Abstract—Bandwidth is one of the most common terms in digital communication as it represents the supported amount of data that a link or network path can send per unit of time. Multimedia application’s performance is directly impacted by available bandwidth because these applications have diverse Quality of Service (QoS) requirements. Internet is now a necessary technology in our daily life, and because of mobile communication systems people can access the network anytime and anywhere, resulting in an increase in the number of users. Such evolution brings new services that will require more data rates. As a result of this load, some services and users in the network will demand for certain data rate which cannot be guaranteed. In the past, there have been several proposals that provide some mechanism for bandwidth estimation based on the size of web message and TCP/IP stack. However, these solutions tend to provide unreliable bandwidth estimation due to TCP/IP buffering. This paper proposes a bandwidth estimation mechanism (BEM) that is capable of providing reliable estimation of the available bandwidth to users or applications. The experimental results validate the intrusiveness and the ability of the proposed BEM to provide accurate bandwidth estimates.

Keywords—Quality of Service, Iterative Probing, Available Bandwidth, Bandwidth Estimation Techniques

I. INTRODUCTION

The evolution in technology has significantly increased the users of mobile communication. This has then led to the rapid development of new services particularly in web-based applications and hence demands more data rates [1, 2]. When each user and service starts utilizing the network by sending their traffic across the network path, the network will experience high load. As a result, certain services and users in the network will require more bandwidth. Therefore, comprehending the dynamic characteristic of the end-to-end available bandwidth (ABw) is crucial to provide better services for web-based users. Nevertheless, the scale of the different systems, the variation of the traffic characteristics and the multitude of network technologies make the end-to-end ABw characterization a challenging task.

In the past, there have been several proposals [3, 4, 5, 6, 7] that provides different QoS requirements for users and applications. However, these proposals unable to provide the end-hosts or applications a mechanism that can dynamically detect and adapt the variability of network conditions. This issue has triggered many researchers to focus on designing algorithms and tools that can monitor the network state through end-to-end measurement [8]. The ability for an application to adapt its behavior to changing network conditions depends on the underlying bandwidth estimation mechanism (BEM) that the application or transport protocol uses. As such, accurate bandwidth estimation algorithms and tools can benefit a large class of data-intensive and distributed scientific applications [9].

In our proposed solution, we incorporate bandwidth estimation tools called pathload into our web-based e-learning system to monitor network conditions, which serves as an input for making intelligent choices for downloading activities. Given the current knowledge of available bandwidth allows better information regarding bandwidth utilizations and path latencies is critical for proactive and reactive resource management. The rest of the paper is organized as follows. Section II describes the preliminaries of bandwidth estimation, self-loading probe technique and related works. Section III explains the Pathload modifications for BEM. Section IV provides the deployment and implementation of BEM in web-based applications and Section V examines the experimental results. Finally, we briefly discuss future work and conclude in Section VI.

II. PRELIMINARIES

A. Bandwidth Estimation

Bandwidth estimation refers to the end-to-end measurement of bandwidth-related metrics, such as capacity, available bandwidth and bulk TCP transfer capacity. Several applications can benefit from knowing the bandwidth characteristics of their network paths. For example, peer-to-peer applications, intelligent routing systems, end-to-end admission control, and multimedia streaming applications can all benefit from bandwidth estimation techniques [9]. Available bandwidth is the maximum unused bandwidth at a link or end-to-end path in a network, therefore available bandwidth depends on the link capacity and the traffic load, because of such dependency the available bandwidth is typically a time varying metric [9]. In a given hop i the available
bandwidth \( A_i \) of end-to-end link over a certain time interval is given by the unutilized part of the capacity [9]:

\[
A_i = (1 - U_i)C_i
\]

(1)

where \( U_i \) is the average utilization of hop \( i \) in the given time interval, and the \( C_i \) is the capacity of hop \( i \). By extending the available bandwidth definition to an \( H \) hop path, the available bandwidth of the end-to-end path, \( A \), will be the minimum available bandwidth of all \( H \) hops:

\[
A = \left( \min_{i=1}^{H} A_i \right)
\]

(2)

The term tight link is used to describe the minimum available bandwidth of the end-to-end path. On the other hand, the term narrow link used to describe the minimum capacity of the end-to-end path. The tight and narrow links are both indicate the bottleneck of a network but they are not necessarily at the same hop [9].

**B. Self-Loading Probe Technique**

Available Bandwidth Estimation (ABE) techniques can be classified into three main groups: Direct Probing, Iterative Probing and Mixed techniques. The main characteristic of Direct Probing is that every probing stream provides an estimation of the ABw. If the tight-link capacity is known or can be estimated, an estimation of the ABw can be obtained by measuring the output rate or estimating the cross-traffic rate. Iterative Probing techniques are based on the fact that rates greater than the ABw increases the queuing delay and, therefore, reduces the output rate. This technique determines whether the input rate exceeds the ABw by sending a stream of packets and studying the behavior of the queuing delay or measuring the output rate. The stream is sent iteratively, sometimes changing some of its parameters, in order to be more accurate. Example of bandwidth estimation tools that employed iterative probing techniques are Train of Packet Pairs (TOPP) [10, 11] and Self-Loading Periodic Streams (SLoPS) [12]. The Mixed techniques combine both Direct and Interactive Probing. In our approach, we incorporate self-loading probe technique in our bandwidth estimation mechanism.

Self-loading techniques measure the available bandwidth of the end-to-end network path which is also known as self-induced congestion [13]. There are many tools that apply a variety of self loading techniques, for example Pathload [12], Packet Transmission Rate (PTR) [14] and pathChirp [13]. Self-loading techniques probe the end-to-end network path using multiple rate traffic. When the probing rate \( R \) becomes larger than the available bandwidth \( A \), the probing stream packets \( K \) become queued at the tight link router, therefore there will be an increased delay \( D_t \) of the \( t \)th packet on the receiver node \( RCV \). On the other hand, if the probing packets \( R \) go through the tight link and did not encounter any delay at the receiver side that means the available bandwidth is greater than the probing rate. The available bandwidth can be spotted at turning point of the probing rate where the queue delay start to increase and that is done by analyzing the delay at the receiver side.

**C. Pathload Algorithm**

This section briefly describes the Pathload algorithm that estimates the available bandwidth of an end-to-end path. A more detail information regarding Pathload can be found in [12, 18]. The algorithm is divided into two phases, the probing phase and the measurement phase. These two phases are separately described in the following two subsections.

i. Probing Phase

Pathload sends \( N \) streams of a periodic UDP packet stream of rate \( R_n \). Each stream consists of \( K \) packets of size \( L \) bits, transmitted periodically in every \( T \) seconds. All streams in a set have the same rate \( R_n = L/T \).

ii. Measurement Phase

Measurement starts with sending the first periodic packet stream of UDP packets from the sender to the receiver. The sender time stamps the send time and the receiver time stamps each probe packet when it arrives. Pathload measures the variation of One Way Delays (OWD) by subtracting the receiver timestamp with the arrival time (3) and computes relative one-way delay for each packet based on (4).

\[
D_k = T_{rcv} - T_{snd}
\]

(3)

\[
\Delta D_k = D_{k+1} - D_k
\]

(4)

If the probing rate of the packet stream is higher than available bandwidth, the OWD delay of a periodic packet stream shows increasing trend (more stable). Otherwise, if the rate is lower than available bandwidth, OWD will show decreasing delay (less stable). Pathload uses two complementary statistics: Pairwise Comparison Test (PCT) and PDT (Pairwise Difference Test), given in (5) and (6) to determine whether the OWD values are increasing or decreasing.

\[
PCT = \frac{\sum_{k=2}^{R} I(D_k > D_{k-1})}{R-1}
\]

(5)

\[
PDT = \frac{\sum_{k=2}^{R} |D_k - D_{k-1}|}{\sum_{k=2}^{R} |D_k - D_{k-1}|}
\]

(6)

The algorithm terminates when \( R_{max} - R_{min} \leq \chi \), where \( \chi \) is the user-specified estimation resolution.

**D. Network Monitoring**

Currently there are two techniques for monitoring network. The two common techniques are the passive and active monitoring. Passive monitoring relies on internetworking devices such as routers and switches or end node hosts to monitor the traffic. This technique uses
the trace history of existing data transfers to measure the available bandwidth, analyze network flows or provide current network traffic statistics. Active measurement, in contrast depends on the sending of artificial probe packets from the sender to the receiver via a network. The basic strategy of injecting a set of probe packets into the network is to infer network condition of the path. This method is also known as end-to-end measurement methods because it needs access to two hosts: a traffic source and receiver.

E. Related Work

A considerable amount of research work has been published on bandwidth estimation. Some research efforts [15, 16, 17] have focused on passive monitoring to observe changes in a network condition. Numerous works [11, 13, 14, 18, 19] have been conducted on active monitoring to estimate end-to-end available bandwidth or link capacity by injecting a set of probe packets. Several attempts [20, 21, 23, 24] have been made to incorporate active network monitoring and web-based application together. For instance, in [21] they built a web server that capable of adapting multimedia content based on the clients’ device capabilities and network condition. However, in their solution, they only provided static information regarding the network bandwidth without incorporating any mechanism for detecting the actual bandwidth.

One of the preliminary works on integrating active network monitoring into web-based application was undertaken by Ma et al. [20]. They implemented two methods for measuring network bandwidth: the size of web message and TCP/IP stack. In the first method, the web server would take the size of a web message and divide it by the amount of time it takes to complete the transmission. Due to TCP/IP buffering, they found that this method has a tendency to overrate the bandwidth. They also discovered that the second method tends to underrate the bandwidth when available bandwidth information is retrieved during the establishment of TCP/IP connection. Due to TCP/IP’s “slow start” phase and the short-lived nature of HTTP, bottleneck would never occur and thus the actual available bandwidth is underestimated. Having these margins, accurate bandwidth information is still vague in today’s web-based applications. Therefore, a mechanism is needed to measure and detect the available bandwidth accurately across a network path in order to uphold a more worthy web-based application services for users.

III. PATHLOAD MODIFICATION

This section explains how and where we have modified Pathload in order to deploy our proposed BEM. To automate the bandwidth estimation loops, we added the loop command for \( \text{Loop}=0; \text{loop} < \text{MaxLoop} ; \text{loop}++ \) in pathload_snd.c. This instruction will capture the maximum number of loops \( \text{MaxLoop} \) as well as the IP address \((\text{hostname})\) of the sender node \( \text{SND} \) from a text file. Then \( \text{SND} \) starts to establish connection with the receiver node \( \text{RCV} \) by executing this command: 

```
./pathload_rcv -s %s”, hostname.
```

The main purpose of BEM is to provide information regarding the ABw to users or applications in a database server which can be accessed by the web server. Therefore, we have modified some parts of the original code to allow ABw estimation results to be stored in the database.

IV. THE DEPLOYMENT AND IMPLEMENTATION OF BEM IN WEB-BASED APPLICATION

This section discusses the network infrastructure that has been implemented in order to develop the proposed BEM in web-based application. The infrastructure and its various components are illustrated in Figure 1. We briefly describe the function of each component as follows:

- Web Server. The web server has been implemented using PHP 5.2.9, MySQL 5.1.33 and Apache Web Server 2.2.11. This server processes the clients’ activity, controls and maintains the database and provides all the functionality of e-learning applications.
- Bandwidth Estimator (\( \text{RCV} \)). The receiver performs a trend test on one-way delays of packets in the train after all packets in a train are received. The trend of the fleet will be determined based on the fraction of \( n \) trains that is of increasing or no-trend. When the trend information for a fleet is sent back to the sender, BEM searches for an available bandwidth region by increasing or decreasing the input probing rate \( R \) at the sender in a binary search fashion based on the trend information.
- Probe Packet Generator (\( \text{SND} \)). This host which is also known as the sender, probes the network path by sending the first periodic packet stream of UDP packets from to the receiver and timestamps each packet. It uses adjustable input rates of packet-trains to infer available bandwidth of the path and sends a fleet of \( n \) probe-packet trains, each of which consists of \( N \) back-to-back packets with inter-packet spacing \( x \). The sender time stamps the send time while the receiver time stamps the probe packet reception. The time stamp difference, defined as the \( \text{OWD} \) is calculated for each packet \( (D_1, D_2, \ldots , D_{n}) \).

We have incorporated BEM in a web-based application, developed specifically for e-learning purposes. Figure 2 shows the user interface of our web-based e-learning system. Whenever students need to download files from the application, they can first check the ABw indicator. They can decide to download the required files if the ABw is adequate for downloading or postpone to a later time.
V. EXPERIMENTAL RESULTS

In this section we describe the network testbeds, the evaluation metrics, the experimental setup and validation methodology used in our experiments. We have performed several experiments to validate our proposed method. The first aims at evaluating the proposed BEM under reproducible and controllable conditions. The second experiment is to evaluate the intrusiveness of BEM.

A. Performance Metrics

We use two performance metrics, accuracy and intrusiveness, to evaluate the proposed BEM. To evaluate estimation accuracy, the true available bandwidth of the wired networks under different configurations is needed which can be obtained from the network setup or measurement on the router such as the Multi Router Traffic Grapher (MRTG) [25]. MRTG which is based on the Simple Network Management Protocol (SNMP)-derived values is the most accurate way to verify the output of available bandwidth estimation tools [19, 26, 27]. In order to quantify the accuracy the bandwidth estimation, we compute its AB estimation error as the difference between the estimated AB and the actual AB. We use the metric relative measurement error to evaluate available bandwidth measurement accuracy. It is defined as:

\[
\text{Relative Error} = \frac{AB_{\text{estimated}} - AB_{\text{actual}}}{AB_{\text{actual}}} \quad (7)
\]

where, \(AB_{\text{estimated}}\) is the available bandwidth estimates generated by BEM, \(AB_{\text{actual}}\) is the actual available bandwidth detected by MRTG.

Intrusiveness is one of the main concerns for evaluating the bandwidth estimation techniques. We define intrusiveness based on whether the performance of web applications can be impacted by the heavy probing traffic. Therefore, low intrusiveness is critical for reducing the impact caused by the probing traffic itself. Intrusiveness is defined as:

\[
\text{Intrusiveness} = \left( \frac{T_{\text{BEM}}}{AB} \right) \times 100 \quad (8)
\]

where, \(T_{\text{BEM}}\) is the amount of traffic (in bits) sent by BEM during the probing time. \(AB\) is the actual available bandwidth detected by MRTG.

B. Experimental Setup

We verify the measured available bandwidth by comparing it with MRTG and by measuring how BEM responds to induce changes in available bandwidth. We repeatedly run BEM together with MRTG for a total monitoring time of 1 hour. Both BEM and MRTG use the same machines. Since MRTG data provides an average over a period of 5 minutes, we smooth the measurements by taking the average output of each tool over similar 5 minutes periods.

It is essential to use test traffic that closely simulates
traffic on real networks and reproduces its most critical characteristics. In our study we conducted two series of laboratory experiments using Nuttcp to generate and send different volume of contending traffic to the network. In order to simulate the dynamic traffic conditions that characterize real Internet links, we varied traffic load level from 0%, 25%, 50%, and 75%.

C. Testbed Topology

We evaluated the ability and efficiency of the proposed BEM to compute the available bandwidth in the laboratory settings. To study the accuracy and intrusiveness of the bandwidth estimation in realistic scenarios, we ran experiments under different load conditions and path configurations: 100 Mbps and 10 Mbps. The testbed topology is shown in Figure 3 and 4. In order to simulate the dynamic traffic conditions that characterize real Internet links, we varied traffic load level from 0%, 25%, 50%, and 75%.

The two Dell Dimension PCs (PC3 and PC4) running under Ubuntu 7.08 are used as the sender and the receiver to generate probe traffic. Another two PCs (PC1 and PC2) run Nuttcp traffic generator to generate controllable cross-traffic. The probing sender is connected to a switch. The receiver is connected via a 100Mbps Ethernet or Gigabit switch. The probing traffic traverses the switch, the hub/switch and finally arrives at the respective receiver. We can observe that this testbed has one bottleneck link at the 10Mbps or 100Mbps Ethernet hub/switch. The cross traffic stream and the probing stream compete for the 10Mbps or 100Mbps bottleneck capacity.

D. Accuracy

We evaluated the ability of the proposed BEM to compute the ABw under controlled laboratory settings. BEM estimated the ABw reasonably well in both 10 and 100 Mbps tight links. The discrepancy between its readings and actual available bandwidth was less than 20% in most cases. Figures 5(a) and 5(b) show that BEM successfully estimates the ABw. To some extent, this accuracy can be explained based on the Capacity Estimation technique used which is Packet Pair Dispersion. When there is cross traffic in the path, the probing packets can experience additional queuing delays due to cross traffic. The cross-traffic can either increase or decrease the dispersion distribution. If other traffic queues between the probe packet at a certain link, the dispersion increases leading to capacity underestimation, while packets in front of the first packet can make it queue and wait for the second packet, resulting in capacity overestimation.

E. Intrusiveness

The intrusiveness of a bandwidth estimation mechanism is evaluated on its capacity of producing a result while sending the minimum amount of extra traffic. The main reason behind this experiment is to find out whether the injection of extra data for probing causes a significant decrease in the AB. Table I shows the results of a 60-minutes experiment, performed similarly with the experiment 1. We monitored the 5-minutes average AB in the path using MRTG, and used Nuttcp tool to generate cross-traffic that closely simulates traffic on real network. In average, the total amount of network probe traffic sent by BEM for each run-times is about 500 Kbps.

<table>
<thead>
<tr>
<th>Tight Link Capacity (Mbps)</th>
<th>0%</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6.42</td>
<td>9.06</td>
<td>13.16</td>
<td>29.24</td>
</tr>
<tr>
<td>100</td>
<td>0.6</td>
<td>0.82</td>
<td>1.47</td>
<td>3.34</td>
</tr>
</tbody>
</table>

The proposed BEM intrusiveness is between 6 to 30% for 10 Mbps tight links and 0.6 to 4% in 100Mbps tight links.

![Figure 5. Available bandwidth measurement error for a) 10 Mbps and b) 100Mbps](image-url)
The results of this experiment indicate that the BEM intrusiveness slightly increases with the ABw (when the cross-traffic decreases) and can reach up to 30% load when the ABw is lower. This indicates that BEM is non-intrusive because its average probing traffic rate during the measurement process does not constitute a significant load on the network which is in agreement with Murray’s et al. [9] statement.

VI. CONCLUSION

In this paper, we have successfully incorporated BEM in a web-based application for wired network and made the estimated results accessible to the end users. Although through some experiments, we have tested that the proposed BEM is non intrusive and capable of providing accurate estimation; further experimental investigations are needed to produce results that are close to the “ground truth” especially for wireless network. The current techniques of bandwidth estimation cannot perform as expected in wireless networks due to the channel fluctuations that cause changeability in wireless capacity and available bandwidth. Furthermore, channel impairment in signals transmission such as noise, interference and distortion could limit the accuracy of bandwidth estimation. Therefore, in our future works we will incorporate a suitable bandwidth estimation algorithm for wireless network.

REFERENCES