A Novel Bandwidth Estimation Method Based on MACD for DASH

Van-Huy Vu, Ibrahim Mashal, Tein-Yaw Chung
Department of Computer Science and Engineering, Yuan Ze University
135 Yuan-Tung Road, Chung-Li, Taoyuan, Taiwan.
[huyvv, ibrahimmashal]@netlab.cse.yzu.edu.tw, csdchung@saturn.yzu.edu.tw
*Corresponding author: Tei-Yaw Chung

Received August 16, 2016; revised November 11, 2016; revised December 20, 2016; accepted January 22, 2017; published March 31, 2017

Abstract

Nowadays, Dynamic Adaptive Streaming over HTTP (DASH) has become very popular in streaming multimedia contents. In DASH, a client estimates current network bandwidth and then determines an appropriate video quality with bitrate matching the estimated bandwidth. Thus, estimating accurately the available bandwidth is a significant premise in the quality of video streaming, especially when network traffic fluctuates substantially. To cope with this challenge, researchers have presented various filters to estimate network bandwidth adaptively. However, experiment results show that current schemes either adapt slowly to network changes or adapt fast but are very sensitive to delay jitter and produce sharply changed estimation. This paper presents a novel bandwidth estimation scheme based on Moving Average Convergence Divergence (MACD). We applied an MACD indicator and its two thresholds to classifying network states into stable state and agile state, based on the network state different filters are applied to estimate network bandwidth. In the paper, we studied the performance of various MACD indicators and the threshold values on bandwidth estimation. Then we used a DASH proxy-based environment to compare the performance of the presented scheme with current well-known schemes. The simulation results illustrate that the MACD-based bandwidth estimation scheme performs superior to existing schemes both in the speed of adaptively to network changes and in stability in bandwidth estimation.

Keywords: DASH, Bandwidth Estimation, MACD indicator, Passive Technique, Video Streaming

https://doi.org/10.3837/tiis.2017.03.011 **ISSN : 1976-7277**

1. Introduction

The massive evolution of the Internet has led to the popularity of media streaming such as music, videos, and online streaming. Recently, these multimedia services have been extended to Internet on vehicles [1, 2], cloud-based video services [3] and even Fog computing based services [4]. Multimedia streaming is a very challenging service because it requires large bandwidth and real-time constraint. In the past, the Real-time Transport Protocol (RTP) has been used for video streaming [5]. RTP uses User Datagram Protocol (UDP) as the underlying transport protocol to reduce overhead or latency of retransmissions. Later on, the Hyper Text Transfer Protocol (HTTP) is used to deliver the video streaming [6]. HTTP uses Transmission Control Protocol (TCP) as the underlying transport protocol for HTTP. HTTP becomes a popular choice although it's reliable and downloading model seems not appropriate for video streaming. This is mainly because HTTP is simple and is widely support in any smart phones, notebooks and PCs. More importantly, it can easily reach all users bypassing any firewall protections or network address translation (NAT) gateways [7].

To exploit the advantages of HTTP while making it effective in video streaming, Dynamic Adaptive Streaming over HTTP (DASH) [8, 9] was proposed by MPEG and 3GPP to achieve this end. In DASH, a video is encoded into multiple versions at various bitrates, frame rates, resolutions, and audio quality. After the encoding, the video is divided into a series of small segments. During streaming, a client downloads segment by segment of a video from a DASH server to emulate the packet-based video streaming of the RTP scheme.

To make DASH adaptive to network states, the encoding information of a video is placed in a profile referred as Media Presentation Description (MPD), based on which a client can choose an appropriate quality of video segment for downloading with current network state. A client fully manages the streaming session through HTTP GET, i.e., it handles the initiation time of request of each segment, and switching video bitrates adaptively to match the available bandwidth. Thus, DASH can provide the best possible video quality to the clients. One critical work in DASH is to estimate current available bandwidth accurately so that we can choose a suitable bitrate matching the estimated bandwidth. Furthermore, due to the inherent bandwidth fluctuation, the client needs to estimate continuously the available bandwidth based on previous segment downloading condition to ensure good quality of video streaming [10]. The more precise the bandwidth is estimated, the more appropriate video bitrate and smooth in playback can a client determine. **Fig. 1** is the overview of DASH technique.

Estimating accurately the network bandwidth in real time is very challenging due to end-to-end delay variance and the dynamics nature of Internet. In the past, several schemes have been presented to estimate the available bandwidth. The traditional scheme, such as that for TCP RTT estimation, can give stable estimation but adapt slow to available bandwidth variation. The weakness of traditional schemes are mainly because they use fixed coefficients [11]. To cope with this problem, the authors in [12] proposed a novel filter with variable coefficients. The advantage of this approach is that it is sensitive to delay jitter. However, it becomes very unstable with the change of network. The unstable bandwidth estimation has significant impact on the services for the customers as claimed by [13]. This can lead to unnecessary oscillation of video quality services [14]. Thus these methods cannot detect abrupt change of network traffic. Therefore, a new bandwidth estimation scheme is needed for DASH services.

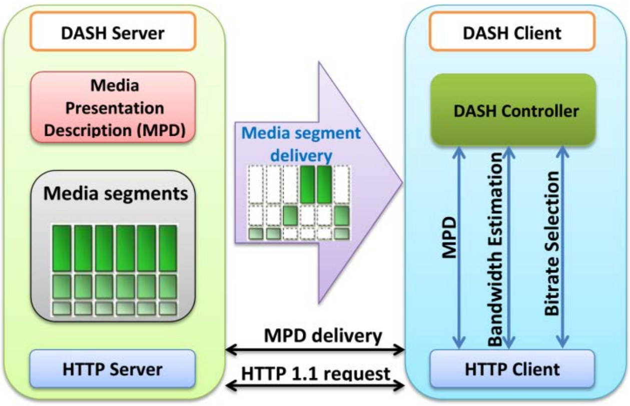


Fig. 1. The overview of Dynamic Adaptive Streaming over HTTP technique.

Based on previous discussion, a good filter must be stable, i.e. less sensitive to delay jitter, while adapting fast to real network traffic change. To meet this criteria, instead of presenting a new filter, this work presented a novel approach called Moving Average Convergence Divergence (MACD) -based bandwidth estimation scheme (MBES) for DASH services. In MBES, we first use an MACD indicator to detect the degree of traffic changes in the network, stable or agile, and then apply different filter accordingly to achieve stable while very agile bandwidth estimation in real time. MACD has been widely used for time series forecasting in the financial area such as stock or forex market to analyze or predict price movement [15]. An MACD indicator uses the series of history values to predict the trend of the observation movement. Therefore, an MACD indicator can be used to determine the network condition state whether the available bandwidth fluctuate slightly or substantially.

In the paper, we analyze the characteristics of MACD indicator and its use in network state detection. Then the filters used in each network state are presented. Finally, we used a DASH proxy-based environment to simulate the performance of MBES and compare that with existing popular bandwidth estimation schemes. The results show that MBES performs superior to existing schemes in both stability and agility of bandwidth estimation. The key contributions of this paper are summarized as follows.

- Use an MACD indicator to determine the trend of the available bandwidth changes, and classify the network condition into two states, a *stable state* and an *agile state*, by assigning two thresholds to the MACD indicator.
- Study and analyze the characteristics of MACD indicator and the impact of threshold values in classifying network states using simulation.
- Propose an innovative bandwidth estimation scheme based MACD. Design two filters to precisely estimate the available bandwidth to meet several requirements such as accuracy in both states, smooth in the stable state, sensitive in the agile state, and quick adaption to the real change of current bandwidth.
- Develop simulation using Omnet++ on a DASH proxy-based environment to study the performance of MBES and some chosen existing schemes.

The rest of paper is organized as follows. Section 2 introduces the related works. Section 3 introduces the application of MACD and its thresholds to estimate the available bandwidth in. In Section 4, we simulated our proposed mechanism to prove its effectiveness as well as compare and analyze our method with other methods. We concluded the paper in Section 5.

2. Related Work

The concept of bandwidth in networking refers to the amount of data or the speed of bit transmission that a link, a data channel, or a network path can transfer per unit of time. The maximum possible bandwidth that a link or an end-to-end path can deliver its data is known as the link capacity. The maximum bandwidth not used on a link or an end-to-end path is called available bandwidth, and usually, is measured in bits per second. The available bandwidth to the application such as multimedia streaming directly affects the performance of application [16]. The available bandwidth can vary over time, thus measuring it quickly plays an important role, especially in the case of multimedia streaming, to adapt the transmission rates.

In the past, many techniques for estimating available bandwidth have been presented and can be classified into four main categories [17]: active probing techniques (APT), passive technique (PT), technique only for wireless (TOWN), and other bandwidth estimation techniques (OBET). In APT, a client must actively generate traffic between itself and the server to estimate end-to-end bandwidth, while in PT, a client estimates bandwidth based on the data downloading time from the server. The TOWN schemes consider the data loss problem in bandwidth estimation. OBET cover the techniques other than the previous three categories. Since DASH clients must estimate network bandwidth continuously while they are downloading each segment, the passive technique suits the best for DASH. Thus here we only reviewed relevant techniques in the category of PT.

Several passive techniques on estimating bandwidth through multimedia segments' throughput are proposed using the running average of a connection's throughput. Sammar et al. [18] compare Netflix and Microsoft Smooth Streaming player based on traditional smoothed average method. The result shows that their proposed method works well under the persistently varying available bandwidth. However, the delay in bitrate switching may cause buffer underflow or sub-optimum Quality-of-Experience (QoE). Junchen et.al [19] has introduced Fair, Efficient, and Stable adapTIVE algorithm (FESTIVE) to estimate the next segment bandwidth by using harmonic mean [20] of last 20 segments' throughput. The harmonic mean technique shows its advantages in case of slight fluctuation of available bandwidth, but slow bandwidth adaptation becomes its drawback when bandwidth variation is significant. Similarly, the median of last several segments' throughput is used in estimating next bandwidth [21]. The proposed method is accurate but slow to adapt to persistent bandwidth variations. Wei et al. [12] has introduced User Adaptive Video-enabled DASH (UDASH) to estimate bandwidth based on each segment's throughput. The algorithm, however, fails to estimate the available bandwidth well.

All aforementioned methods are either stable but slow to adapt to available bandwidth variation or adapt fast with variable coefficient in the estimation filter but become unstable when estimated throughput has some noise. Thus, a better bandwidth estimation scheme is needed for various network states.

3. Bandwidth Estimation Based on MACD

In the DASH technique, a client dynamically selects an appropriate bitrate of media segments in the MPD file based on the network condition. To do this, a client must continuously estimate current bandwidth so that it can choose a suitable bitrate. In this section, we proposed a method based on the MACD indicator and its thresholds to determine the degree of change in the available bandwidth. In order to smooth the varying of estimated bandwidth and avoid the transient change of available bandwidth leading to unnecessary oscillation of estimated bandwidth [22], we use a harmonic mean combination with MACD thresholds in the specific stable state to deal with this purpose.

We present our proposed network bandwidth estimation in three parts. The first is the application of MACD indicator. The second is determining the stable state and agile state of available bandwidth variation based on the MACD and its thresholds. We also studied the sensitivity of MACD with various threshold levels. Lastly is applying an appropriate bandwidth estimation scheme in the stable state and agile state, respectively, to cope with various degrees of bandwidth change.

3.1 MACD indicator

Moving average convergence/divergence (MACD), was invented in the late 1970s by Geral Appel [15], is one of the most common indicators applied in stock or forex market to analyze or predict price movement. The MACD value (for short, we use the MACD) is the subtraction of the two *EMAs* (exponential moving average), one is known as the short period, and the other as the long period. MACD can be calculated as follows [23].

$$MACD = EMA_{short} - EMA_{long} \tag{1}$$

$$EMA_x = \frac{p_1 + (1-\alpha)p_2 + (1-\alpha)p_3 + (1-\alpha)p_4 + ...}{1 + (1-\alpha) + (1-\alpha)^2 + (1-\alpha)^3 + ...} \tag{2}$$

$$\alpha = \frac{2}{N+1} \tag{3}$$

where *x* represents for the short or long, p_i (i = 1, 2, ..., N) represents a sequence of observations. *N* is the number of observations to calculate EMA_{short} or EMA_{long}. For instance, we can choose a pair of samples is (12, 26) to calculate two EMAs, that mean, we use $N = 12$ samples to calculate EMA_{short} and $N = 26$ samples to calculate EMA_{long}.

The MACD oscillates above or below the zero line (the centerline). The positive MACD indicates that the short period EMA or faster-moving average is above the long period EMA or slower moving average and vice versa the negative MACD indicates that the short period EMA is below the long period EMA. The MACD does not have an upper or a lower limit.

The MACD indicator can tell us about its story through the trend, the sensitivity and the strength of this signal.

- Firstly, if MACD is above the zero line and rising, then the gap between the short period EMA and the long period EMA is expanding. That means the faster moving average is above the slower moving average. Upward momentum is increasing, and this would mean that the observation value is increasing. If MACD is declining and goes below the zero line, then the negative gap between the faster-moving average and the slower moving average is widening. Downward momentum is taking place, and the observation value tends to be decreasing.
- Secondly, to make MACD more sensitive to the change of data series, the coefficient α for a short period EMA is chosen large and *α* for a long period EMA will be small.

The adjusting result of α is obtained by Equation (3) via changing number of time series data (N). The smaller value of N, the greater α we will have $(0 < \alpha \le 1)$. If EMA_{short} has a large value of α, it indicates that MACD responds faster to small variations of observation samples. In other words, the less number of time series data in usage to calculate EMA_{short}, the more sensitive of MACD becomes. In contrast, the more number of time series data used in the short period EMA, the slower react to the change of data observation.

- Lastly, the distance of MACD to zero line, or in other words, the gap between the faster-moving average and the slower moving average, indicates the strength of MACD. If MACD goes further away from the zero line, and MACD is declining, the observation data changes significantly and becomes smaller. Vice versa, the observation data also changes largely, but MACD is in an uptrend, the observation data is increasing.

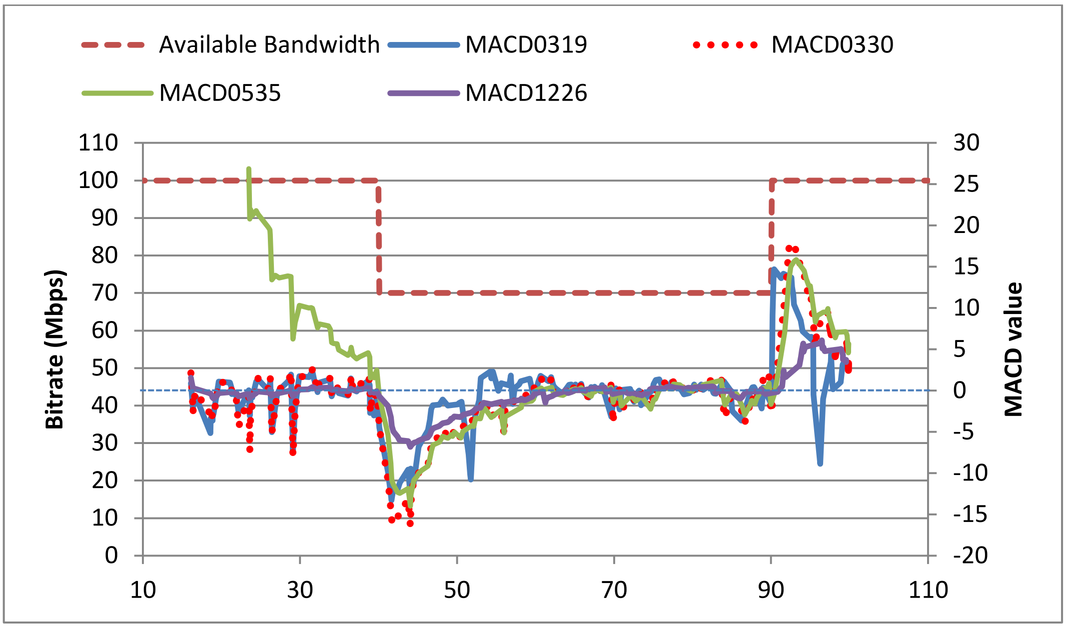


Fig. 2. MACD indicator shapes with various pairs of EMAs

Let N_f and N_s be the number of bandwidth samples to calculate *faster EMA* and *slower EMA* respectively. **Fig. 2** presents the difference when the different number of bandwidth samples is used to calculate the MACD. We observe four pairs of MACD indicators (03, 19; 03, 30; 05, 35; 12, 26). For instance, ***MACD0319*** means N_f = *03* and N_s = *19*. The most common pair of (N_f, N_s) to calculate *short EMA* and *long EMA* in the stock market is (12, 26). In this example, the maximal link bandwidth is 100 Mbps and background traffic of 30Mbps is injected to the link at the 40th second and finishes at the 90th second. Based on **Fig. 2**, we have the following observations:

- When the network bandwidth becomes bottleneck starting from the 40th second, the MACD goes down below the zero line. At the 90th second, the network bandwidth is back to 100 Mbps; the MACD indicator moves upward away from the zero line.

- If N_f is small, then MACD reacts rapidly to the change of bandwidths. As we can see in **Fig. 2**, when $N_f = 3$, ***MACD0319*** and ***MACD0330*** change faster comparing with that of ***MACD0535*** and ***MACD1226***. On the other hand, MACD becomes noisier when N_s is smaller and is more stable if N_s is larger. This can be seen by comparing ***MACD0330*** with $N_s = 30$ and ***MACD0319*** with $N_s = 19$. With the same $N_f = 03$, ***MACD0330*** is more stable and less noise as comparing with ***MACD0319***.
- The strength, or the gap between *faster EMA* and *slower EMA*, of MACD indicates the sensitivity of the bandwidth changing. If the gap is widening largely, the MACD is more sensitive to the bandwidth changing. It is obvious that ***MACD0330*** is more sensitive than ***MACD1226***.

3.2 MACD thresholds

The behavior of MACD indicates the convergence or divergence state of the faster EMA and the slower EMA. If MACD moves around and closer to the zero line, it implies that the faster EMA and slower EMA are converging. In other words, the MACD is in a stable state. In contrast, if the MACD goes further away from the zero line, this means the faster EMA and the slower EMA are diverging. Thus, the MACD is in an agile state. To reduce the impact of small fluctuation of MACD, or noise, in determining when the MACD is in a stable state or agile state, we assigned two thresholds Th_N *and* Th_P for the positive and negative of MACD correspondently. The noise appears during the estimation bandwidth because of jitter, the difference of RTT, in segment downloading. Jitter makes throughput calculation fluctuate, and is unavoidable because of the queuing scheduling in routing nodes and the TCP slow-start as claimed by the author in [24]. Thus, by using thresholds for MACD indicators, we can reduce the effect of noise in bandwidth estimation.

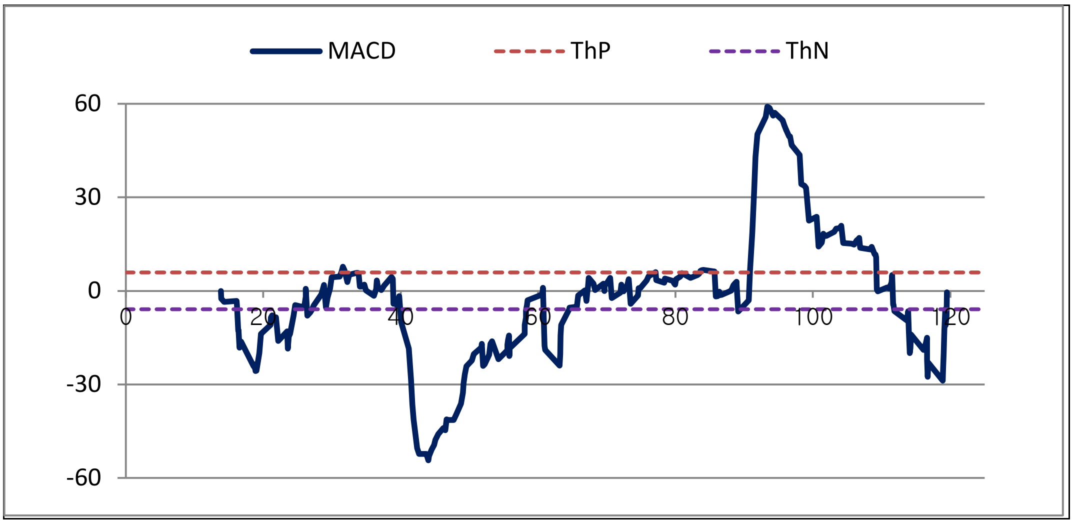


Fig. 3. MACD and two thresholds

The thresholds are symmetric around the zero-line of MACD as shown in **Fig. 3**. If the MACD is between the positive threshold and the negative threshold, it is said that current bandwidth is in a *stable state*, or the available bandwidth changing is considered small. Otherwise, the MACD is in an *agile state*, and the available bandwidth is assumed to be fluctuating significantly.

To estimate accurately also adapt fast to the available bandwidth, the different filters should be used in the stable and the agile state. In the stable state, a new bandwidth sample should be weighed close to those previous samples so that the estimated bandwidth will not fluctuate substantially even when a new sampled bandwidth value deviates from the average value of the bottleneck bandwidth largely. On the contrary, in the agile state, a new bandwidth sample should be weighted much more than previous samples so that the estimated bandwidth can adapt fast to the large change of the bottleneck bandwidth. Based on the above observation, we summarize our MACD-based design for bandwidth estimation as follows.

- If $MACD \in (Th_N, Th_P)$, the network condition is in a *stable state*. The *harmonic mean* technique is used in this state to ensure slight fluctuation of network bandwidth estimated.
- Otherwise, $MACD \leq Th_N$ or $MACD \geq Th_P$ means that network condition is in the *agile state*. For quick adaptation to the network change, we use an exponentially weighted moving average filter [12, 23, 25] which is widely applied to estimate the available bandwidth.

To make our bandwidth estimation method stable but adapt fast to network changes, we use two filters with variable coefficients for the stable and the agile state respective. The merit of using variable coefficients in the low pass filters is in its adaptation to network conditions. In our design, we make the coefficient of the low-pass filter for the stable state less sensitive, while making the coefficient more sensitive when the network is in the agile state.

3.3 MACD-based Bandwidth Estimation Scheme

This section presents the novel MACD-based bandwidth estimation scheme (MBES). As in the previous section we know that, when MACD moves within the stable area, any drastic change of bandwidth in this state is considered noise in calculation or not real bandwidth change. Therefore, a method for estimating bandwidth to make the result not only stable but also smooth need to be addressed.

In MBES, instead of using the average throughput as the conventional approaches, the harmonic mean technique [20] is used to smooth the slight fluctuation of estimating available bandwidth in the stable state area. The benefit of using harmonic mean is that it is more robust to larger outliers and more suitable when we want to calculate the average rate as claimed in [19]. The equation to calculate the harmonic mean is given by Eq. (4).

$$T_e^H(i) = \frac{n}{\sum_{i=1}^{n} \frac{1}{T(i)}} \tag{4}$$

where n is the number of throughput samples, $T(i)$ is the throughput i^{th}. In this paper, we use the number of throughput samples $n = 20$.

Equation 5 summarizes the filters used in MBES for both *stable stage* and *agile state*.

$$T_e(i) = \begin{cases} \delta_1 T_e^H(i-1) + (1-\delta_1)T_s(i-1), \text{ if MACD } \in (\text{Th}_N, \text{Th}_P) & \text{(5a)} \\ \delta_2 T_e(i-1) + (1-\delta_2)T_s(i-1), \text{ if MACD } \geq \text{Th}_P \text{ or MACD } \leq \text{Th}_N & \text{(5b)} \\ T_s(i-1), \qquad \text{if (i = 1)} & \text{(5c)} \end{cases}$$

where $T_e(i)$ is the estimated bandwidth for multimedia segment i^{th}, $T_e^H(i-1)$ is the *harmonic mean* of the last n throughput samples which is calculated in Eq. (4), $T_e(i\text{-}1)$ is the estimated bandwidth for the last segment, $T_s(i\text{-}1)$ is the measured throughput in downloading the last multimedia segment and is calculated by Eq. (10), δ_1 and δ_2 are the variable weights which are computed as below.

$$\delta_1 = \frac{1}{1+e^{-k*(\rho-P_0)}} \tag{6}$$

$$\delta_2 = \frac{1}{1+e^{k*\Delta}} \tag{7}$$

where

$$\rho = \frac{|T_s(i-1) - T_e(i-1)|}{T_e(i-1)} \tag{8}$$

$$\Delta = \frac{T_s(i-1) - \overline{T_s}(i-1)}{\overline{T_s}(i-1)} \tag{9}$$

k and P_0 are parameters of the logistic function as in [25]. $\overline{T_s}$ is the average of last m throughput samples (we use $m = 7$ in our system). The throughput of channel v can be estimated as below.

$$T_s^v = \frac{r^v(i) * \tau}{T_d(i)} \tag{10}$$

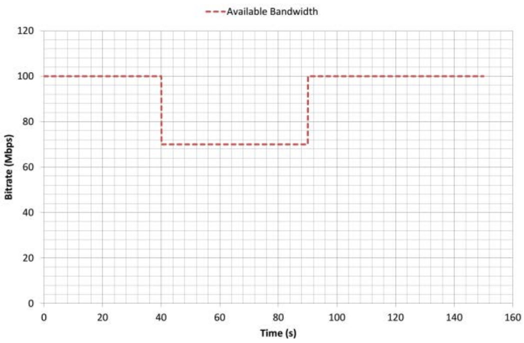
where $r^v(i)$ is the bitrate of channel v for multimedia segment i^{th}, τ is the playback duration and $T_d(i)$ is the time consumed to download segment i^{th}.

When MACD is in the *stable state*, MBES tries to reduce the drastic change of bandwidth estimation but allow fast adaptation to small change. When throughput samples have close values, the value of $T_s(i\text{-}1)$ and $T_e(i\text{-}1)$ are closer. Therefore, ρ in Eq. (8) tends to zero, which makes the smooth factor δ_1 in Eq. (6) becomes small. Thus, bandwidth result now trusts more on the instant estimation throughput T_s. Hence, according to Eq. (5a) our result is more accurate and more sensitive to bandwidth change within the *stable state*. In contrast, if throughput samples T_s varies strong, the difference between each sample and T_e gets large. Then ρ in Eq. (8) increases, which causes δ_1 to increase. Consequently, bandwidth calculation is mainly based on the harmonic mean, which smooths the variation of bandwidth change.

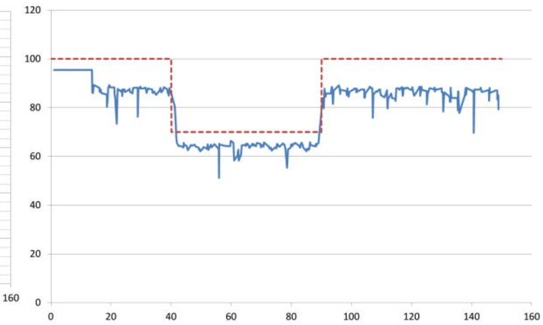
When MACD is in the *agile state*, MBES uses a filter to satisfy two purposes, fast adaptation to the drastic change of bandwidth and converge quickly to a stable value. To achieve this, we make the coefficient δ_2 change quick. Instead of using harmonic mean, we use a smaller number of throughput samples in Eq. (9) to calculate Δ. From Eq. (7), it can be readily seen that δ_2 gets small quickly when throughput values drastically change. Consequently, Eq. (5b) almost directly uses the previous segment to estimate the bandwidth, achieving fast adaptation to bandwidth change.

Fig. 4 shows the influence of two thresholds Th_N and Th_P to the result of bandwidth estimation. We observed different scenarios with multiple levels of threshold Th_N and Th_P with respect to 0.1%, 0.5%, 3%, 5.5% and 8% time of initial bandwidth (here, Th_N and Th_P are symmetrical over zero line) as follows.

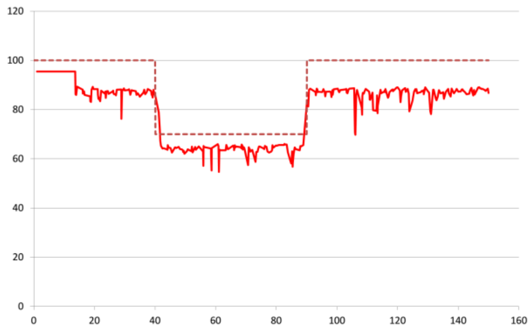
- When MACD thresholds have small values, in other words, the distance between Th_N and Th_P is small, or the stable state area for MACD is set to be narrow, the bandwidth estimation is more sensitive to the fluctuation of bandwidth. It also adapts faster to the bandwidth change comparing with broader stable state areas as we can see in **Fig. 4(b)**, **(c)** and **(d)**. However, the disadvantage of a narrow stable state area is that the bandwidth noise can easily trick the MACD indicator and makes the MACD indicator move quickly out of the stable state area. Consequently, the estimated bandwidth includes unwanted noise.
- When MACD thresholds have large values, the stable state area is wide. The MACD indicator will exceed the stable zone if the network bandwidth has large change. Consequently, our bandwidth estimation is slow in reacting to bandwidth changes, or our system is less sensitive to bandwidth variation as we can see in **Fig. 4(e)** or **(f)**. Widening stable state area makes MACD remain in the stable state most of the time and resulting in small fluctuation of estimated bandwidth. However, the drawback of widening stable state area is to make bandwidth estimation slow to adapt bandwidth changes.
- The use of threshold values has great impact on how fast we can detect major network bandwidth change and how vulnerable is the MACD to the delay variance. Thus, there is tradeoff in choosing the value of MACD thresholds. The EMA_{long} can better smooth out the impact of delay variance than that of EMA_{short}, and the effect becomes more obvious when the difference between N_f and N_s becomes larger. As a result, when the difference between N_f and N_s becomes larger, the MACD indicator becomes more sensitive to the delay variance. Thus, when the difference between N_f and N_s is larger, the threshold values must be larger to reduce the probability of incorrect judgement of network state due to delay variance, which means the value of N_f and N_s has impact on the threshold value setting. However, how to select an appropriate threshold for different MACD indicators is not easy because it is highly related to the traffic pattern of the network traffic.



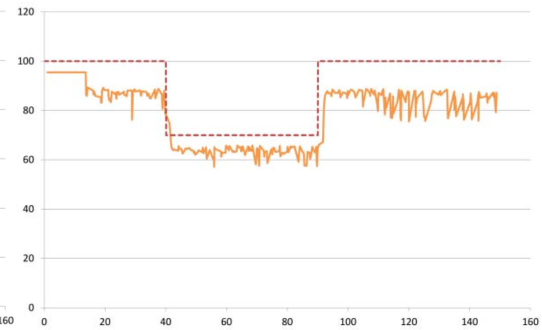
(a). Available bandwidth shaping



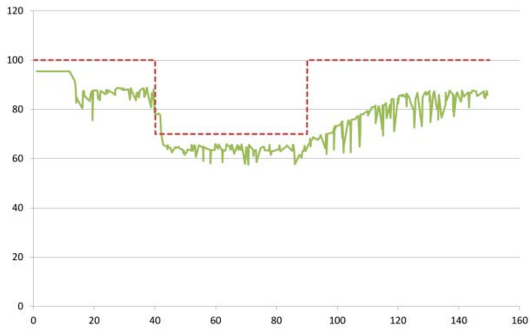
(b). Two thresholds level are 0.1% BW*



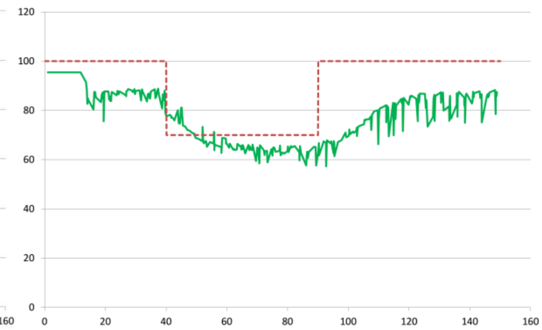
(c). Two thresholds level are 0.5% BW*



(d). Two thresholds level are 3% BW*



(e). Two thresholds level are 5.5% BW*



(f). Two thresholds level are 8% BW*

Fig. 4. Bandwidth estimated by MBES in various threshold levels (BW* is initial bandwidth)

4. Performance Evaluation

This section introduced the simulation environment and how video bit rate is chosen to best utilize current available bandwidth. Then, we analyzed the simulation results and compared the performance of MBES and current well-known schemes.

4.1 Simulation Environment and Configuration

To study the performance of MBES, we consider a proxy-based DASH service in which a multimedia proxy is located between clients and servers. Multiple users who are interested in some selected popular multimedia channels such as Youtube or a football match channel share the same proxy. That means the proxy manages several channels, and each channel has a number of users. Thus, our multimedia proxy serves as a powerful DASH client agent by which all requests from the clients to the server are handled. Here, we assume that, the data link among clients and proxy is LANs' connection. Thus we do not have any congestion between them. In contrast, at the server side, the network bottleneck can happen between the proxy and the server. Therefore, applying MBES in the proxy to estimate the available bandwidth between the proxy and the server is easier to study its performance.

The simulation environment and configuration is shown in **Fig. 5** where BWE is the bandwidth estimation process using the MACD indicator and BRS is the bitrate selection process using a genetic algorithm, to be described in the next section. In our scenario, the system processes can be briefly described as follows:

- The clients send requests to the server through a proxy.
- If it is cache hit, the proxy responses data to the clients. If cache misses, the proxy ensures that the buffer is not overflow before doing the next task; otherwise, the system forces the request of that channel waits for a segment playback period.
- The BWE estimates the available bandwidth between the server and proxy based on MACD indicator.
- BRS uses GA to determine a new media bitrate for each channel after receiving the available bandwidth from BWE.
- The proxy forwards client's request to the server with the optimum bitrate chosen by GA or rewrites client request if the initial bitrate requested by client is different from the bitrate chosen by GA.
-

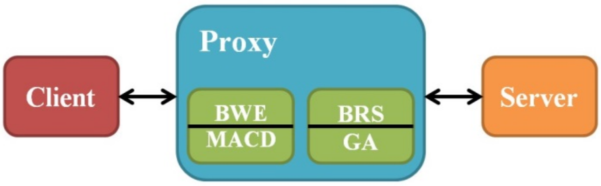


Fig. 5. Simulation model

We evaluated our proposed approach over a network simulation tool – *Omnet++ 4.6* and *Inet Framework 3.2* [26, 27]. **Fig. 6** is our simulation environment which consists of five components: a server, three clients as a representative of three user groups, two routers, a proxy and a pair of hosts for generating background traffic.

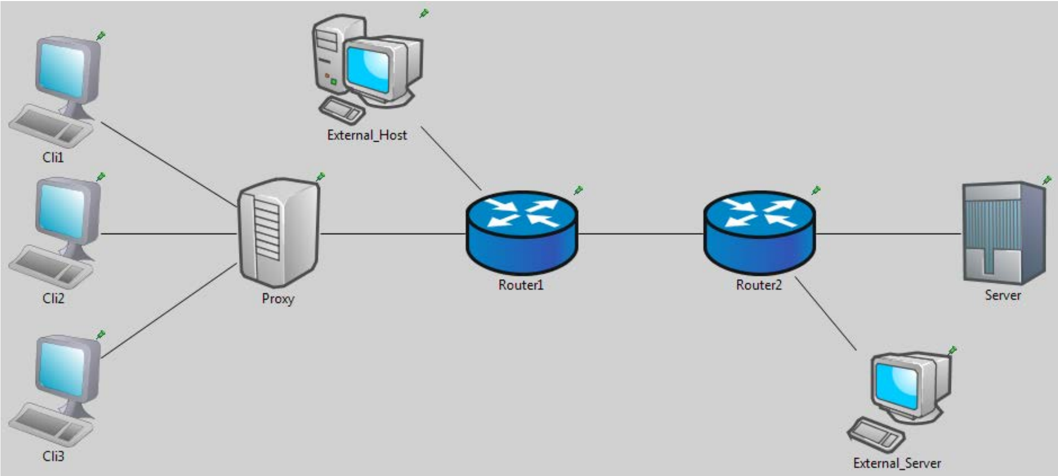


Fig. 6. Network simulation model

- The shared link is 100 Mbps. The TCP protocol in our model is *TCPReno*; *MSS* is 1452byte; bit error rate is 0; link delay is 0.1us. *External_server* transmits data to *External_host* using UDP to generate background traffic in different scenarios.
- *Cli1*, *Cli2*, and *Cli3* represent three user groups that are watching three different channels, each of which consists of 10, 6, 18 users respectively.
- The list of bitrate includes 9 levels as shown in **Table 1**.
- For the genetic algorithm, we set the population size as 50, the number of generation is 50, the number of iteration is 50, the constraint for bitrate switching levels is less than or equal to 2.
- The buffer size for each channel is 30s. The video segment duration is 2s.
- A pair of EMAs, 3 and 30 samples are used to calculate MACD (***MACD0330***).

In the initial simulation, we used a constant bandwidth to fill up the buffer and took enough samples for calculating MACD value. Based on our observation after many trials, the MACD threshold value is set from *0.1%* to *0.5%* of our initial bandwidth (*BW**). In particular, we set the threshold is *0.5%BW** in all of our simulation. This means that the stable state is when MACD falls in the interval (Th_N, Th_P), equivalent to the range (-*0.5%BW**, *0.5%BW**). Outside of this interval is the agile state.

4.2 Bitrate Selection using Genetic Algorithm

To best use the available bandwidth between the proxy and the server, this study developed a genetic algorithm to choose bitrate for each channel dynamically. In DASH, a client can dynamically select an appropriate bitrate based on its network condition from the list of media bitrates provided by the MPD file. If the instant video bitrate selected by the client is much less than the available bandwidth, the service quality is low while the network bandwidth is ineffective in usage. On the other hand, if the instant video bitrate is greater than the available bandwidth, the destination buffer may become underflow resulting in frequent interruption during video playback. Thus, choosing an appropriate bitrate for a video is important in DASH.

When some competing clients share the same link, selecting media bitrate plays an important role not only to guarantee the service quality but also to optimize the bandwidth utilization. To achieve optimal bitrate selection and user perceive quality, this work uses a Genetic Algorithm (GA) [28] to solve the problem.

The GA is designed to determine the bitrate in each channel under the following rules. The input of the GA consists of the total estimated bandwidth *BW*, the number of channels *k*, the number of clients per channel u_i^k, a list of bitrates $\{R^k = r_1^k, r_2^k, \ldots, r_n^k\}$ in each channel provided by the MPD files. **Table 1** is the list of bitrate levels that we are using in our system. The output of the GA is a list of bitrates for each channel that has been optimized.

Table 1. List of bitrate levels

Level	L1	L2	L3	L4	L5	L6	L7	L8	L9
Bitrate (Kbps)	250	350	450	500	800	1000	1500	2000	2500

The constraints for the GA include:

- The total bitrate of all channels cannot exceed the bandwidth estimated.
- Switching media bitrate directly influences the user-perception quality. To reduce the impact of user-perceived quality when switching bitrate [29], the number of bitrate switching level change cannot be larger than l ($l \leq 2$).

Within the scope of this article, we mainly focused on bandwidth estimation technique, so we will not explain thoroughly about how GA chooses the bitrate.

4.3 Results and Discussion

In this section, we implemented several bandwidth estimation mechanisms to compare with our proposed method. We compare the conventional approach (CVA), FESTIVE [19], UDASH [12], Harmonic mean with the conventional approach (HMCA) and our approach (MBES).

In the CVA, the network bandwidth is estimated by Eq. (10), and coefficient δ = 0.8.

$$T_e(i) = \delta T_e(i-1) + (1-\delta)T_s \tag{10}$$

The HMCA, which using 20 samples in calculating Harmonic mean, is calculated as in Eq. (11) and coefficient δ = 0.8.

$$T_e(i) = \delta T_e^H(i-1) + (1-\delta)T_s \tag{11}$$

Scenario 1: In our simulation, the *External_Server* stream video data to *External_Host* using UDP to generate backbone traffic as shown in **Fig. 7a**.

- The maximal bandwidth is 100 Mbps
- The available bandwidth becomes 80 Mbps between the 40th second to the 90th second.
- The available bandwidth becomes 60 Mbps between the 150th second to the 200th second.

- Total simulation time is 230s.

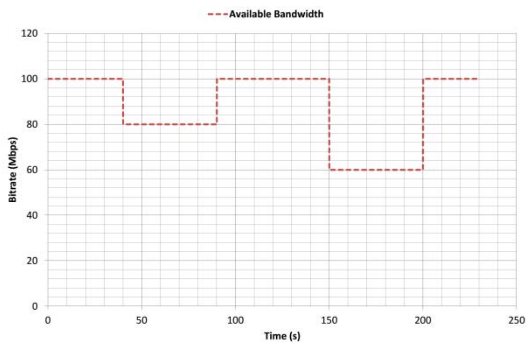
With the same configuration, the bandwidth estimation result achieved as we can see in **Fig. 7**. The network bandwidth change is as shown in **Fig. 7a**. The estimated bandwidth based on the CVA is as shown in **Fig. 7b**. The benefit of this technique is that it is simple, but its drawback is that it uses constant coefficient and hence cannot adapt quickly to network bandwidth fluctuation. In particular, it cannot handle noise in segment throughput calculation. Thus, the network bandwidth obtained by this approach has many variations.

Fig. 7c shows the performance using the harmonic mean approach like that in FESTIVE with 20 samples. Because of multiple channels sharing the same data link, the total estimated bandwidth using the harmonic mean has two issues. The first problem is that the harmonic mean slowly responds to the drastic changes because it equally treats some history samples (20 samples in this case) to estimate the next network bandwidth. Secondly, this approach does not separate the real change from the short time fluctuation of bandwidth. In couple with multiple channels sharing the same link, transient changes could mislead the harmonic mean approach to deviate from the actual network state and get a less accurate estimation in comparison with our results.

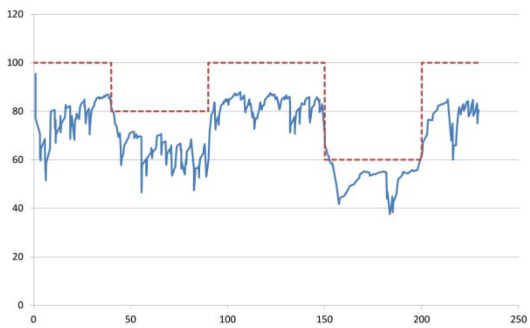
What the result shows in **Fig. 7d** is based on the UDASH approach. The advantage of UDASH is its fast adaptation in the real change of bandwidth, in particular when the cases like that at the 40th second, the 90th second, the 150th second or the 200th second. In our approach, we use the same method to calculate weight in order to control the smooth factor in estimating the next segment bandwidth. Thus, the weight changes adaptively with respective to the network condition making our scheme adapt as fast as UDASH. However, the disadvantage of UDASH is that it tends to generate largely fluctuated bandwidth estimation, especially when several channels sharing the same data link. Therefore, the results show that our proposed approach can achieve smoother and more accurate bandwidth than that of UDASH.

Fig. 7e shows the bandwidth estimated by the HMCA using Eq. (11). By using the last 20 downloaded segments to calculate the harmonic mean, the harmonic mean in this case just makes this method less "noise" or less bandwidth fluctuation as compared with UDASH or original conventional approach, but it is slow to respond bandwidth change.

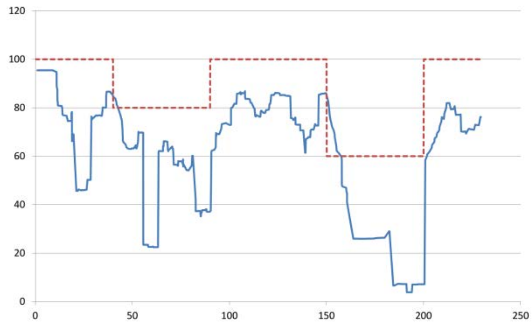
Fig. 7f demonstrates that by applying MACD and its two thresholds, our proposed method works well in estimating bandwidth. It quickly adapts to bandwidth change and eliminates most of the bandwidth fluctuation in estimating network bandwidth when multiple clients share the same network bottleneck.



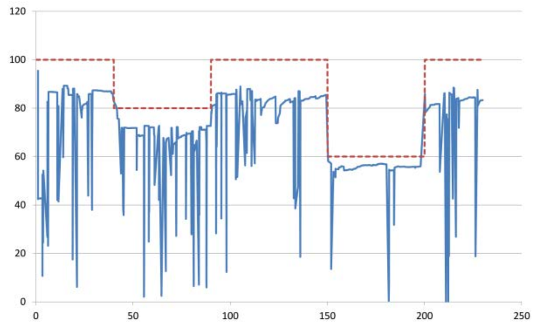
(a). Available bandwidth



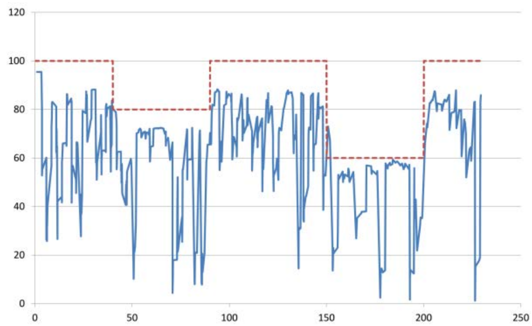
(b). Conventional approach (CVA)



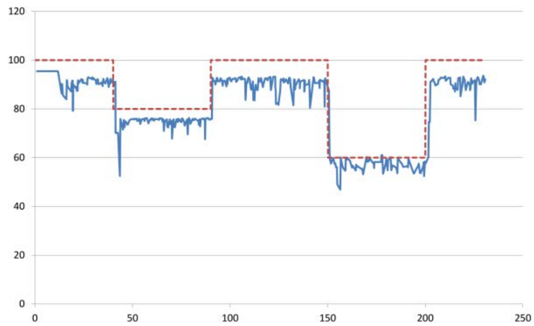
(c). FESTIVE



(d). UDASH



(e). HMCA



(f). MBES

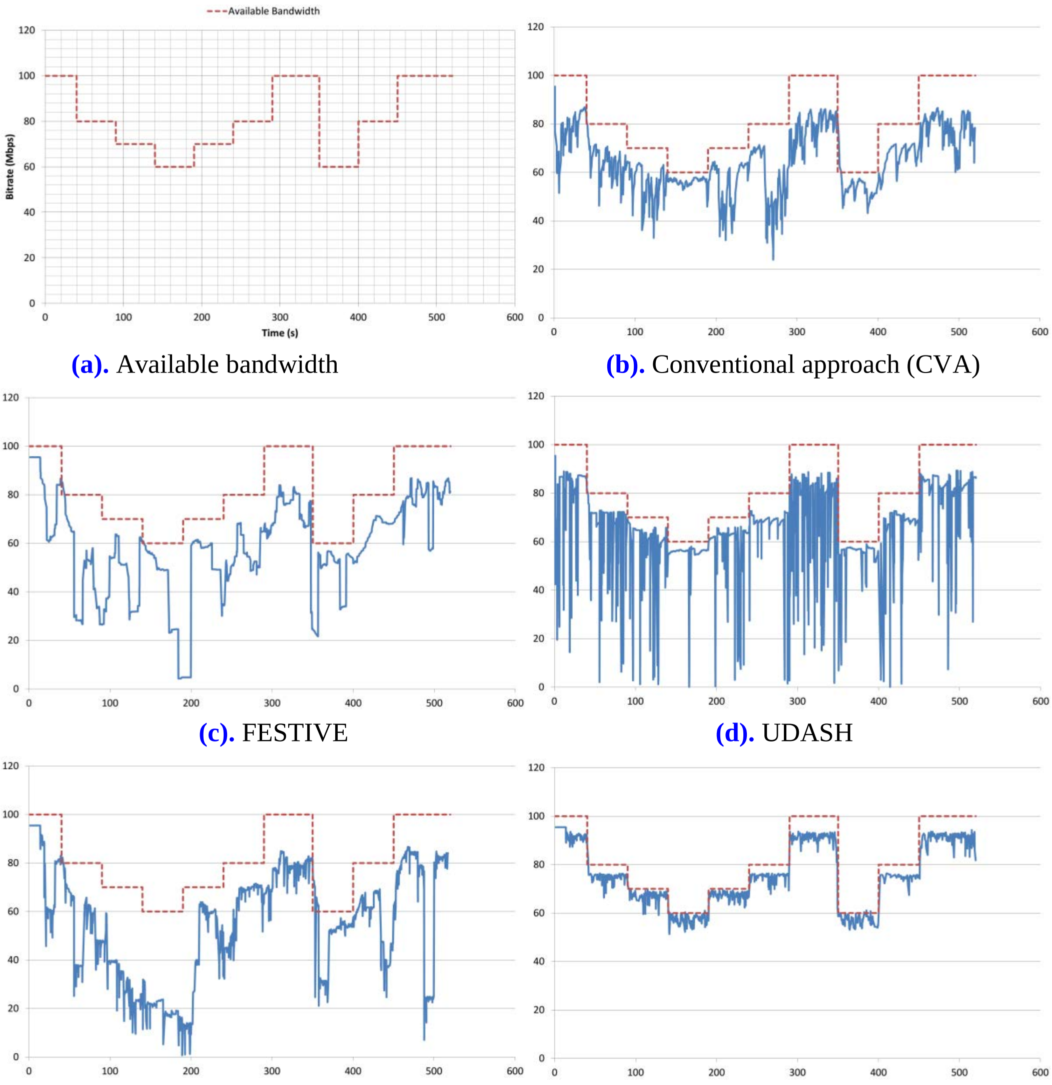
Fig. 7. Comparison of bandwidth estimation with various approaches in scenario 1

Scenario 2: The available network bandwidth modulated by the background traffic generated between *External_Server* and the *External_Host* as shown in **Table 2** and the shape looks like in **Fig. 8a**. We also compare our proposed method with four other mechanisms as those in scenario 1. Total simulation time is 510s.

Table 2. The available bandwidth of scenario 2.

No	Time start (s)	Time end (s)	Available bandwidth (Mbps)
1	0^{th}	40^{th}	100
2	40^{th}	90^{th}	80
3	90^{th}	140^{th}	70
4	140^{th}	190^{th}	60
5	190^{th}	240^{th}	70
6	240^{th}	290^{th}	80
7	290^{th}	350^{th}	100
8	350^{th}	400^{th}	60
9	400^{th}	450^{th}	80
10	450^{th}	510^{th}	100

By observing various bandwidth changes, the result obtained is quite similar to the result as describing in scenario 1. The *CVA* is slow to respond to bandwidth changes and has more fluctuation in bandwidth estimation as we can see in **Fig. 8b**. The *FESTIVE* approach and *HMCA* are much slower than other approaches in adaptation to bandwidth changes as shown in **Fig. 8c** and **Fig. 8e** respectively. *UDASH* as shown in **Fig. 8d** can adapt fast to bandwidth variation but is too sensitive to noise in throughput noise, which causes it confused between real bandwidth changes and transient variation of bandwidth. Therefore, the bandwidth estimated by *UDASH* has many sharp variations, especially in the case when multiple users are competing for a bottleneck bandwidth. **Fig. 8f** illustrates the result of our proposed method. Apparently, our proposed method performs well and gets more accurately as compared with all other methods. It is sensitive to bandwidth changes while eliminating most of the oscillation in estimating bandwidth which affects user quality of experience significantly [14]



(a). Available bandwidth **(b).** Conventional approach (CVA)

(c). FESTIVE **(d).** UDASH

(e). HMCA **(f).** MBES

Fig. 8. Comparison of bandwidth estimation with various approaches in scenario 2

To better compare the performance among schemes, we use 95% confidence interval - CI(95%) [30] in analyzing the simulation results. Table 3 summarizes the analysis of the two scenarios above, where the number of samples (DIFF) is the difference between the estimated bandwidth and the available bandwidth. In other words, DIFF is the estimated error of each bandwidth estimation schemes. The value of CI(95%) illustrates the variation of the estimated bandwidth from the mean value in each scheme. The results show that in both scenarios, the means of DIFF or the bandwidth error of MBES are much smaller than those of other approaches. Therefore, MBES can obtain more accurate bandwidth estimation than other schemes. Moreover, in the both scenarios, the CI(95%) of MBES are the smallest, which means MBES is more stable than other compared methods.

Table 3. The 95% confidence interval analysis of two scenarios

	No	Schemes	Number of samples	Mean of DIFF	Standard Deviation of DIFF	Confidence Interval (95%)
Scenario 1	1	CVA	437	16.97	7.02	0.66
	2	FESTIVE	437	26.95	16.75	1.57
	3	UDASH	437	17.64	16.77	1.57
	4	HMCA	437	26.38	17.65	1.66
	5	MBES	437	7.14	5.40	0.51
Scenario 2	1	CVA	1062	16.67	9.66	0.58
	2	FESTIVE	1062	24.09	14.45	0.87
	3	UDASH	1062	16.19	17.06	1.03
	4	HMCA	1062	25.16	14.33	0.86
	5	MBES	1062	5.37	3.44	0.21

5. Conclusion

In this paper, we presented a novel MACD-based bandwidth estimation scheme (MBES). In MBES, an MACD indicator with two thresholds is used to detect the network state, either stable or agile, and then the respective filter is used to estimate the available bandwidth. In the *stable state*, the filter uses appropriate variable coefficient which makes it immunes better to delay variance in segment downloading, while in the *agile state*, other variable coefficients based on delay variance are used to make the filter adapt fast to the network changes. Simulation results illustrated that MBES can effectively estimate the bandwidth for DASH in various network states. Besides, the results also depicted that the bandwidth estimation by MBES is smooth in the stable state and adapts fast in the agile state. Moreover, the bandwidth achieved by our method is much closer to the actual available bandwidth than that of the CVA, UDASH, FESTIVE, and HMCA.

In the future, we plan to study our proposed method by using multiple levels of thresholds for the MACD indicator in estimating bandwidth. We expect that multiple level thresholds can make the system more robust when the environment has more frequent bandwidth changes. Another goal is to apply our design for the wireless mobile environment where link quality changes much faster when clients move fast.

References

[1] O. Kaiwartya, A. H. Abdullah, Y. Cao, A. Altameem, M. Prasad, C. T. Lin, et al, "Internet of Vehicles: Motivation, Layered Architecture, Network Model, Challenges, and Future Aspects," *IEEE Access*, vol. 4, pp. 5356-5373, 2016. Article (CrossRef Link)

[2] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-efficient Adaptive Resource Management for Real-time Vehicular Cloud Services," *IEEE Transactions on Cloud Computing*, vol. PP, pp. 1-1, 2016. Article (CrossRef Link)

[3] A. Soltanian, M. A. Salahuddin, H. Elbiaze, and R. Glitho, "A resource allocation mechanism for video mixing as a cloud computing service in multimedia conferencing applications," in *Proc. of 2015 11th International Conference on Network and Service Management (CNSM)*, pp. 43-49, 2015. Article (CrossRef Link)

[4] P. G. V. Naranjo, M. Shojafar, H. Mostafaei, Z. Pooranian, and E. Baccarelli, "P-SEP: a prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks," *The Journal of Supercomputing*, pp. 1-23, 2016. Article (CrossRef Link)

[5] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," 2003. Article (CrossRef Link)

[6] A. Begen, T. Akgul, and M. Baugher, "Watching video over the web: Part 1: Streaming protocols," *IEEE Internet Computing*, vol. 15, pp. 54-63, 2011. Article (CrossRef Link)

[7] M. Michalos, S. Kessanidis, and S. Nalmpantis, "Dynamic adaptive streaming over HTTP," *Journal of Engineering Science and Technology Review*, vol. 5, pp. 30-34, 2012. Article (CrossRef Link)

[8] T. Stockhammer, "Dynamic adaptive streaming over HTTP --: standards and design principles," in *Proc. of the second annual ACM conference on Multimedia systems, San Jose*, CA, USA, 2011. Article (CrossRef Link)

[9] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *MultiMedia, IEEE*, vol. 18, pp. 62-67, 2011. Article (CrossRef Link)

[10] D. Yun and K. Chung, "Rate Adaptation for HTTP Video Streaming to Improve the QoE in Multi-client Environments," *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*, vol. VOL. 9, 11, November 2015. Article (CrossRef Link)

[11] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. of* the second annual ACM conference on Multimedia systems, San Jose, CA, USA, 2011. Article (CrossRef Link)

[12] W. Chen, L. Ma, G. Sternberg, Y. A. Reznik, and C. C. Shen, "User-aware DASH over Wi-Fi," in *Proc. of 2015 International Conference on Computing, Networking and Communications, ICNC 2015*, pp. 749-753,2015. Article (CrossRef Link)

[13] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: evidence from a large video streaming service," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 187-198, 2014. Article (CrossRef Link)

[14] L. Sunghee, Y. Kimyung, and C. Kwangsue, "Adaptive video quality control scheme to improve QoE of MPEG DASH," in *Proc. of 2015 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 126-127, 2015. Article (CrossRef Link)

[15] G. Appel, *The Moving Average Convergence-divergence Trading Method: Advanced Version*: Scientific Investment Systems, 1985. Article (CrossRef Link)

[16] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE Network*, vol. 17, pp. 27-35, 2003. Article (CrossRef Link)

[17] S. Chaudhari and R. Biradar, "Survey of Bandwidth Estimation Techniques in Communication Networks," *Wireless Personal Communications*, pp. 1-52, 2015/03/05 2015. Article (CrossRef Link)

[18] S. Akhshabi, A. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. of ACM Conf. on Multimedia Systems*, 2011. Article (CrossRef Link)

[19] J. Junchen, V. Sekar, and Z. Hui, "Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive," *IEEE/ACM Transactions on Networking*, vol. 22, pp. 326-340, 2014. Article (CrossRef Link)

[20] Mathwords. *Harmonic Mean*. Available: http://www.mathwords.com/h/harmonic_mean.-htm

[21] R. Dubin, O. Hadar, and A. Dvir, "The effect of client buffer and MBR consideration on DASH Adaptation Logic," in *Proc. of 2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2178-2183, 2013. Article (CrossRef Link)

[22] S. Lee, K. Youn, and K. Chung, "Adaptive video quality control scheme to improve QoE of MPEG DASH," in *Proc. of 2015 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 126-127, 2015. Article (CrossRef Link)

[23] S. Akhshabi, S. Narayanaswamy, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptive video players over HTTP," *Signal Processing: Image Communication*, vol. 27, pp. 271-287, 2012. Article (CrossRef Link)

[24] J. M. Jeong and J. D. Kim, "Effective bandwidth measurement for Dynamic Adaptive Streaming over HTTP," in *Proc. of 2015 International Conference on Information Networking (ICOIN)*, pp. 375-378, 2015. Article (CrossRef Link)

[25] T. Truong Cong, H. Quang-Dung, K. Jung Won, and A. T. Pham, "Adaptive streaming of audiovisual content using MPEG DASH," *IEEE Transactions on Consumer Electronics*, vol. 58, pp. 78-85, 2012. Article (CrossRef Link)

[26] OMNET. *OMNET++ Discrete Event Simulator*. Available: https://omnetpp.org/omnetpp

[27] INET. *INET framework for omnet++*. Available: https://inet.omnetpp.org

[28] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*: Addison-Wesley Publishing Company, 1989. Article (CrossRef Link)

[29] N. Cranley, P. Perry, and L. Murphy, "User perception of adapting video quality," *International Journal of Human-Computer Studies*, vol. 64, pp. 637-647, 2006. Article (CrossRef Link)

[30] J. Neyman, "Outline of a Theory of Statistical Estimation Based on the Classical Theory of Probability," *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 236, pp. 333-380, 1937. Article (CrossRef Link)

Van-Huy Vu received the B.S degree in Information Technology from Thai Nguyen University of Technology, Vietnam in 2006 and M.S. degree in Computer Science from Thai Nguyen University, Vietnam in 2009. Currently, he is a Ph.D. candidate in the Department of Computer Science and Engineering, Yuan Ze University, Taiwan. His research interests include Multimedia Communication, video streaming over wired and wireless networks, mobile TV, IoT and future communications systems.

Ibrahim Mashal is a Ph.D. candidate in the Department of Computer Science and Engineering, Yuan Ze University, Taiwan. In 2007 he received his M.S. in computer science from the University of Jordan, Jordan. From 2007 to 2012 he was a lecturer in Al-Zaytoonah university of Jordan. His research interests include future communications systems, Internet of Things, Wireless mobile network, and sensor networks.

Tein-Yaw Chung received the M.S. and Ph.D degrees in Electrical and Computer Engineering from North Carolina State University, Raleigh, NC, USA, in 1986 and 1990, respectively. From Feb.1990 to Feb 1992, he was with Network Service Division, IBM, RTP, NC, USA. Since May 1992, he has been with Yuan Ze University, Chung-Li, Taiwan and is a full Professor now. His current research interests include overlay network, Multimedia Communication and mobile networking. He is a member of IEEE.