Adaptive Multiple TCP-connection Scheme to Improve Video Quality over Wireless Networks

Dongchil Kim and Kwangsue Chung
Department of Communication Engineering, Kwangwoon University
447-1, Wolgye-dong, Nowon-gu, Seoul, Korea
[e-mail: dckim@cclab.kw.ac.kr, kchung@kw.ac.kr]
*Corresponding author: Kwangsue Chung

Received March 15, 2014; revised August 6, 2014; revised September 11, 2014; accepted October 6, 2014; published November 30, 2014

Abstract

Due to the prevalence of powerful mobile terminals and the rapid advancements in wireless communication technologies, the wireless video streaming service has become increasingly more popular. Recent studies show that video streaming services via Transmission Control Protocol (TCP) are becoming more practical. TCP has more advantages than User Diagram Protocol (UDP), including firewall traversal, bandwidth fairness, and reliability. However, each video service shares an equal portion of the limited bandwidth because of the fair sharing characteristics inherent in TCP and this bandwidth fair sharing cannot always guarantee the video quality for each user. To solve this challenging problem, an Adaptive Multiple TCP (AM-TCP) scheme is proposed in this paper to guarantee the video quality for mobile devices in wireless networks. AM-TCP adaptively controls the number of TCP connections according to the video Rate Distortion (RD) characteristics of each stream and network status. The proposed scheme can minimize the total distortion of all participating video streams and maximize the service quality by guaranteeing the quality of each video streaming session. The simulation results show that the proposed scheme can significantly improve the quality of video streaming in wireless networks.

Keywords: Video Streaming, Qualtiy of Service (QoS), Rate Distortion (RD), Wireless Networks, Mobile Devices, Adaptive Multiple TCP (AM-TCP)

This paper has been conducted by the Research Grant of Kwangwoon University in 2014. It was also supported by the ICT R&D program of MSIP/IITP. [14-000-11-002, Development of Object-based Knowledge Convergence Service Platform using Image Recognition in Broadcasting Contents].

http://dx.doi.org/10.3837/tiis.2014.11.023

1. Introduction

In recent years, the display capacity and processing power of mobile devices (e.g., smart phones, smart pads, and personal media players) have improved. Wireless network technologies are also capable of supporting high-rate data services as well as high coverage efficiency. In these environments, video streaming services (e.g., mobile IPTV and video conferencing) have become more popular among various types of multimedia services [1][2]. Video streaming services have several unique characteristics that need to be considered when transmitting video in a congested network or wireless networks. First, video applications are affected by packet losses. Second, they are delay-sensitive. Therefore, video packets are associated with stringent delay constraints by which they must be decoded. Also, since video streaming applications require persistently high bandwidth, low packet loss ratio, and timely packet delivery, guaranteeing Quality of Service (QoS) for video streaming services in wireless networks presents several challenges [3].

To efficiently transmit the video packets, User Datagram Protocol (UDP) rather than Transmission Control Protocol (TCP) was originally preferred. However, UDP is not a congestion-aware protocol since it does not reduce its sending rate in the presence of network congestion, and therefore potentially results in a congestion collapse. Congestion collapse occurs when a router drops a large number of packets due to its inability to handle a large amount of traffic from many senders at the same time [4]. Recent measurements show that streaming over TCP is becoming practical and popular, including web-based TV, P2P streaming, video sharing websites, and social media systems. The trend of TCP streaming is mainly due to the fact that the deployment and the use of video applications are easier than relying on UDP given the wide use of Network Address Translation (NAT) and firewalls [5]. However, each video streaming service shares the limited bandwidth equitably due to the fair sharing characteristics inherent in TCP. This bandwidth sharing cannot always guarantee the video quality, because the equally allocated bandwidth for each user may be undesirably large or small. To solve this problem, several methods for multiple TCP connections have been proposed, such as Real-time Online Multimedia Environment (TCP-ROME), TCP-FIT, Multiple Adaptive TCP connections for Streaming (MATS), Dynamic MultiPath streaming (DMP-streaming), and Multiple TCP connections (Multi-TCP) [6-10]. However, since these schemes only focus on improving throughput and fairness among flows, without considering video characteristics and wireless network status, they do not guarantee the user perceived quality of video streaming services [11-13].

This paper proposes an Adaptive Multiple TCP (AM-TCP) scheme in order to guarantee the video quality for mobile devices in wireless networks. To adaptively control the number of TCP connections, AM-TCP calculates the optimal rate for each user according to the video Rate Distortion (RD) characteristics for each stream and network status; i.e., the network buffer status and Carrier to Interference and Noise Ratio (CINR). Furthermore, the proposed scheme allocates the number of TCP connections for each user by using a modified max-min fairness algorithm in order to maximize the video quality among streaming services in the case where a lack of available bandwidth occurs. Also, AM-TCP can minimize the total distortion of all participating video streams and maximize the service quality by guaranteeing the quality of each video streaming. The remainder of this paper is organized as follows. An overview of the disadvantages of TCP for multimedia streaming and the existing multiple TCP connection schemes are discussed in Section 2. In Section 3, this paper describes the concepts and algorithms introduced in AM-TCP. Simulation results are described in Section 4 and the concluding remarks are given in Section 5.

2. Related Work

2.1 Disadvantages of TCP for Multimedia Streaming

TCP, a well-known congestion control scheme, is a transport layer protocol used by various applications that require guaranteed delivery in the Internet. TCP provides congestion control that attempts to maximize the throughput, whereas it prevents the network collapse. TCP uses a congestion window to send packets based on an Additive Increase Multiplicative Decrease (AIMD) algorithm. However, TCP is unsuitable for multimedia streaming due to its fluctuating throughput and its lack of precise rate control. It is also designed for end-to-end reliability and fast congestion avoidance. To provide end-to-end reliability, a TCP sender retransmits the lost packets based on the packet acknowledgment from a TCP receiver [7]. These packet retransmissions cause an increase of Round Trip Time (RTT). This has led to a degradation of total throughput and system performance. Also, the throughput fluctuation of TCP is attributed to the reduction of the sending rate by half upon detection of a loss event. To alleviate this throughput fluctuation, TCP can be modified to reduce the sending rate by a small factor less than a half upon detection of a loss. However, various disadvantages are associated with these approaches. First, these changes affect all TCP connections and must be performed by recompiling the OS kernel of the sender. Second, changing the decreasing multiplicative factor may potentially lead to the instability of TCP. Third, it is not clear how these factors can be changed to dynamically control the sending rate. To address these disadvantages, several studies have been proposed to research the multiple TCP connection schemes. These schemes can achieve higher throughput and precise rate control to dynamically distribute streaming data over multiple TCP connections.

2.2 Schemes for Multiple TCP Connections

Many researches have been carried out to study the multiple TCP connection schemes shown in **Table 1**. TCP-ROME is a parallel TCP-based streaming protocol for real-time multimedia performed by establishing multiple TCP connections from multiple servers to a single user [6]. TCP-ROME is readily applicable to many applications without re-implementations. Furthermore, it coordinates the media streaming at the granularity of individual segments over the different connections and merges them at the receiver. TCP-ROME also ensures that media segments are transmitted over these connections where the bandwidth and delay conditions promise a timely delivery. If segments are lost or expected to be delivered late, TCP-ROME can re-request the segments over alternative connections.

Multi-TCP was proposed to provide multimedia streaming over the Internet [7]. The aim of Multi-TCP is to provide resilience against short-term insufficient bandwidth by using multiple TCP connections for the same application. Furthermore, Multi-TCP enables the application to achieve and control the sending rate during periods of congestion, which in many cases cannot be achieved using a single TCP connection. Finally, the Multi-TCP algorithm is implemented at the application layer, and hence kernel modification to TCP is not necessary. The application can achieve the desired throughput by using Multi-TCP in many scenarios, which cannot be achieved using the traditional single TCP approach.

TCP-FIT is a novel TCP congestion control algorithm for heterogeneous networks that contain Bandwidth Delay Product (BDP) links and wireless links [8]. TCP-FIT was inspired by parallel TCP, but with the important distinctions of only one TCP connection with one congestion window is established for each TCP session, and no modifications to other layers of the end-to-end system need to be made. In TCP-FIT, the AIMD mechanism is employed to directly adjust the congestion window. However, the decreasing factor of TCP-FIT uses $2/(3N+1)$ instead of $1/2N$ in the Multi-TCP when a packet loss occurs. It helps maintain the throughput of the TCP-FIT flows. N is the number of TCP sessions.

Table 1. Characteristics of existing multiple TCP connection schemes

Schemes	Type of TCP Connections	Goal	Multimedia Support	Video RD Awareness	Wireless Channel Awareness
TCP-ROME [6]	Multiple connections	Improving throughput and timely delivery	Less bandwidth fluctuation and guarantee delay deadline	No	No
Multi-TCP [7]	Multiple connections	Improving throughput	Less bandwidth fluctuation and pre-buffering	No	No
TCP-FIT [8]	Multiple connections	Improving throughput and fairness	No	No	No
DMP-streaming [9]	Multiple connections	Improving throughput	Less bandwidth fluctuation and pre-buffering	No	No
MATS [10]	Multiple connections	Guaranteeing the video quality	Playout buffer consideration	No	No

DMP-streaming is a simple and practical multipath TCP-based streaming scheme [9]. It dynamically distributes packets over multiple paths to accommodate bandwidth fluctuations by implicitly inferring the available bandwidths on these paths. DMP-streaming is simple in that it requires no explicit probing and feedback on network bandwidths. It does not perform any error recovery (e.g., using redundant packets and coding) or rate adaptation at the application level. DMP-streaming can also be used regardless of whether or not the multiple TCP flows share bottleneck links. Although this scheme can prevent short-term bandwidth fluctuation, it cannot properly handle the long-term decrease in available bandwidth caused by the fair-share characteristics of TCP. This is because it uses a fixed number of TCP connections for a streaming session regardless of the video RD characteristics and current network status.

To guarantee the quality of each streaming session, the MATS scheme is proposed [10]. MATS guarantees the quality of each streaming session in the face of long-term streaming bandwidth decrease, due to the bandwidth fair-share characteristics of TCP, whereas it maintains fairness of quality with other services. To achieve these goals, multiple auxiliary TCP connections are adaptively created when bandwidth shrinkage is detected. The creation of the auxiliary TCP connections is controlled by the server according to the Playout Buffer (PoB) status of a streaming user. However, this scheme focuses on improving throughput and fairness without specifically considering the characteristics of the streaming video such as video RD. These have led to a significant degradation on media quality for users in wireless networks. The importance of RD optimized rate allocation has been recognized in this study for multiplexing multiple video streams over a common bottleneck link. The majority of such schemes is centralized in nature, and requires knowledge of video RD information of all participating streams. To guarantee the video quality for each user in wireless networks, the AM-TCP scheme adaptively controls the number of TCP connections for the users according to the video RD characteristics of each stream and network status.

3. Adaptive Multiple TCP (AM-TCP) Scheme

This section presents the proposed Adaptive Multiple TCP (AM-TCP) scheme to improve video quality for mobile devices in wireless networks. To guarantee the quality of video streaming services, the proposed scheme adaptively controls the number of TCP connections for users according to the video RD characteristics for each stream and network status. To achieve this goal, the proposed scheme consists of an optimal rate calculation and TCP connection control algorithm for improving the user perceived quality of video streaming.

3.1 Architecture of AM-TCP Scheme

Fig. 1 shows the architecture of AM-TCP in wireless network environments. The proposed scheme calculates the optimal rate and adjusts the number of TCP connections in a server. For this purpose, the information relating to QoS control is periodically exchanged between a server and users. Also, AM-TCP uses the video RD characteristics of all participating streams to reduce quality fluctuation for each stream. A Base Station (BS) estimates the wireless channel status for each mobile user and sends it to the server.

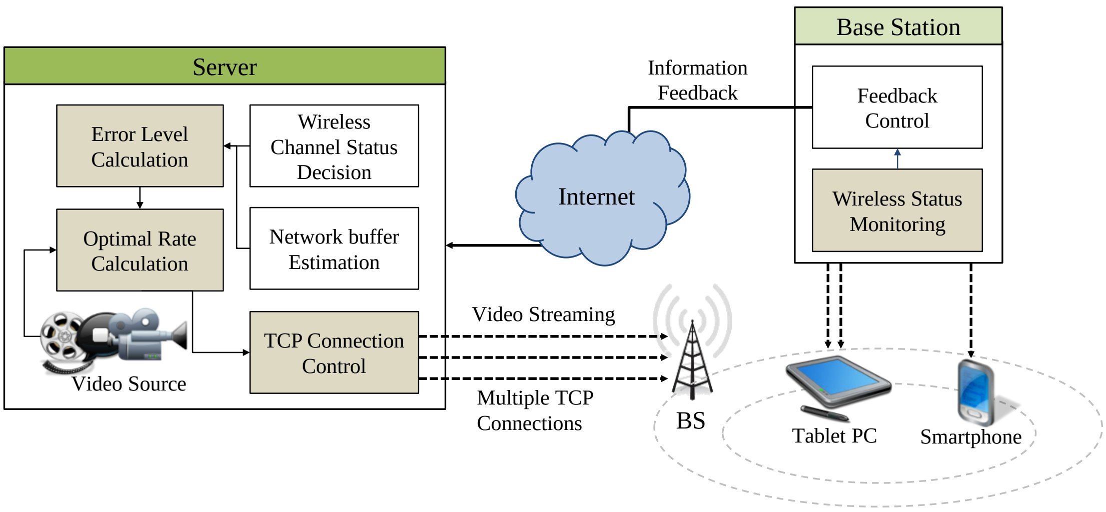

Fig. 1. Architecture of AM-TCP scheme

- The Error Level Calculation (ELC) module determines the network error level based on network buffer status and CINR for mobile users. The server periodically estimates the number of outstanding packets in the network buffer and CINR.

- The Optimal Rate Calculation (ORC) module determines the optimal rate for each stream according to the video RD characteristics and error level. When multiple video streams compete for shared network resources, it is desirable that allocated rates reflect their respective video RD characteristics and network status.

- The TCP Connection Control (TCC) module controls the number of TCP connections for each user based on the calculated optimal rate. When the sum of the number of TCP connections based on the optimal rate for each user falls below the available number of TCP connections, TCC computes the number of TCP connections using modified max-min fairness.

- The Wireless Status Monitoring (WSM) module estimates the CINR of each user and relays it to the Wireless Channel Status Decision (WCSD) module in the server.

3.2 Optimal Rate Calculation

Existing multiple TCP connection schemes typically aim at allocating an equal rate among competing flows, with the underlying assumption that they bear the same rate utility function. However, this bandwidth fair sharing cannot always guarantee the user's perceived quality due to specific requirements for video streaming services. To guarantee the quality of video streaming services, AM-TCP shares the bandwidth among competing video streams by adapting the rate of each video stream according to its video RD characteristics and network status; i.e., the network buffer condition and CINR.

In **Fig. 2**, a video coding algorithm focuses on the tradeoff between the distortion and bit rate, where a decreasing distortion usually corresponds to an increasing rate and vice-versa. To allocate the rate for each video stream, AM-TCP calculates the optimal rate between increasing the rate to reduce video distortion and decreasing the rate to reduce the network error. A higher network error leads to a lower allocated rate whereas a lower network error encourages a higher video rate. Furthermore, the same value of network error level leads to different optimal rates for video streams that have different video RD parameters. This can be formulated as in (1). r_i^* is the optimal rate of video stream i, $d_i(r_i)$ denotes the video RD function for each video stream i, E_i is the network error level of each user i, and r_i is the rate of video stream i.

$$r_i^* = \arg\min_{r_i} \left[d_i(r_i) + E_i \times r_i \right] \tag{1}$$

The proposed scheme adopts the parametric model for characterizing the video RD tradeoff [14]. The parameters d_i^0, r_i^0, and Θ_i are fitted from the empirical video RD points of a pre-encoded video stream i. d_i^0 is the distortion for each encoded video stream. It is measured as the Mean Squared Error (MSE) between original and re-constructed pixel values. The MSE distortion typically decreases nonlinearly with increasing rate, r_i^0. Θ_i is the video RD factor. These can be fitted to a parametric video RD model as in (2). These parameters are affected by factors including the coding scheme, the encoder configuration, and the video scene complexity. They are also updated periodically to track the changes of contents in the video stream, typically once for every Group of Pictures (GOP) [11][12].

$$d_i(r_i) = d_i^0 + \frac{\theta_i}{r_i - r_i^0} \tag{2}$$

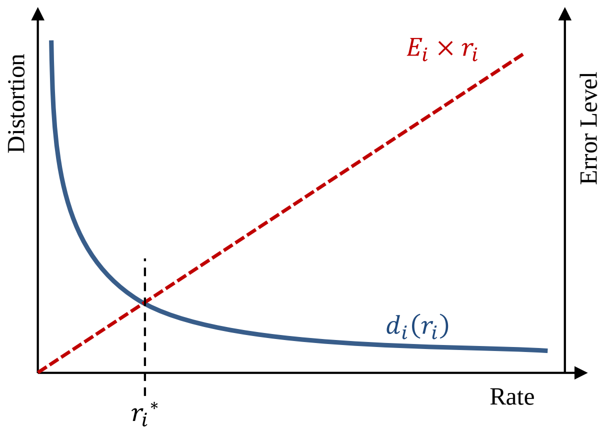

Fig. 2. Optimal rate calculation

To calculate the optimal rate for each user i, the ELC module computes the E_i as shown in **Fig. 3**. The ELC module uses the number of outstanding packets in the network buffer and CINR. When the level of the current network status for each user i, N_i^*, is lower than the minimum level of network status, N_{min}, the error level is E_{min}. If the level of the current network status is higher than the maximum level of the network status, N_{max}, the error level is the maximum error level, E_{max}. On the other hand, when $N_{min} < N_i^* < N_{max}$, the error level is proportionally increased as in (3). For simulations, this scheme obtained two threshold factors of the error levels where $E_{min} = 1$ and $E_{max} = 10$.

$$E_i = \left\lceil \frac{E_{\max} - E_{\min}}{N_{\max} - N_{\min}} \times (N_i^* - N_{\min}) + E_{\min} \right\rceil \tag{3}$$

To calculate the current network status level, the proposed scheme first estimates the number of outstanding packets in the network buffer as in (4). O_i denotes the estimated number of outstanding packets that are currently queued in the network buffer, and are calculated in a way that is similar to existing delay-based TCP and hybrid TCP variants. $RTT_{i,avg}$ is the average RTT of the current updating period, $RTT_{i,min}$ is the minimal observed RTT.

$$O_i = (RTT_{i,avg} - RTT_{i,\min}) \times \frac{W_{i,avg}}{RTT_{i,avg}} \tag{4}$$

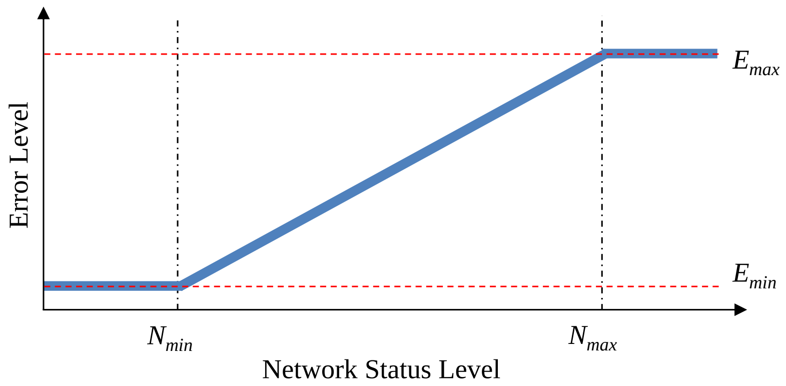

Fig. 3. Error level calculation

$RTT_{i,avg}$ - $RTT_{i,min}$ represents the estimated queuing delay. Since a TCP session sends w packets during a RTT, $W_{i,avg}$ / $RTT_{i,avg}$ could be considered as the current TCP session's transmission rate. $W_{i,avg}$ is the average window size. The proposed scheme then calculates the weighted value of wireless channel status, W_i, according to the CINR. The weighted value will be chosen according to the following criteria:

$$W_i = \begin{cases} 0, & CINR_i > CINR_{\max} \\ 1, & CINR_i < CINR_{\min} \\ \dfrac{CINR_{\max} - CINR_i}{CINR_{\max} - CINR_{\min}}, & otherwise \end{cases} \tag{5}$$

When the current CINR, $CINR_i$, is higher than the maximum CINR, $CINR_{max}$, the weighted value is 0. On the other hand, if the current CINR is lower than the minimum CINR, $CINR_{min}$, the weighted value is 1. Otherwise, the weighted value is calculated by CINR levels. The CINR according to the Modulation and Coding Scheme (MCS) level and the corresponding Physical data Rate (RPHY) is summarized as shown in **Table 2**. AM-TCP used the threshold factor of the CINR where $CINR_{min}$ = -1dB and $CINR_{max}$ = 26dB.

AM-TCP calculates the network status, N_i, based on the number of outstanding packets in the network buffer and weighted value based on CINR for each user as in (6). To obtain a more stable network status value, the smoothed current network status, N_i^*, is calculated by using the Exponentially Weighted Moving Average (EWMA) method as in (7). The weight k determines the reflection ratio of the current measured network status. For k=8, this gives weights of ω_1, ω_2, ω_3, ω_4= 1; ω_5= 0.8, ω_6= 0.6, ω_7= 0.4, ω_8= 0.2.

$$N_i = \lceil O_i + (O_i \times W_i) \rceil \tag{6}$$

$$N_i^* = \left\lceil \frac{\sum_{j=0}^{k-1} N_i(j) \times \omega_{j+1}}{\sum_{j=0}^{k-1} \omega_{j+1}} \right\rceil \tag{7}$$

Table 2. Physical data rate with CINR level

CINR (dB)	Modulation	Coding Rate	RPHY (Mbps)
26	64-QAM	5/6	18.43
23	64-QAM	3/4	16.59
20	64-QAM	2/3	14.74
18	16-QAM	5/6	12.29
16	16-QAM	3/4	11.05
14	16-QAM	2/3	9.83
12	16-QAM	1/2	7.37
10	QPSK	2/3	4.91
6	QPSK	1/2	3.69
3	QPSK	1/3	2.46
1	QPSK	1/6	1.23
-1	QPSK	1/12	0.61

3.3 Allocation of the Number of TCP Connections

A fixed number of TCP connections may be unnecessarily large or small. If the network error level is high, the algorithm can reduce the number of TCP connections for each user. Similarly, if the network error level is low, the algorithm can increase the number of TCP connections to obtain the desired throughput, whereas it improves the media quality of video streaming. To guarantee fairness of services, whereby no users receive higher quality of services than requested, the proposed AM-TCP algorithm should consider the video RD characteristics for adjusting the number of TCP connections. Users with unsatisfied demands split the remaining resource in proportion to their weights. In this section, the proposed scheme allocates the number of TCP connections dynamically depending on the video RD characteristics and network error level as shown in **Fig. 4**.

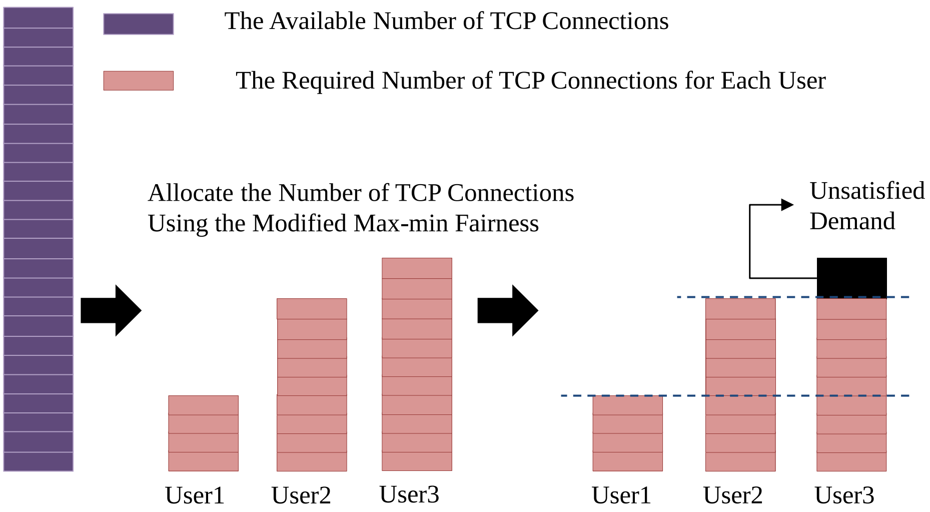

Fig. 4. Allocation of the number of TCP connections for each user using the modified max-min fairness

Fig. 5 shows the pseudo code for the allocation of the number of TCP connections for each user. If the sum of the number of TCP connections according to the rate for video streams, $\sum_{i=0}^{n-1}\lceil r_i^* / TC_{rate}\rceil$, falls below the available number of TCP connections, $\lfloor (C_l - \sum_{i=0}^{n-1} r_i') / TC_{rate}\rfloor$, the server calculates the number of TCP connections for users based on the calculated optimal rate for each stream. C_l denotes the capacity of the bottleneck link, $\sum_{i=0}^{n-1} r_i'$ is the rate of all non-video streams traversing that link, n is the total number of TCP connections, and TC_{rate} is the rate of one TCP connection. On the other hand, when the sum of the number of TCP connections for each user is higher than the available number of TCP connections, the server adjusts the number of TCP connections that use the modified max-min fairness algorithm. The max-min fairness is one of the most widely used fairness concepts for resource sharing in communication networks. It also satisfies many intuitive fairness properties. The principle of max-min fairness is to allocate network resources in such a way that the bit rate of a flow cannot be increased without decreasing the bit rate of a flow having a smaller bit rate. TC_x^* is the number of TCP connections for each user in ascending order and $newTC_x$ is the new adjusted number of TCP connections for each user.

* Allocation of the number of TCP connections for each user using the modified max - min fairness *

if $\left(\left\lfloor\left(C_l - \sum_{i=0}^{n-1} r_i'\right) / TC_{rate}\right\rfloor \geq \left(\sum_{i=0}^{n-1}\left\lceil r_i^* / TC_{rate}\right\rceil\right)\right)\{$

  for $(i = 0; i < n; i++)$

    $TC_i = \left\lceil r_i' / TC_{rate}\right\rceil$

$\}$

else $\{$

  //The required number of TCP connections for each user in ascending order

  $TC_x^* = \{TC_0^*, TC_1^*, TC_2^*, \cdots, TC_{n-1}^*\}, 0 \leq x < n$

  $newTC_x = \{0, 0, 0, \cdots, 0\}, 0 \leq x < n$

  for $(i = 0; i < n; i++)\{$

    if $\left((TC_i^* - newTC_i) \times (n - i) \leq \left(\left\lfloor\left(C_l - \sum_{i=0}^{n-1} r_i'\right) / TC_{rate}\right\rfloor - \sum_{x=0}^{n-1} newTC_x\right)\right)$

      for $(j = i;\ j < n;\ j++)$

        $newTC_j = TC_i^*$

    else $\{$

      for $(j = i;\ j < n;\ j++)\{$

        if $\left((TC_j^* - newTC_j) \leq \left(\left\lfloor\left(C_l - \sum_{i=0}^{n-1} r_i'\right) / TC_{rate}\right\rfloor - \sum_{x=0}^{n-1} newTC_x\right)\right)$

          $newTC_j = TC_j^*$

        else

          $newTC_j = newTC_j + \left(\left\lfloor\left(C_l - \sum_{i=0}^{n-1} r_i'\right) / TC_{rate}\right\rfloor - \sum_{x=0}^{n-1} newTC_x\right)$

      $\}$

    $\}$

$\}$

Fig. 5. Pseudo code for allocation of the number of TCP connections

To adaptively allocate the number of TCP connections for users, AM-TCP can minimize the total distortion of all streams and achieve a target utilization level at the bottleneck link by using the modified max-min fairness. It controls the number of TCP connections using the following procedure:

- Step 1: start from the number of TCP connections equal to zero for all flows.
- Step 2: increase the number of TCP connections of all flows at the same connections until the number of TCP connections of some of the flows is constrained by the available number of TCP connections; freeze the number of TCP connections of these flows.

- Step 3: apply step 2 repeatedly to non-frozen flows until the number of TCP connections of all flows is constrained by the available number of TCP connections.

- Step 4: allocate the number of TCP connections for each user by using the modified max-min fairness algorithm.

3.4 Adaptive Scheduling Algorithm

To simultaneously distribute video packets over multiple TCP connections, our scheme controls the packet transmission. **Fig. 6** shows the operation of the adaptive scheduling algorithm for packets of each application. To reduce the number of out-of-order packets, our scheme predicts the packet receiving sequences using the available bandwidth of each TCP connection. To calculate the available bandwidth for each user, we use the TCP-Friendly Rate Control (TFRC) which provides smoother sending rates because it adjusts its sending rate based on the throughput equation as in (8). S is the packet size, b is the number of acknowledged packets, p is the packet loss rate, t_{RTT} is the round-trip time, and t_{RTO} is the retransmission timeout [15]. After obtaining the available bandwidth of each TCP connection, AM-TCP begins to split the traffic over multiple TCP connections as shown in **Fig. 6**.

$$R_{TFRC} = \frac{S}{RTT\sqrt{\frac{2p}{3}} + t_{RTO}\left(3\sqrt{\frac{3bp}{8}}\right)p\left(1+32p^2\right)} \tag{8}$$

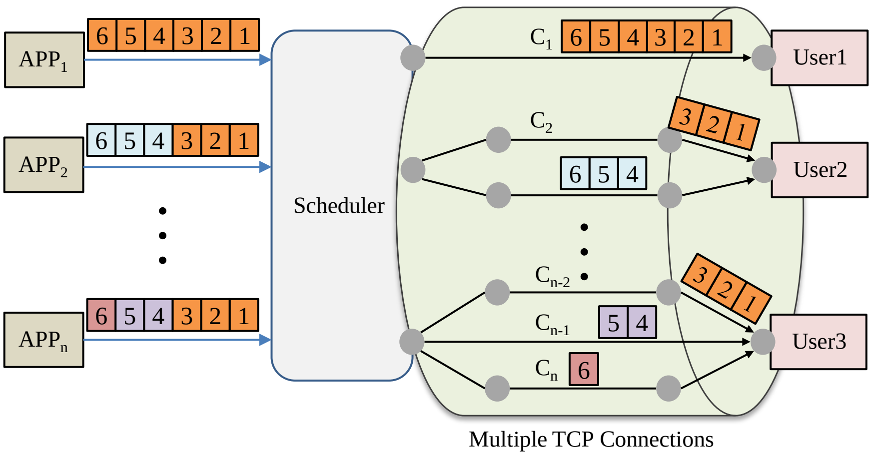

Fig. 6. Operation of adaptive scheduling algorithm

Fig. 7 shows the pseudo code of the adaptive scheduling algorithm. It is independently operated in each application. Our scheduling algorithm can also properly adjust the Packet Sequence Number (PSN) increment value, Δ_{PSN}, for the next packet over the multiple TCP connection. Δ_{PSN} depends on the bandwidth of the available TCP connection. Based on Δ_{PSN}, the PSN of the next packet sent over each TCP connection is determined according to the algorithm as shown in **Fig. 7**. Once the server receives a data acknowledgement (ACK) from the sink, it estimates the available bandwidth for each TCP connection. In order to better ensure an in-order receiving sequence at the sink, if the *j-th* TCP connection is chosen, the PSN of the next packet to be sent for the TCP connection *j*, $S_{PSN,j+ConnectionCount}$, is simply L_{PSN} + $\Delta_{PSN,j+ConnectionCount}$, where L_{PSN} is the largest packet PSN sent over the multiple TCP connections. $\Delta_{PSN,j+ConnectionCount}$ is determined by TFRC calculated in (8). The *ConnectionCount* is the TCP connection number required to transmit the next packet.

```
* Adaptive scheduling algorithm for each application *
//Input : available bandwidth of each TCP connection by estimated TFRC in (8)
//ACK for PSN packet recevied at the server
ConnectionCount = 0
for (i = 0; i < n; i++) {
  if (newTC_i is selected) {
    for ( j = 0; j < newTC_i; j++) {
      Δ_PSN,j+Connection Count = ⌊Rate_Connection,j+Connection Count / PacketSize⌋
      S_PSN,j+Connection Count = L_PSN + Δ_PSN,j+Connection Count
      L_PSN = S_PSN,j+Connection Count
    }
    ConnectionCount = j
  }
}
```

Fig. 7. Pseudo code for adaptive scheduling algorithm

4. Simulation Results

This section presents the simulation results of AM-TCP performance in a wireless network. In order to evaluate the performance of AM-TCP, a number of experiments have been performed on the basis of the Network Simulator (NS) of Lawrence Berkeley National Laboratory (LBNL). **Fig. 8** shows the simulation topology for a video streaming service. All links have a capacity of 100Mbps each. The router and BS have a typical drop-tail queue that will reject the incoming packets in the case of an overflow leading to packet loss. The video server transmits video streams to R1 and R2 over the Internet. The video receivers are connected in wireless links. Also, two background traffic flows occur: the FTP traffic transmitted by TCP packets, where the FTP traffic rate is 40Mbps, and the exponential traffic transmitted by UDP packets. The exponential traffic rate is also 40Mbps. The transmission rate for the traffic is 1 Mbps; burst and idle time are both set to 0.5ms.

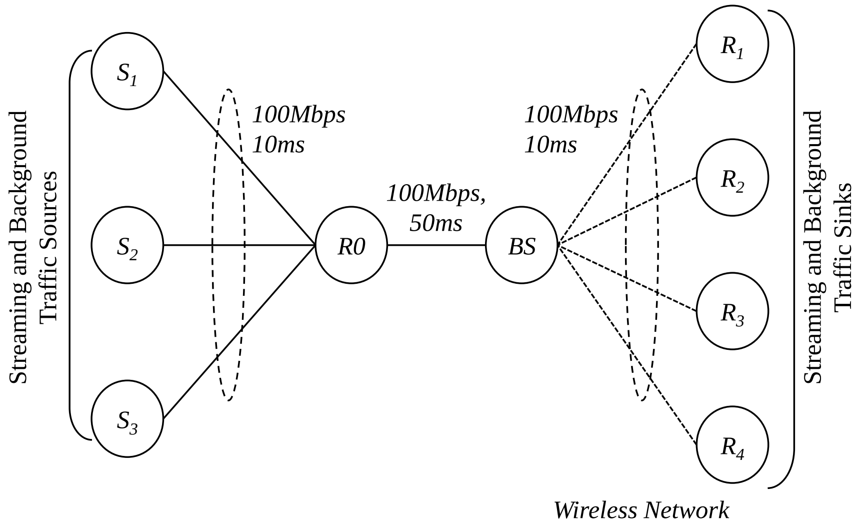

Fig. 8. Simulation topology

Table 3 shows the configuration of simulation for the wireless network. The proposed scheme can be applied to any wireless networks, because AM-TCP calculates the network status using the weighted value according to CINR for each user. Also, PHY rate is determined according to the CINR. The simulation uses the video sequences to evaluate the performance of the proposed scheme as shown in **Table 4**. The test sequence is the "CITY" video and is encoded by using the scalability extension of H.264/AVC. The sequence has a resolution of CIF (352×288) pixels per frame and a frame rate of 30 frames per second. The video packets are further segmented into network packets with a size of 1500 bytes for transmission. The receiver sends an acknowledgement packet upon receipt of each packet. This simulation uses the video layer of 16 and 19.

Table 3. Configuration of simulation for the wireless network

Parameter	**PHY rate**	**Frequency Band**	**Duplex Type**	**Multiple Access**	**Cell Coverage**
Value	0.61~18.43Mbps	2.3 GHz	TDD / 5msec	OFDMA	1 Km

Table 4. Bit-stream characteristics of the video layers

Video Layer	**Spatial Level**	**Temporal Level**	**Quality Level**	**Resolution**	**Frame Rate**	**Bitrate (kbps)**	**Y-PSNR (dB)**
16	1	4	1	352 x 288	30	353.3	30.6
17	1	4	2	352 x 288	30	503.4	32.3
18	1	4	3	352 x 288	30	735.3	35.5
19	1	4	4	352 x 288	30	921.4	38.1

Fig. 9 shows a comparison of the throughput among the proposed AM-TCP, Multi-TCP, and TCP-Veno. TCP-Veno was originally designed considering wireless networks [16]. TCP-Veno distinguishes between congestion loss and random loss caused by network channel error. However, it shows a low throughput for the receiver required video layer of 19, as shown in **Fig. 9**, because it does not consider video RD characteristics and network status. This results in a severe reduction of the throughput in the presence of burst packet loss. Multi-TCP achieves higher throughput than TCP-Veno, but it still shows a reduction of throughput, because it uses a fixed number of TCP connections for a streaming session, regardless of the wireless network status and service requirements. Also, the Multi-TCP scheme treats any loss packet as a signal of network congestion and adjusts its transmission rate accordingly. This rate reduction is unnecessary if the packet loss is due to the error occurring in the wireless channel, which in turn causes bad performance for end-to-end delivery quality. Thus, it triggers spurious retransmission caused by the reordering of packets. On the other hand, AM-TCP achieves a much higher throughput than other schemes as the proposed scheme adjusts the number of TCP connections based on the reflection of differences in the video RD characteristics of two streams and network status. Through this simulation result, the proposed scheme can successfully achieve service quality fairness.

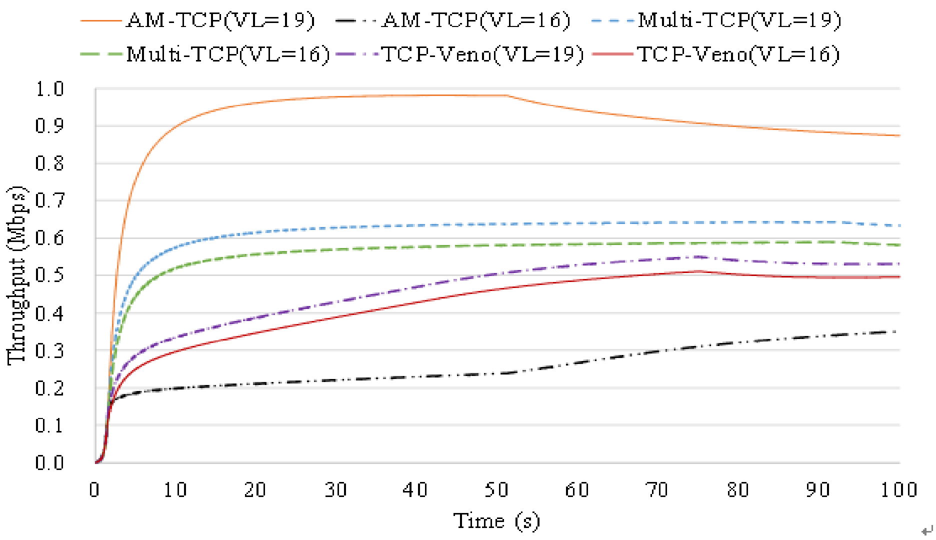

Fig. 9. Comparison of the throughput among AM-TCP, Multi-TCP, and TCP-Veno

Fig. 10 shows a comparison of the number of video buffer underflows at the receiver. AM-TCP and Multi-TCP achieve lower the number of video buffer underflows than TCP-Veno, because these schemes use the multiple TCP connection scheme for a streaming session. Therefore, AM-TCP and Multi-TCP schemes can reduce the video buffer underflows by achieving higher throughput than TCP-Veno scheme. To evaluate the service fairness, we used Jain's fairness index equation as in (9) which is defined in [17], where n is the number of connections and x_i is the throughput of the *i-th* connection. As f reaches closer to 1, the scheme has more fairness. We compare f among the proposed AM-TCP, Multi-TCP, and TCP-Veno schemes. **Fig. 11** shows the service fairness index of these schemes. The service fairness index of the proposed AM-TCP scheme was very close to 1, indicating that service quality fairness among users was achieved.

$$f = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n\sum_{i=1}^{n} x_i^2} \tag{9}$$

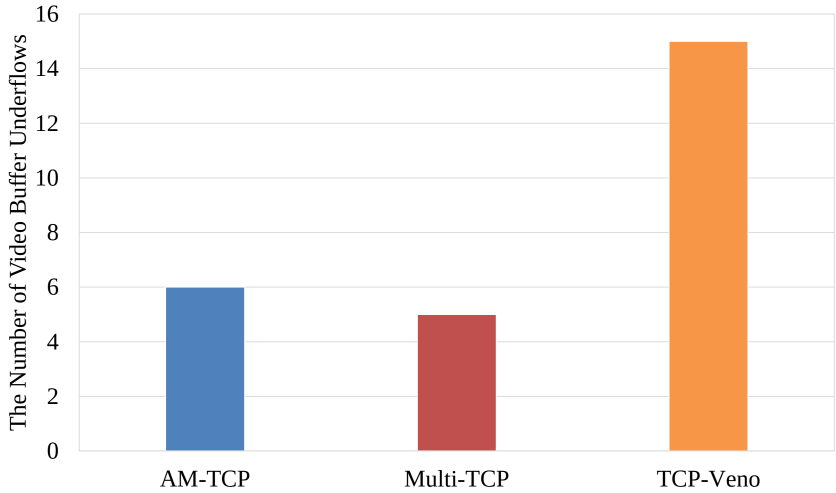

Fig. 10. Comparison of the number of video buffer underflows among AM-TCP, Multi-TCP, and TCP-Veno

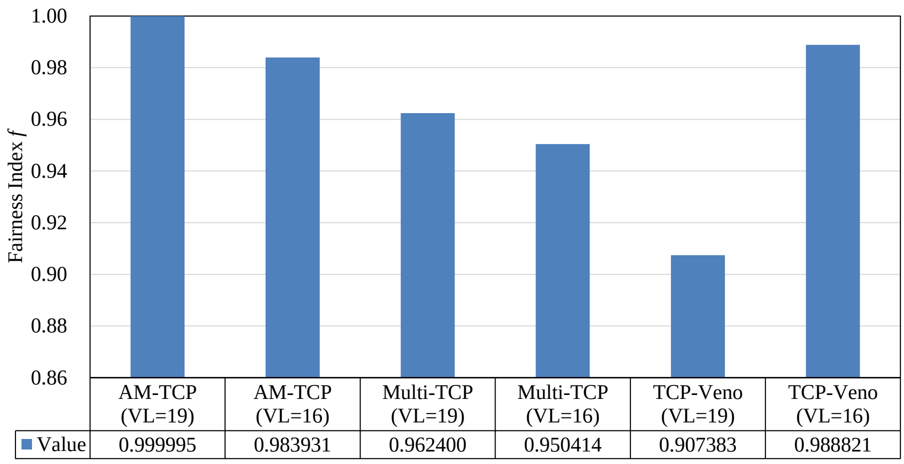

Fig. 11. Comparison of the service fairness among AM-TCP, Multi-TCP, and TCP-Veno

Fig. 12 presents a performance comparison of the packet loss ratio. It is clearly seen that AM-TCP reduces the packet loss ratio because AM-TCP considers the QoS required for users. **Fig. 13** shows a comparison of the PSNR among the proposed scheme, Multi-TCP, and TCP-Veno. The PSNR of AM-TCP is higher than that of the existing schemes as the proposed scheme adaptively controls the number of TCP connections for each stream. This is based on the estimated network error level and video RD characteristics.

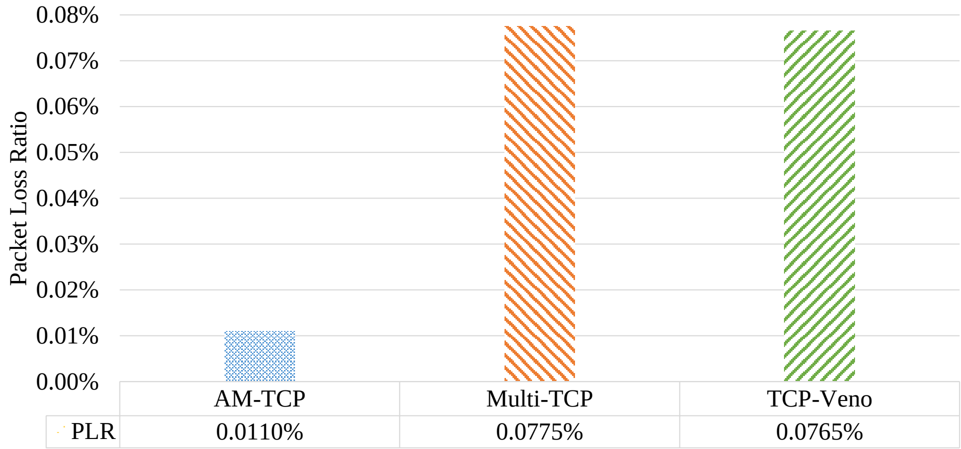

Fig. 12. Comparison of the packet loss ratios among AM-TCP, Multi-TCP, and TCP-Veno

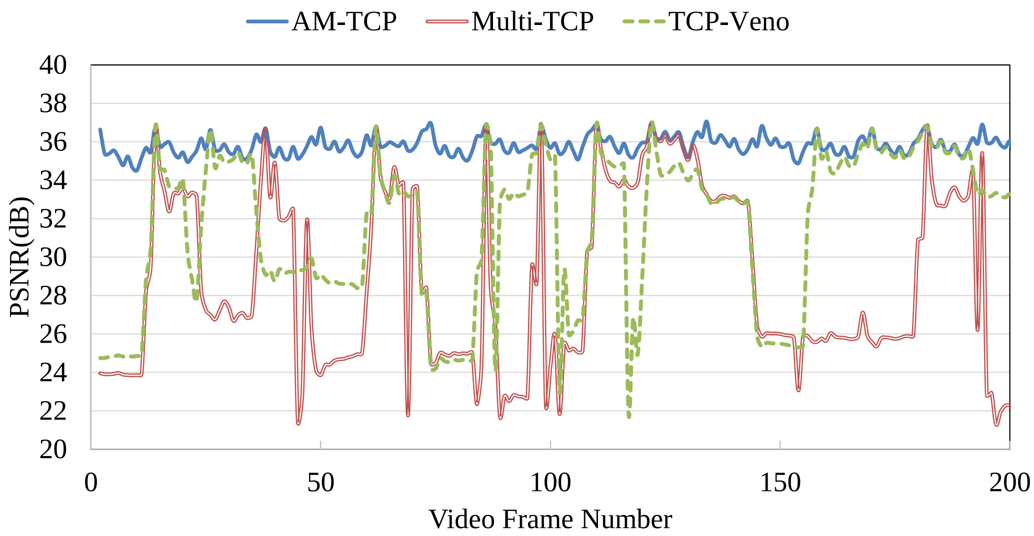

Fig. 13. Comparison of the PSNR among AM-TCP, Multi-TCP, and TCP-Veno

The PSNR compares the maximum possible signal to the noise between a source and a destination image, I_D and I_S respectively. The PSNR is defined as in (10), where $MSE\ (I_D, I_S)$ is the mean square error of the two images and $Vpeak = 2h - 1$, with h as the bit color depth.

$$PSNR(I_D, I_S) = 20\log_{10}\frac{V_{peak}}{MSE(I_D, I_S)}[dB] \tag{10}$$

It is more tractable to represent the PSNR frame by frame than to calculate the average PSNR of all the frames, because an average PSNR may not map well to the overall subjective impression during a video playback. Therefore, in this paper, the frame-wise PSNR is used to assess the video quality. **Table 5** shows the heuristic mapping of PSNR to the Mean Opinion

Score (MOS) [18]. The MOS is a numerical measure of perceptual quality at the receiving end. The metric virtually indicates the video quality perceived by the end-user on a scale from 1 to 5. Mapping the PSNR of **Fig. 13** to MOS grades, based on **Table 5**, provides a simplified and widely used numerical description of the perceived visual quality. All received video frames are characterized by a MOS grade of either good or excellent, achieving a satisfactory performance on video delivery throughout the entire connection [2].

Table 5. Possible PSNR to MOS conversion

PSNR (dB)	**MOS**
>37	5 (Excellent)
31-37	4 (Good)
25-31	3 (Fair)
20-25	2 (Poor)
<20	1 (Bad)

Fig. 14 shows a comparison of the picture of reconstructed sequences. We use two test sequences that are the "CITY" video and the "CREW" video. The sequences have a resolution of CIF (352×288) pixels per frame and a frame rate of 30 frames per second. The impairments are induced by the loss of the media frame when network congestion and wireless channel error occur. AM-TCP improves throughput of each video stream and reduces the packet loss rate significantly for test sequences. It also prevents the error propagation to the successive frames. These simulation results indicate that the proposed scheme provides seamless and high quality video streaming services.

(a) (b)

(c) (d)

Fig. 14. Comparison of the pictures of reconstructed sequences

(a) Original video: "CITY" and "CREW" (b) AM-TCP: "CITY" and "CREW"

(c) Multi-TCP: "CITY" and "CREW" (d) TCP-Veno: "CITY" and "CREW"

5. Conclusion

Guaranteeing the QoS requirement for each user in wireless networks is an important factor for a video streaming application. AM-TCP was proposed in this paper to guarantee video quality for mobile devices in wireless networks. AM-TCP calculated the optimal rate required to adaptively control the number of TCP connections according to the video RD characteristics of each stream and network status; i.e., the network buffer condition and CINR. Furthermore, AM-TCP allocated the number of TCP connections using the modified max-min fairness algorithm in order to maximize the quality of video streaming services in the case where a lack of available bandwidth occurs. The performance evaluation results proved that the proposed scheme can significantly improve the quality of video streaming by avoiding the packet loss during network error and considering the video RD characteristics. The proposed scheme can also minimize the total distortion of all participating video streams and hence maximize the service quality by guaranteeing the quality of each video streaming session. In future work, we intend to extend the proposed AM-TCP scheme for heterogeneous wireless networks, because the future network situation becomes more complicated in the context of multiple available links that involve a low bandwidth, high lossy, or long-delay wireless network [19][20].

References

[1] P. Rengaraju, C. Lung, F. Yu and A. Srinivasan, “On QoE monitoring and E2E service assurance in 4G wireless networks,” *IEEE Wireless Communications*, vol. 19, Issue 4, pp. 89-96, August, 2012. Article (CrossRef Link).

[2] D. Kim and K. Chung, “A network-aware quality adaptation scheme for device collaboration service in home networks,” *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 374-381, May, 2012. Article (CrossRef Link).

[3] H. Shiang and M. Schaar, “A quality-centric TCP-friendly congestion control for multimedia transmission,” *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp.896-909, June, 2012. Article (CrossRef Link).

[4] T. Minhas, O. Lagunas, P. Arlos and M. Fiedler, “Mobile video sensitivity to packet loss and packet delay variation in terms of QoE,” in *Proc. of the International Packet Video Workshop*, pp.83-88, May, 2012. Article (CrossRef Link).

[5] J. Yan, W. Mühlbauer and B. Plattner, “Analytical framework for improving the quality of streaming over TCP,” *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1579-1590, December, 2012. Article (CrossRef Link).

[6] J. Park, R. Karrer and J. Kim, “TCP-ROME: A transport-layer parallel streaming protocol for real-time online multimedia environments,” *Journal of Communications and Networks*, vol. 13, no. 3, pp. 277-285, June, 2011. Article (CrossRef Link).

[7] S. Tullimas, T. Nguyen, R. Edgecomb and S. Cheung, “Multimedia streaming using multiple TCP connections,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 4, Issue 2, pp. 12:1-12:20, May, 2008. Article (CrossRef Link).

[8] J. Wang, J. Wen, J. Zhang and Y. Han, “TCP-FIT: An improved TCP congestion control algorithm and its performance,” in *Proc. of the IEEE INFOCOM*, pp. 2894-2902, April, 2011. Article (CrossRef Link).

[9] B. Wang, W. Wei, Z. Guo and D. Towsley, “Multipath live streaming via TCP: scheme, performance and benefits,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 5, Issue 3, pp. 25:1-25:23, August, 2009. Article (CrossRef Link).

[10] Y. Jung and Y. Choe, "Resource-aware and quality-fair video-streaming using multiple adaptive TCP connections," *Computers and Electrical Engineering*, vol. 35, no. 4, pp. 702-717, July, 2010. Article (CrossRef Link).
[11] X. Zhu, R. Pan, N. Dukkipati, V. Subramanian and F. Bonomi, "Layered internet video engineering (LIVE): network-assisted bandwidth sharing and transient loss protection for scalable video streaming," in *Proc. of the IEEE INFOCOM*, pp. 1-5, March, 2010. Article (CrossRef Link).
[12] X. Zhu, R. Pan, N. Dukkipati, V. Subramanian and F. Bonomi, "Layered internet video adaptation (LIVA): network-assisted bandwidth sharing and transient loss protection for video streaming," *IEEE Transactions on Multimedia*, vol.13, no.4, pp. 720-732, August, 2011. Article (CrossRef Link).
[13] W. Li, F. Yang and G. Ren, "High-speed rate estimation based on parallel processing for H.264/AVC CABAC encoder," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 237-243, February, 2013. Article (CrossRef Link).
[14] K. Stuhlmuller, N. Farber, M. Link and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012–1032, June, 2000. Article (CrossRef Link).
[15] S. Floyd, M. Handley, J. Padhye and J. Widmer, "TCP friendly rate control (TFRC): protocol specification," *IETF RFC5348*, September, 2008.
[16] C. Fu and S. Liew, "TCP veno: TCP enhancement for transmission over wireless access networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 215-228, February, 2003. Article (CrossRef Link).
[17] R. Jain, *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*, Wiley, New York, 1991. Article (CrossRef Link).
[18] J. Gross, J. Klaue, H. Karl and A. Wolisz, "Cross-layer optimization of OFDM transmission systems for MPEG-4 video streaming," *Computer Communications*, vol. 27, no.11, pp. 1044-1055, July, 2004. Article (CrossRef Link).
[19] J. Wu, Y. Shang, J. Huang, X. Zhang, B. Cheng and J. Chen "Joint source-channel coding and optimization for mobile video streaming in heterogeneous wireless networks,'' *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, Issue 1, pp. 1-16, December, 2013. Article (CrossRef Link).
[20] X. Zhu, P. Agrawal, J. Singh, T. Alpcan and B. Girod, "Distributed rate allocation policies for multihomed video streaming over heterogeneous access networks," *IEEE Transactions on Multimedia*, vol. 11, no.4, pp. 752-764, June, 2009. Article (CrossRef Link).

Dongchil Kim received the B.S. degree and M.S. degree from Kwangwoon University, Seoul, Korea, from the Communications Engineering Department, in 2009 and 2011 respectively. Currently he is a Ph.D. candidate in the Communications Engineering Department, Kwangwoon University. His research interests include multimedia, congestion control, smart TV, multipath TCP, and Internet QoS.

Kwangsue Chung received his B.S. degree from Hanyang University, Seoul, Korea, M.S. degree from KAIST (Korea Advanced Institute of Science and Technology), Seoul, Korea, and Ph.D. degree from the University of Florida, Gainesville, Florida, USA, all from the Electrical Engineering Departments. Before joining Kwangwoon University in 1993, he spent 10 years with ETRI (Electronics and Telecommunications Research Institute) as a research staff. He was also an adjunct professor of KAIST from 1991 to 1992, and a visiting scholar at the University of California, Irvine from 2003 to 2004. His research interests include communication protocols and networks, QoS mechanism, and video streaming.