min}, k\tau_{min}]$.

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. Note that we have not explicitly shown the $i-j$ dependence of the defined variables. This is because the analysis holds for any node pair independent of others. We also define

- p is the number of past measurements in prediction,
- $\widehat{A}[k+\eta]$ is the prediction for $A(t)$ for time $[k\tau_{min}, (k+\eta)\tau_{min}]$.

- $\widehat{L}_\tau[k+\eta]$ is the prediction of utilization for time $[k\tau_{min}\ (k+\eta)\tau_{min}]$.

We propose a linear regression based algorithm for prediction of the link utilization. Superimposed on the regression is another algorithm to vary the sampling frequency on a scale smaller than the sampling frequency itself. During the interval when the sampling frequency is held constant, the linear prediction can be specified as

$$L_\tau[k+\eta] = \sum_{n=0}^{p-1} L_\tau[k-\eta n]\, a_p[n] \tag{1}$$

where on the right side are the past samples $L_\tau[k-\eta n]$ and the prediction coefficients $a_p[n]$ and on the left side, the predicted value. We use a Gradient Adaptive Lattice filter [17] to find the prediction coefficients to minimize the forward and backward prediction errors. We propose to dynamically change the value of η based on the traffic dynamics to calculate a more efficient prediction of the utilization.

The MABE algorithm is given in Fig. 1. In step 2 of the algorithm, we need to find the reflection coefficients of the Gradient Adaptive Lattice filter. The filter minimizes the forward and backward prediction errors, $f_j(k)$ and $b_j(k)$ respectively, for $j = 1, 2, \ldots, p$, where

$$f_j(k) = L_\tau[k] + \sum_{n=1}^{p} L_\tau[k-\eta n]\, a_j[n],$$

$$b_j(k) = L_\tau[k-\eta j] + \sum_{n=1}^{p} L_\tau[k-\eta j+\eta n]\, a_j[n].$$

With the steepest descent approach to minimize the total error, we get the reflection coefficient update equation for $j = 1, 2, \ldots p$ as

$$\Gamma_j(k+1) = \Gamma_j(k) - \mu_j(n)\,\{f_j(k)b_{j-1}(k-1) + f_{j-1}(k)b_j(k)\}$$

where μ_j is a time-varying step size used to normalize the gradient adaptive lattice filter and

$$\mu_j(n) = \frac{2}{\sum_{l=0}^{n}(0.5)^{(n-l)}(f_{j-1}^2(l) + b_{j-1}^2(l-1))}.$$

From the reflection coefficients, the prediction coefficients a_p can be calculated as

$$a_{j+1}[i] = a_j[i] + \Gamma_{j+1} a_j[j-i+1]$$

```
Algorithm:
 1. At time instant $k$, available bandwidth measurement is desired.
 2. Find the coefficient vector $a_p$ using Gradient Adaptive Lattice
    method for given $p$ and the previous measurements.
 3. Predict $\hat{L}_\tau[k+\eta]$ from equation 1 and $[L_\tau[k-(p-1)\eta],\ldots,L_\tau[k]]$ .
 4. Find $\hat{A}[k+\eta]$.
 5. At time $(k+\eta)\tau_{min}$, get $L_\tau[k+\eta]$.
 6. Find the error $e_\tau[k+\eta]$ and its mean $(\mu)$ and standard deviation
    $(\sigma)$.
 7. Set $k = k+\eta$.
 8. Obtain new value for $\eta$ from Fig. 2.
 9. If $\eta$ has changed then call the transient algorithm.
10. Go to step 1.
```

Figure 1. The MABE Algorithm.

where j is the order of the prediction and $i = 0, 1, \ldots j+1$. The prediction coefficients a_p can then be used to predict $\widehat{L}_\tau[k+\eta]$ using equation (1). Next step is to obtain an estimate of the available bandwidth $\widehat{A}[k+\eta]$ for the interval $[k\tau_{min}, (k+\eta)\tau_{min}]$ as $\widehat{A}[k+\eta] = C - \widehat{L}_\tau[k+\eta]$.

After obtaining the actual load value $L_\tau[k+\eta]$ at time $(k+\eta)\tau_{min}$, we find the prediction error vector $[e_\tau[k-(p-2)\eta], \ldots, e_\tau[k+\eta]]^T$ where each element is given as

$$e_\tau[k+a\eta] = \left(L_\tau[k+a\eta] - \widehat{L}_\tau[k+a\eta]\right)^2, \quad a = -p+2, -p+1, \ldots, 1.$$

Next, we propose an algorithm to estimate new value for η based on a metric derived from the mean (μ) and standard deviation (σ) of error e_τ. The algorithm is given in Fig. 2.

In the algorithm, we have introduced η_{min} and η_{max} because small value of η implies frequent regression re-computation while large value of η decreases the reliability of the regression. Also, we have introduced the thresholds Th_1 and Th_2 to decide when to change the value of η. They are determined based

Algorithm:

1. If $\sigma/\mu > Th_1$, decrease η multiplicatively till η_{min} i.e. $\eta = \eta/N, N > 1$.
2. If $Th_1 > \sigma/\mu > Th_2$, keep η constant.
3. If $\sigma/\mu < Th_2$, increase η multiplicatively till η_{max} i.e. $\eta = N\eta$.

Figure 2. Algorithm for η.

on the traffic characteristics and the conservatism requirements of the network domain. They should be chosen such that the variations in η are not too frequent. If the algorithm of Fig. 2 decides to change the value of η, *i.e.* the sampling frequency is changed, we propose to use the transition algorithm, as explained next. A decrease in the value of η implies more frequent sampling in the future. During the transition stage, when we do not have enough samples from the past (with proper spacing) for the prediction, we propose to obtain some predictions with the old far-spaced samples and then linearly interpolate between them to obtain closely spaced samples. On the other hand, if η is increased, we have to increase the sampling interval. This can be achieved by obtaining some predictions with the old near-spaced samples and then filtering to drop some of the obtained values. This algorithm introduces a smooth transition period where we still have enough samples for the linear regression. Once the value of η remains constant for a while, the system is able to achieve a stable operating point where the required past samples can be obtained directly from measurements.

Another method to predict the link utilization is to use linear regression for multi-step prediction with constant sampling frequency. In this method, τ is kept constant and a variable number of samples from the past are used in the linear regression to predict a variable number of samples in the future. This way the sampling frequency remains constant. This method is computationally less intensive as the prediction coefficients can be calculated by solving a Toeplitz matrix. However, this method suffers from the drawback of reduced efficiency as multi-step prediction is error-prone.

4 Implementation

In this section, we describe how MABE is implemented. The issues addressed include MRTG traffic rate calculation and modification of MRTG for reduced sampling time. In an SNMP network, the managed devices collect and store management information in MIBs and make it available to the managers through an agent running on the device. Each element in the MIB is identified by a sequence of numbers called Object Identifier (OID). The NMS can then retrieve specific information from the device using these identifiers. IETF has defined a standard [15] with specifications, grouping and relationships of managed objects in an SNMP compatible network. MRTG can be used to sample rates of almost any OID. By default, it is used to periodically fetch in-bound and out-bound traffic counters on the router interfaces and calculate the traffic rate on each one of them. These variables are available through the OIDs corresponding to in-bound and out-bound counters (in bytes) for each interface. MRTG stores the traffic rates for each interval of time, calculated by taking the difference of the counters and dividing by the interval length. MRTG database has a very simple layout. Each line has 5 values: timestamp, in-bound average rate, out-bound average rate, in-bound maximum rate and out-bound maximum rate. MRTG also keeps track of the counter values at the last sample in order to calculate the rates for the next period. Even though MRTG provides real-time available bandwidth measurements for a link, it may not be useful because of the 5 minute averaging intervals. Even if the RRDtool is used, the 300 seconds interval is hard coded in the MRTG source code. Patches are available to bring the interval detail down to 1 minute. However, in some cases, 1 minute might still be too coarse. We developed a patch to MRTG, called MRTG++, which provides up to 10 seconds detail. This provides a much finer granularity of measurements. First of all, the RRD database must be created with enough slots to store the larger amount of information. Then, the consolidation function parameters, i.e. how many samples the database will consider when calculating the average, must be adjusted for the new intervals. Our database is now able to store 10 seconds averages for up to 24 hours. Next step is to modify the script to send the correct queries to RRDtool when creating graphs. Since the intervals have changed, the scale and the set of data for the script must also be changed. Finally, MRTG++ must be run every 10 seconds to get the information from the routers.

5 Experimental Results

In this section, we describe the experiments we used to validate and quantify the utility of the MABE algorithm. First we describe the methodology for running the experiments. The proposed algorithm MABE for available bandwidth estimation on a link does not make any assumption about the traffic models. It works based on the measurements obtained from the network link. Thus, we do not need a network simulator. Instead, we can apply the algorithm to traffic traces obtained from real networks. In the following, we present the traffic profile predicted by MABE together with the actual profile. We do not present the predicted available bandwidth profile because that can be calculated by taking the difference of the link capacity and the utilization and thus it does not present significant information, when compared to the predicted utilization.

The choice of the thresholds Th_1, Th_2 and N, η_{min}, η_{max} used for updating the value of η 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 $Th_1 = 0.9$, $Th_2 = 0.5$, $N = 1.5$, $\eta_{min} = 10$, $\eta_{max} = 30$ and $p = 20$. The traffic trace used in the following results have been obtained from Abilene, the advanced backbone network of the Internet2 community, on March 13, 2002. In Fig. 3, the MABE algorithm is applied to the input traffic on the Cleveland router from the Cleveland-NYC link. As we can see, the utilization estimation obtained provides a conservative estimate. We propose the use of overestimation as a metric to quantitatively measure the performance of the proposed scheme MABE. For the case in Fig. 3, the mean overestimation is 1.31 MB/s which is very small. When compared with

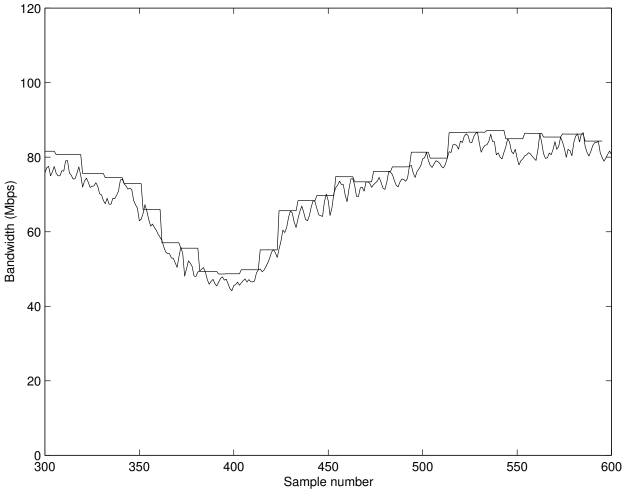

Figure 3. Input traffic on Cleveland router

MRTG with low granularity, we provide the available bandwidth estimates less frequently without a large compromise in the reliability of the estimate. In other words, the utilization profile obtained as a result of MRTG coincides with the actual traffic profile in Fig. 3, but MABE provides an estimate of the link utilization which is nearly accurate with a reduced computational effort.

6 Conclusions

We have proposed MABE, an algorithm to predict the available bandwidth on a link of an MPLS network. The algorithm is based on MRTG++ which has a small sampling time to query the routers. The algorithm modifies the MRTG++ sampling time based on the traffic characteristics to obtain a suitable prediction of the utilization. The predicted available bandwidth information is then useful for routing decisions and CAC for future bandwidth requests etc. The algorithm can be further refined by introducing a method to derive the threshold values based on the traffic characteristics.

Acknowledgments

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

References

1. T. Anjali, C. Scoglio, J. de Oliveira, I. Akyildiz, and G. Uhl, Optimal Policy for LSP Setup in MPLS Networks, *Computer Networks*, vol 39, no. 2, June 2002.
2. Z. Wang and J. Crowcroft, Quality-of-Service Routing for Supporting Multimedia Applications, *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1288–1234, September 1996.
3. R. Guerin, A. Orda, and D. Williams, QoS Routing Mechanisms and OSPF Extensions, *Proceedings of 2nd Global Internet Miniconference (joint with GLOBECOM'97)*, Phoenix, USA, November 1997.
4. J. de Oliveira, C. Scoglio, I. Akyildiz, and G. Uhl, A New Preemption Policy for DiffServ-Aware Traffic Engineering to Minimize Rerouting,, *Proceedings of IEEE INFOCOM'02*, New York, USA, June 2002.
5. K. Lai and M. Baker, Measuring Bandwidth, *Proceedings of IEEE INFOCOM'99*, New York, USA, March 1999.
6. V. Jacobson, *Pathchar*, ftp://ftp.ee.lbl.gov/pathchar/, 1997.

7. S. Keshav, A Control-Theoretic Approach to Flow Control, *Proceedings of ACM SIGCOM'91*, Zurich, Switzerland, 1991.
8. K. Lai and M. Baker, Nettimer: A Tool for Measuring Bottleneck Link Bandwidth, *Proceedings of 3rd USENIX Symposium on Internet Technologies and Systems*, San Francisco, USA, March 2001.
9. R. L. Carter and M. E. Crovella, Measuring Bottleneck Link Speeds in Packet-Switched Networks, *Boston University Technical Report BU-CS-96-006*, 1996.
10. V. Paxson, End-to-end Internet Packet Dynamics, *Proceedings of ACM SIGCOMM'97*, Cannes, France, September 1997.
11. M. Jain and C. Dovrolis, Pathload: A Measurement Tool for End-to-End Available Bandwidth, *A workshop on Passive and Active Measurements*, Fort Collins, USA, March 2002.
12. Iperf Tool, *http://dast.nlanr.net/Projects/Iperf/*.
13. Cisco IOS Netflow Website,
http://www.cisco.com/warp/public/732/Tech/netflow/.
14. J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, *Introduction to version 2 of the Internet-standard Network Management Framework*, IETF RFC 1441, http://www.ietf.org/rfc/rfc1441, April 1993.
15. K. McCloghrie and M. Rose, *Management Information Base for Network Management of TCP/IP-based Internets: MIB-II*, IETF RFC 1213, http://www.ietf.org/rfc/rfc1213, March 1991.
16. MRTG Website, *http://people.ee.ethz.ch/ oetiker/webtools/mrtg/*.
17. M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley and Sons, 1996.