ABEst: An Available Bandwidth Estimator within an Autonomous System

T. Anjali, C. Scoglio, L. C. Chen, I. F. Akyildiz, and G. Uhl *

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

* Swales Aerospace & NASA Goddard
Beltsville, MD 20705
Email: uhl@rattler.gsfc.nasa.gov

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 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 number of past samples that are used for prediction and also the the duration for which the prediction holds.**

Keywords—**Passive measurement, Available bandwidth, MRTG.**

I. INTRODUCTION

In recent years, there has been a tremendous growth in the Internet. New applications present new traffic patterns. The need for dynamic configuration of network devices to adjust to the evolving traffic has grown fast. Understanding the composition and dynamics of the Internet traffic is of great importance for network management. But observability of Internet traffic is difficult because of the network size, large traffic volumes, and distributed administration. The main tools towards controlling traffic to ensure appropriate Quality of Service (QoS) to all applications will be measurements of the network and simulations.

Measurement is necessary for the 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. Some QoS metrics have been defined by the IPPM [1] working group of IETF. Some of these can be measured in the core of the network and others 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. To obtain measured statistics from each network element is possible if individual users can monitor each such device. Due to security reasons, this is not possible in a network. Thus, 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 who can then make them publicly available. 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. 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 [2], routing (Shortest Widest Path [3], Widest Shortest Path [4]), LSP preemption [5], 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 ABEst, the Available Bandwidth Estimator. 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 algorithms.

II. 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. There are two definitions of available bandwidth. First one defines the available bandwidth on a single link (physical or virtual) of the network. This information can be used

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

for congestion avoidance, routing etc. Second one defines the available bandwidth of a route on the network which manifests as the 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. In [6], the authors have described a few bottleneck bandwidth algorithms. They can be split into two families: those based on pathchar [7] algorithm and those based on Packet Pair [8] 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. In [9], the authors have proposed another tool to measure bottleneck link bandwidth based on packet pair technique. Some other tools based on the same technique for measuring bottleneck bandwidth of a route have been proposed in [10], [11]. None of them measures the available bandwidth or utilization of a desired link of a network. In [12], 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. Another active approach to measure a path's available capacity is given in [13]. Iperf [14] from NLANR is another active approach that sends streams of TCP/UDP flows. Cisco has introduced the NetFlow [15] 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 a certain link of the network. He has access to the routers/switches of the network and can measure available bandwidth from the routers 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) [16] which 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) [17] 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 [18] is a tool based on SNMP to monitor the network links. 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 [18]. 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 the ABEst that is a linear regression algorithm to predict the utilization of a link. The algorithm is adaptive because a varying number of past samples can be used in the regression depending on the traffic profile. Using ABEst, we predict the utilization and the reliability interval for the prediction.

III. ABEST: AN AVAILABLE BANDWIDTH ESTIMATOR FOR A DIFFSERV NETWORK

In our approach, we concentrate on one 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.

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: Length of the averaging interval of MRTG,
- $L_\tau[k]$, $k \in \mathrm{N}$: 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. Note that we have not explicitly shown the $i-j$ dependence of the the defined variables. This is because

```
Algorithm:
1. At time instant $k$, available bandwidth measurement is desired.
2. Find the vectors $w_a, a \in [1,h]$ using covariance method given $p$ and the previous measurements.
3. Find $\left[\hat{L}_\tau[k+1], \ldots, \hat{L}_\tau[k+h]\right]^T$ from equation 1 and $[L_\tau[k-p+1], \ldots, L_\tau[k]]$.
4. Predict $A_h[k]$ for $[(k+1)\tau, (k+h)\tau]$.
5. At time $(k+h)\tau$, get $[L_\tau[k+1], \ldots, L_\tau[k+h]]^T$.
6. Find the error vector $[e_\tau[k+1], \ldots, e_\tau[k+h]]^T$.
7. Set $k = k+h$.
8. Obtain new values for $p$ and $h$.
9. Go to step 1.
```

Fig. 1. The ABEst Algorithm.

the analysis holds for any node pair independent of others. 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:

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

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 using covariance method [19]. We propose to dynamically change the values of p and h based on the traffic dynamics. This is what distinguishes our prediction method from other schemes based on linear regression.

The ABEst algorithm is given in Fig. 1. We define p_0 and h_0 as the initial values for p and h. In step 2 of the algorithm, we need to solve the covariance equations. They are given in a matrix form as $\mathbf{R}_L w_a = r_a$, for $a = 1, \ldots, h$, where

$$\begin{aligned}
\mathbf{R_L} &= \begin{bmatrix} r_L(0,0) & \cdots & r_L(0,p-1) \\ \vdots & \ddots & \vdots \\ r_L(p-1,0) & \cdots & r_{L(p-1,p-1)} \end{bmatrix} \\
w_a &= [w_a(0)\; w_a(1) \cdots w_a(p-1)] \\
r_a &= [r_L(0,-a)\; r_L(1,-a) \cdots r_L(p-1,-a)]
\end{aligned}$$

In order to derive the covariance from the available measurements, we estimate it as $r_L(n,m) = \sum_{i=k-N+p}^{k} L_\tau[i-n] L_\tau[i-m]$ where N affects the accuracy of the estimation, *i.e.*, more samples we consider,mo re precise the estimation is. The number of samples needed for a given n and N is $(n+N)$. Since the assumption about stationarity of the measurement sequence may not be accurate, we update the values of the covariance every time we change the value of p in step 8 of the algorithm. The solution of the covariance equations will provide w_a that can be used for predicting $\hat{L}_\tau(k+a), a = 1, \ldots, h$.

```
Algorithm:
1. Initialize $M = 0$ and $i = k$,
2. Obtain the prediction $\hat{L}_\tau[i]$,
3. Update $M = (1 - \frac{1}{i})M + \frac{1}{i}exp(s\tau\hat{L}_\tau[i])$,
4. If $i < k+p$,go to step 2,
5. $\alpha(s) = \frac{1}{s\tau}log(M); Stop.$
```

Fig. 2. Effective bandwidth estimator algorithm.

From the knowledge of the prediction coefficients w_a's, we predict $\left[\widehat{L}_\tau[k+1], \ldots, \widehat{L}_\tau[k+h]\right]^T$ using equation (1). Next step is to obtain an estimate of the available bandwidth for the interval $[(k+1)t, (k+h)t]$. This is done to obtain a single representativev alue valid for the whole interval. We can use two methods for the same, based on the requirements of the network operator. The representative available bandwidth value $A_h[k]$ can be given either as $A_h[k] = C - max\left\{\widehat{L}_\tau[k+1], \ldots, \widehat{L}_\tau[k+h]\right\}$ or by the use of effective bandwidth α as $A_h[k] = C - \alpha$. The former gives a strictly conservative estimate of the available bandwidth on the link for the entire duration. The latter gives a more realistic estimate which is tunable based on the network operators bandwidth requirements. Effective bandwidth [20] is a measure of the traffic stream that characterizes its steady state behavior and is given as

$$\alpha(s) = \lim_{t \to \infty} \frac{1}{st} \log E[e^{sL[0,t]}] \tag{2}$$

where s is the decay rate of queue size distribution tail probability and $L[0,t]$ is the total traffic load arrived during the time interval $[0,t]$. The equation (2) provides an effective bandwidth value between the peak and average traffic in $[0,t]$. An on-line block estimator for the effective bandwidth formulation is given in [21] and can be modified as given in Fig. 2.

After obtaining the actual load $[L_\tau[k+1], \ldots, L_\tau[k+h]]^T$ at time $(k+h)t$, we find the prediction error vector $[e_\tau[k+1], \ldots, e_\tau[k+h]]^T$ where each element is given as

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

Next, we propose the following algorithm to estimate new values for p and h based on a metric derived from the mean (μ) and standard deviation (σ) of error e_τ. The algorithm is given in Fig. 3.

In the algorithm, M_E is the maximum error value and we have introduced 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. They represent the confidence in the estimation procedure in terms of prediction errors.

```
Algorithm:
1. If $\sigma/\mu > Th_1$,decrease $h$ till $h_{min}$ and increase $p$ till $p_{max}$ multiplicatively.
2. If $Th_1 > \sigma/\mu > Th_2$, decrease $h$ till $h_{min}$ and increase $p$ till $p_{max}$ additively.
3. If $\sigma/\mu < Th_2$, then:
   (a) If $\mu > Th_3 * M_E^2$, decrease $h$ till $h_{min}$ and increase $p$ till $p_{max}$ additively.
   (b) If $Th_3 * M_E^2 > \mu > Th_4 * M_E^2$, keep $h$ and $p$ constant.
   (c) If $\mu < Th_4 * M_E^2$, increase $h$ and decrease $p$ till $p_{min}$ additively.
```

Fig. 3. Algorithm for h and p.

In this section, we have proposed an algorithm for estimating the available bandwidth of a link by dynamically changing the number of past samples for prediction and the number of future samples predicted with high confidence. The objective of the algorithm is to minimize the computational effort while providing a reliable estimate of available bandwidth of a link. It provides a balance of the processing load and accuracy. The algorithm is based on the dynamics of the traffic, *i.e.*, it adapts itself.

IV. IMPLEMENTATION

In this section, we describe how ABEst 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 [17] 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: time-stamp, 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,M RTG++ must be run every 10 seconds to get the information from the routers.

The NMS should decide the optimal MRTG measurement period based on the traffic characteristics, the required granularity for the measured values and the appropriate time-scale of the application utilizing the measured values.

V. EXPERIMENTAL RESULTS

In this section, we describe the experiments we used to validate and quantify the utility of ABEst algorithm. First we describe the methodology in running the experiments. The proposed algorithm ABEst 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 ABEst 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,etc. and h_{min}, p_{max} used for updating the values of p and h in Section III 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 $Th_1 = 0.9$, $Th_2 = 0.7$, $Th_3 = 0.5$, $Th_4 = 0.3$ and $h_{min} = 10$, $p_{max} = 50$. All the traffic traces used in the following results have been obtained from Abilene, the advanced backbone network of the Internet2 community, on March 13, 2002. In Fig. 4, the ABEst algorithm is applied to the input traffic on the Atlanta router of the Atlanta-Washington D.C. link. In Fig. 5,

same is done for the input traffic on the Cleveland router from the Cleveland-NYC link. In both cases, the first curve shows the actual traffic profile and the other two curves show the prediction by utilizing ABEst. The first of the two utilizes the peak-based estimation whereas the second utilizes the effective bandwidth-based estimation. As we can see, in both cases, the utilization estimation obtained by taking the peak prediction provides a conservative estimate, whereas the estimation using

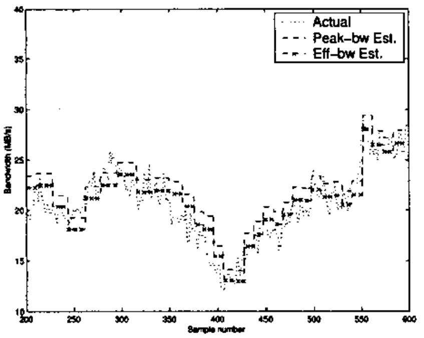

Fig. 4. Input traffic on Atlanta router

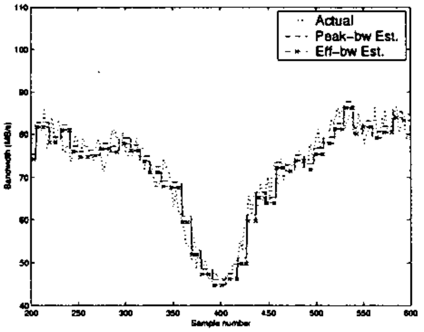

Fig. 5. Input traffic on Cleveland router

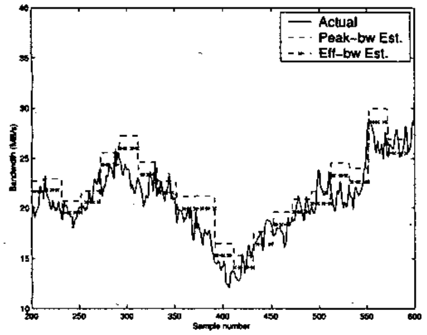

Fig. 6. Input traffic on Atlanta router ($h_{min} = 20$)

the effective bandwidth provides an estimate for lower resource utilization. Also, when h_{min} is increased, the estimation becomes worse (see Fig. 6), in the sense that it does not follow the sequence closely but is still very conservative. We propose the use of overestimation as a metric to quantitatively measure the performance of the proposed scheme ABEst. For the case in Fig. 4, the mean overestimation is 1.31 MB/s for the peak estimation procedure whereas it is 0.73 MB/s for the effective bandwidth estimation procedure. Similar values are obtained for the case depicted in Fig. 5.

When compared with MRTG, 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 Figs. 4 and 5, but ABEst provides an estimate of the link utilization which is nearly accurate with a reduced computational effort.

VI. CONCLUSIONS

Weha ve presented an algorithm to estimate the available bandwidth on a link. 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 number of past samples that are used for prediction and also the the duration for which the prediction holds. The algorithm can be further refined by introducing a method to derive the threshold values based on the traffic characteristics.

REFERENCES

[1] IP Performance Metrics Working Group, *http://www.ietf.org/html.charters/ippm-charter.html.*

[2] T. Anjali, C. Scoglio, J. de Oliveira, I. Akyildiz, and G. Uhl, "Optimal Policy for LSP Setup in MPLS Networks,"*C omputer Networks*, vol. 39, no. 2, pp. 165–183, June 2002.

[3] 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.

[4] R. Guerin, A. Orda, and D. Williams, "QoS Routing Mechanisms and OSPF Extensions," in *Proceedings of 2nd Global Internet Miniconference (joint with GLOBECOM'97)*, Phoenix, USA, November 1997.

[5] J. de Oliveira, C. Scoglio, I. Akyildiz, and G. Uhl, "A New Preemption Policy for DiffServ-Aware Traffic Engineering to Minimize Rerouting,," in *Proceedings of IEEE INFOCOM'02*,N ew York, USA, June 2002.

[6] K. Lai and M. Baker, "Measuring Bandwidth," in *Proceedings of IEEE INFOCOM'99*,N ew York, USA, March 1999.

[7] V.J acobson, *Pathchar*, ftp://ftp.ee.lbl.gov/pathchar/, 1997.

[8] S. Keshav, "A Control-Theoretic Approach to Flow Control," in *Proceedings of ACM SIGCOM'91*, Zurich, Switzerland, 1991.

[9] 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.

[10] R. L. Carter and M. E. Crovella, "Measuring Bottleneck Link Speeds in Packet-Switched Networks," *Boston University Technical Report BU-CS-96-006*,1 996.

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

[12] M. Jain and C. Dovrolis, "Pathload: A Measurement Tool for End-to-End Available Bandwidth," in *A workshop on Passive and Active Measurements*,F ort Collins, USA, March 2002.

[13] S. Banerjee and A. Agarwala, "Estimating Available Capacity of a Network Connection," in *Proceedings of IEEE ICON'00*, Singapoer, Singapore, September, 2000.

[14] Iperf Tool, *http://dast.nlanr.net/Projects/Iperf/.*

[15] Cisco Netflow, *http://www.cisco.com/warp/public/732/Tech/netflow/.*

[16] 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.

[17] 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.

[18] MRTG Website, *http://people.ee.ethz.ch/ oetiker/webtools/mrtg/.*

[19] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley and Sons, 1996.

[20] F. Kelly, *Stochastic Networks: Theory and Applications*, Royal Statistical Society Lecture Notes Series. 4. Oxford University Press, 1996.

[21] J. Yang and M. Devetsikiotis, "On-line Estimation, Network Design and Performance Analysis with EffectiveB andwidths," in *Proceedings of 17th International Teletraffic Congress 2001*, Salvador, Brazil, September 2001.