Efficient Bandwidth Estimation for HTTP Adaptive Streaming

Dooyeol Yun †, Kwangsue Chung ‡ and Jinpyo Hong †††
Department of Electronics and Communications Engineering, Kwangwoon University†
School of Information and Communications of Engineering, Hankuk University of Foreign Studies‡‡
Seoul, Korea
†dyoon@cclab.kw.ac.kr, ‡kchung@kw.ac.kr, and ††jphong@hufs.ac.kr

Abstract—As the multimedia devices are becoming more diverse and the HTTP adaptive streaming techniques grow in popularity, the transmission of adequate video quality becomes increasingly important. HTTP adaptive streaming requires accurate information about available bandwidth of client for quality adaptation. Accurate estimation of available bandwidth is difficult, as a suitable approach must address three challenges. First, it must be accurate enough to make decisions about the admission of high-throughput high-quality streams. Second, it must be fast enough to determine the video quality. Last, it should be non-intrusive to streaming traffic. In this paper we propose a new bandwidth estimation scheme, called Efficient Bandwidth Estimation (EBE), for HTTP adaptive streaming. To provide information which is needed to determine the representation level of HTTP adaptive streaming, the EBE scheme uses the difference between the bitrate of representation levels. We show that our approach is accurate, fast, and non-intrusive compared with other schemes.

Keywords—Available Bandwidth, Bandwidth Estimation, HTTP Adaptive Streaming

I. INTRODUCTION

Due to the explosive growth of multimedia devices such as mobile phone, pad, game console and digital TV, there have been significantly increasing demands for multimedia streaming application. HTTP adaptive streaming is a popular way to transmit the multimedia content to users [1], [2]. HTTP adaptive streaming has been spreading as a form of Internet video delivery with the recent deployments of proprietary solutions such as Apple HTTP Live Streaming, Microsoft Smooth Streaming, and Adobe HTTP Dynamic Streaming. In the meantime, standardization of HTTP adaptive streaming has also made great progress with the recent completion of technical specifications by various standards bodies including the Third Generation Partnership Project (3GPP), Motion Picture Experts Group (MPEG), and Open IPTV Forum (OIPF) [3]. Going forward, future deployments of HTTP adaptive streaming are expected to converge through broad adoption of theses standardized solutions referred to as Dynamic Adaptive Streaming over HTTP (DASH). The benefits of HTTP adaptive streaming include its scalability, high performance and easy deployment, especially the possibility to reuse the already deployed HTTP infrastructure.

HTTP adaptive streaming provides the ability to the client to fully control the streaming session; that is, it can intelligently manage the on-time request and smooth playout of the sequence of segments, potentially adjusting bit rates or other attributes. The client can automatically choose initial content rate to match initial available bandwidth without requiring negotiation with the streaming server and dynamically switch between different bit rate representations of the media content as the available bandwidth changes. Therefore the HTTP adaptive streaming requires accurate information about available bandwidth of client for quality adaptation. In order to estimate a bandwidth, many researches have been developing. We refer to related researches in section II. Although many bandwidth estimation schemes exist for the Internet, none of them fulfills all of the requirements for use in the HTTP adaptive streaming. Among these requirements are the following. The scheme must:

1. Be accurate enough to make informed decisions about video quality adaptation of HTTP adaptive streaming.
2. Have a short measurement time. It should have low convergence time from an end user perspective, and it should be fast enough to react to major changes in the network traffic.
3. Be non-intrusive. It should not disrupt streaming traffic which has a big volume.

In this paper, we propose a new bandwidth estimation scheme called an Efficient Bandwidth Estimation (EBE) scheme to provide the information which is needed to determine the quality of video streams. The proposed scheme calculates the link capacity at initial buffering time. The EBE sends the probing packets based on the difference between the bitrate of representation levels to estimate the available bandwidth after the initial buffering time. Also, it considers the wireless delay such as interframe space in estimating the bandwidth.

The rest of the paper is organized as follows. In the next section, we review some of the related works. In Section III, we describe the concepts and algorithms introduced in the EBE. Simulation results and conclusions are given in Section IV and Section V, respectively.
II. RELATED WORKS

A. Bandwidth Estimation Schemes for Wired Networks

There are many researches about bandwidth estimation in wired networks. Pathload is an active measurement scheme that estimates the available bandwidth of a network path [4]. The main idea is that the one-way delays of a periodic packet stream show increasing trend, when the stream rate is larger than the available bandwidth. A sender process and a receiver process are running in pathload. Periodic packet streams are achieved by UDP. On the other hand, TCP connection is used as control channel between two end-points. This algorithm has a very good approach for estimating bandwidth. It uses equation of \( R = L/T \) where \( T \) is transmission period, \( L \) is packet size and \( R \) is the transmission rate.

Pathchirp is based on the concept of self-induced congestion and uses the self-loading periodic streams methodology called self-loading packet chirps [5]. Pathload uses adaptive search method, for the reason it has a long convergence time. Both algorithms use different methodology, and their outputs are different from each other. The idea of pathchirp is to use exponentially spaced highly efficient chirp probing train. The task is to estimate the available bandwidth over the path based on queuing delays of chirp probe packets transmitted from sender to the receiver and then conducting a statistical analysis at the receiver. As considered in most researches, congestion is supposed to occur at the edge of the network close to the source or receiver. Data packets may encounter two congested queues, one on each end of their paths. Pathchirp is a scheme that presents a robust solution for these kinds of congested queues.

B. Bandwidth Estimation Schemes for Wireless Networks

Bandwidth estimation in wireless network is more challenging issue due to unstable and time-varying condition. Wireless networks have variable conditions such as wireless link rate adaptation, transmission retries, contention and variable packet rate error. Furthermore, the wireless channel is also a shared-access medium, and the available bandwidth also varies with the number of hosts contend for the channel. There are few major bandwidth estimation methods available in wireless networks. Wbest uses probing packet-pair dispersion solution to estimate the effective capacity over a flow path where the last hop is a wireless LAN [7]. In addition, a packet-train technique is used for estimating achievable throughput to infer the available bandwidth. The advantage of wbest stems from avoiding a search algorithm to detect the available bandwidth by statistically detecting the available fraction of the effective capacity to mitigate estimation delay and impact of random wireless channel errors.

Most of the existing wireless bandwidth estimation solution focus on either probing techniques or cross-layer techniques and require either significant bandwidth resources or protocol modifications. To alleviate these problems, analytical Model-based Bandwidth Estimation algorithm (MBE) is proposed for multimedia services over IEEE 802.11 networks [8]. The MBE module for available bandwidth estimation is developed on the basis of novel transmission control protocol/user datagram protocol throughput models for wireless data communications.

Most of early bandwidth estimation techniques seek to provide accurate bandwidth information at the cost of long convergence times and high intrusiveness. Especially, there is no bandwidth estimation scheme for HTTP adaptive streaming to provide the information used for representation level adaptation.

III. EFFICIENT BANDWIDTH ESTIMATION SCHEME

This section proposes the Efficient Bandwidth Estimation (EBE) scheme to provide the information required to determining the representation level. To estimate available bandwidth accurately, the proposed scheme calculates the capacity on the initial buffering time. It also estimates the bandwidth based on the difference between the bitrate of representation levels to guarantee the low convergence time and non-intrusiveness. We assume that the probing packets do not overflow any of the router queues along the flow path. The possibility of queuing loss is reduced by limiting the number of probing packets sent into the network.

A. Link Capacity

The EBE scheme calculates the link capacity on the initial buffer time before the start of playback. It does not interfere with smooth playback of video streaming. Estimation result of packet pair \( i \) can be calculated as in (1).

\[
C_i = \frac{L}{T_i} = \frac{L}{D_i^1 - D_i^2}
\]  

(1)

\( C_i \) is the Capacity, \( L \) is the packet size, and \( T_i \) denotes the one way delay of each packet in the packet pair. To minimize the impact of crossing traffic, EBE uses the median of the \( n \) packet pair capacity estimates to approximate \( C_v \) in the estimation time period:

\[
C_v = median(C_i), \quad i = 1,...,n
\]  

(2)

B. Modeling the Probing Rate for HTTP Adaptive Streaming

Fig. 1 shows the inefficiency of traditional bandwidth estimation scheme which does not consider the application characteristics such as the difference between the representation bitrates of HTTP adaptive streaming. To reduce the overhead of measurement, probing rates should be at the minimum between the representation bitrates.

![Fig. 1. Comparision of the inefficient and efficient probing rates](image-url)
Fig. 2 shows the regions of the adaptation rule. There are two regions: Transition (T), Keep (K). If the estimated bandwidth is in T region, HTTP adaptive streaming adjusts the representation level. Otherwise, it maintains the current representation level. $\Delta$ denotes a barrier factor to prevent the frequent adaptation which can degrade the quality of service.

To reduce the steps of probing rates and to position the probing rates between the quality bitrates as shown in Fig. 2, we design the probing rates which can be calculated as (3) and (4) based on the pathChirp.

$$PR_i = \frac{L}{T_i \prod_{j=0}^{\gamma_j}} \quad , \quad i=1,...,n$$  \hfill (3)  

$$PR_0 = RL_0 = \frac{L}{T_0 \gamma_0}$$  \hfill (4)  

$PR_i$ represents a probing rate of $i_{th}$ probe train and $\gamma_j$ is a spread factor of $j_{th}$ train (0, 1). $RL_0$ is lowest representation bitrates. $T_0$ is calculated on the basis of $RL_0$, $L$, and $\gamma_0$. $DPR_i$ in (5) is the difference between $i_{th}$ and $(i+1)_{th}$ train probing rates. $DRRL_k$ in (6) is the difference between $i_{th}$ and $(i+1)_{th}$ representation bitrates.

$$DPR_i = PR_i - PR_{i+1}$$  \hfill (5)  

$$DRRL_k = RL_{k+1} - RL_k$$  \hfill (6)  

$RL_k$ is the representation bitrate of $k_{th}$ level which is predefined. The EBE scheme adjusts the spread factor to guarantee the condition as shown in (7).

$$\frac{RL_k}{2} \leq DPR_i < RL_k$$  \hfill (7)  

C. Wireless Network Delay

The EBE scheme estimates the bandwidth based on the packet delay. In wireless networks, it should consider the wireless network delay. We consider the IEEE 802.11 WLAN. During the Round-Trip Time (RTT), the receiver can be in one of the following states: idle, successful transmission, or retransmission. The delay for successful transmission is denoted as $D_{\text{succ}}$. We derive (8) and (9), to present the 802.11 MAC layer delay for basic access mode $MAC\_\text{Delay}_{\text{basic}}$ and RTS/CTS mode $MAC\_\text{Delay}_{\text{RTS}}$, where distributed interframe space (DIFS) and short interframe space (SIFS) are contention control parameter defined in 802.11 MAC specifications. $MAC\_\text{ACK}$ represents the acknowledgment packet sent by the MAC receiver.

$$MAC\_\text{Delay}_{\text{RTS}} = DIFS + 3\text{SIFS} + RTS + CTS + MAC\_\text{ACK}$$  \hfill (9)  

Combining (8) and (9), the delay for successful transmission is given by (10), where TCP_ACK represents the acknowledgement packet sent by the TCP receiver. Note that the propagation delay is the time taken to transmit data, which includes the original data packet plus the protocol header.

$$D_{\text{succ}} = D + \{MAC\_\text{Delay}_{\text{basic}}, MAC\_\text{Delay}_{\text{RTS}}\}$$  \hfill (10)  

IV. Performance Evaluation

A. Simulation Results

This section presents the simulation results for the proposed EBE scheme. In order to evaluate the performance of the proposed scheme, we performed experiments on the basis of the NS-2 (Network Simulator) of Lawrence Berkeley National Laboratory (LBNL) [9]. We choose pathload and pathchirp for comparison. We compare their performance in terms of estimated bandwidth and measuring time. The simulation topology is shown in Fig. 3. In this simulation, the prober transmits the probing packets to mirror via router. The link between the R1 and R2 is 50Mbps. The link between the Prober and R1 is 100Mbps. The link between traffic sender and the R1 is set at 100Mbps. There are two cross traffic flows in the simulation. One is a constant bit rate (CBR) by UDP packets. The other is the http traffic transmitted by TCP packets.

$$MAC\_\text{Delay}_{\text{basic}} = DIFS + SIFS + MAC\_\text{ACK}$$  \hfill (8)  

Fig. 4, 5 and 6 show the comparison of the available bandwidth with different cross traffics. Pathload takes 6.4 seconds to measure the bandwidth. Pathload is not suitable to real-time application because it cannot reflect the real-time network status.
The available bandwidth estimated by pathChirp fluctuates between 0 and 61.92. It cannot be accurate information because of the hardness to provide the adequate service for client. On the other hand, the EBE shows the fast convergence time and low fluctuation of the estimated bandwidth. Table I shows the comparison of the available bandwidth and convergence time among the proposed EBE scheme, pathload, and pathchirp with various cross traffics.

<table>
<thead>
<tr>
<th>Available bandwidth (Mbps)</th>
<th>Convergence time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBE</td>
<td>54 31.66 51.67 0.05</td>
</tr>
<tr>
<td>pathload</td>
<td>52.38 49.41 50.03 6.4</td>
</tr>
<tr>
<td>pathchirp</td>
<td>61.92 56.90 56.76 0.05</td>
</tr>
</tbody>
</table>

**B. Empirical Results**

We have implemented the proposed scheme on the top of Ubuntu 10.04 to measure the available bandwidth. As shown in Fig. 7, the wireless testbed consists of a prober that performs the estimation, a traffic generator, a wireless AP, and two clients (Mirror, Traffic Receiver). The AP in the testbed is an ipTIME N604R with IEEE 802.11b/g/n mode. We switch off the automatic rate adaptation and clear to send (CTS) protection mode, and run networks on its maximum physical rates of 54Mbps. We benchmark EBE against the well-known testing scheme iperf, pathchirp, pathload, and wbest. Wbest is the only other real-time probing scheme we know that is applicable to wireless networks. It requires the wireless hop to be in the last link, because it needs to be sure that the probing packets arrive at the bottleneck with rate C. For the estimation of C it uses standard packet gap model and packet-pair dispersion. The testing schemes need to be installed on both the prober/receiver and mirror. We compare their performance with three metrics, namely, available bandwidth, convergence time, and intrusiveness measured by the Wireshark which is an open-source real-time packet analyzer widely accepted by both academic and industrial worlds. There are two cross traffic flows in the testbed. One is the CBR traffic and another is HTTP streaming traffic. For CBR traffic, we installed a traffic generator called Distributed Internet Traffic Generator (D-ITG) for the cross traffic in order to reproduce different network load conditions. D-ITG is a platform capable to produce traffic at packet level accurately replicating appropriate stochastic processes for both packet size random variables and inter departure time. For HTTP adaptive streaming traffic, we use the Apache web server and web based client.

Fig. 8, 9, and 10 show the comparison of the estimated bandwidth with various cross traffics. Iperf shows stable performance except CBR cross traffic environment. Also, the bandwidth measured by wbest has huge variance in every cross traffic which means that it cannot accurate information. Pathload significantly underestimated the bandwidth.
Available Bandwidth (Mbps)

Fig. 9. Comparison of the available bandwidth measured with different schemes and CBR traffic

Fig. 10. Comparison of the available bandwidth measured with different schemes and HTTP traffic

Fig. 11. Comparison of the convergence time

Fig. 12. Comparison of the intrusiveness

V. CONCLUSIONS

To determine the quality of HTTP adaptive streaming, we propose the EBE scheme for the bandwidth estimation. The EBE scheme calculates the link capacity in initial buffering time before start of playback. The EBE scheme models the probing rates efficiently for HTTP adaptive streaming. The proposed scheme adjusts the spread factor dynamically to reduce the overhead. Also, the EBE scheme considers the wireless delay such as MAC delay. Our simulation and evaluation results reveal that the proposed EBE scheme is an accurate, fast and non-intrusive bandwidth estimation scheme. For our future work, we are looking at integrating with a new quality adaptation scheme for QoS of HTTP adaptive streaming.

REFERENCES