Employing In-packet Segmentation with Wireless ARQ Protocols to Improve the Quality of TFRC Video Streaming

Chi-Huang Shih, Jun-Nan Guo, Ce-Kuen Shieh, *Wen-Shyang Hwang
Department of Electrical Engineering,
National Cheng Kung University, Taiwan
*Department of Electrical Engineering,
National Kaohsiung University of Applied Sciences, Taiwan
{chshih, waeric, shieh}@hpds.ee.ncku.edu.tw
*wshwang@mail.ee.kuas.edu.tw

Abstract

While transmitting real-time video stream over wireless networks, the packet losses due to congestion losses and wireless errors cause the degradation of video quality. In order to reduce the network congestion while maintaining the smooth sending rate, TCP-friendly rate control (TFRC) is designed for multimedia services. From previous studies, it is shown that the packet length determines the tradeoff between robustness and efficiency for wireless video transmissions. The smaller the packet size, the lower the packet corruption rate. However, the small-packet transmission induces unfairness problem occurs when small-packet TFRC flows compete the available bandwidth with large-packet TCP flows. In this paper, we propose an In-packet segmentation scheme to solve this problem. The results reveal that our proposed scheme can enhance video quality and improve the throughput fairness of TFRC while using variable packet size.

1. Introduction

In recent years, wireless networks provide the access conveniences to users, compared with the traditional wired/fixed networks. Several wireless network protocols are widely deployed in the current network infrastructure, such as IEEE 802.11a/b/g wireless local area networks (WLAN) [1]. In addition, with the rapid grows of wireless device, wireless network become more and more important in our life. Transporting real-time video stream is one of the popular applications in wireless networks, such as digital home, Internet television, video conference, and mobile learning. Accordingly, the demands of Quality of Service (QoS) concerned in terms of loss, delay and bandwidth have been increased [2]. The general transport protocol, UDP, does not support congestion control schemes and this may cause serious congestion loss as well as delays. Therefore, for congestion loss, an unreliable and equation based congestion control mechanism named TCP-friendly rate control (TFRC) is designed for delay-sensitive multimedia transmissions [6].

In an error-prone network, using varied packet size can reduce packet corruption rate and enhance goodput [3]. TFRC achieves the fair bandwidth sharing with TCP only when using the same packet size as TCP in competing the available network bandwidth. The throughput of TFRC is a linear function of transport packet size [4]. A flow using the small packet size to transmit data packets achieves a fraction of the throughput of a flow that uses large packet size. This causes throughput unfairness when the small packet size is used in TFRC transmissions [5]. To achieve throughput fairness, one can still use large packet size as the parameter to calculate the throughput in the TFRC equation while actually transmitting packets with small packet size [4]. Unfortunately, this introduces a throughput bias since the loss interval between two loss events is over-estimated to cause the under-estimated loss event rate and therefore, the sending rate is higher than that of the flow using large packet size for network transmission [4]. In [4], three different schemes, “virtual packet”, “random sampling” and “LIP scaling” had been proposed to remove the TFRC throughput bias. The main idea of these schemes is to modify the loss measurement of TFRC and make the loss event rate of the small packet size close to the one of the large packet size. The authors comment that the “virtual packet” scheme outperforms the other two schemes. However, the throughput fairness of “virtual packet” scheme only works well when the routers along the packet traverse path employ the random early detection (RED) queues and does not work well when the routers use drop-tail queues.

In this paper, we propose an In-packet segmentation scheme combined with wireless ARQ protocols to
improve the video quality and throughput fairness of TFRC while using variable packet size. The brief description of our proposed scheme is described as follows. The packet is virtually segmented into several parts and transmitted into the wireless Internet. By repeatedly utilizing the packets re-transmitted by the link-layer ARQ, the segment loss recovery is performed at the receiver side for the damaged packet to replace the erroneous segments with the correctly received segments. After the segment loss recovery, the original packet is either completely recovered or partially corrupted with reduced errors in it.

This paper is outlined as follows. Section 2 is our proposed scheme. In Section 3, some performance results are shown. Section 4 gives some conclusions.

2. In-packet segmentation scheme

We use the lightweight user datagram protocol (UDP-Lite) to send damaged packet to upper layer [7]. This protocol is designed to deliver the partial damaged packets to upper applications [8, 9]. The major difference between UDP and UDP-Lite is the way that they adopt to handle the corrupted packets. In UDP, the packet containing errors in it is discarded. On the contrary, UDP-Lite can detect the errors by using a partial checksum technique and ignore the errors to forward the corrupted packet. Therefore, the video applications that have error resilience codec can utilize the data within some bit errors to enhance the perceived quality [10]. To make UDP-Lite work correctly, it is desired to modify the MAC layer to forward error packets to upper layer and this modified link layer refers to MAC-Lite. In this thesis, an architecture that is based on TFRC/UDP-Lite/IP/MAC-Lite is considered for video transmissions in the wireless network. Figure 1 shows the architecture of the wireless video transmission system. This system uses UDP-Lite as the transport layer protocol. At the sender side, the multimedia packets are sent from the application and processed by our proposed scheme. Then, the packets pass through UDP-Lite layer, IP layer and link layer and are transmitted to the receiver side via wired/wireless channels. Because of the function of MAC-Lite and UDP-Lite, the packets that have partial damaged payload can be received by our proposed scheme at the receiver side and passes to the application after the processing of the proposed in-packet segmentation scheme. Besides, the receiver side measures the network information that is required by the sender and feeds it back periodically.

In TFRC, the throughput of the TFRC flow is reasonably fair when TFRC uses the same packet size as TCP to compete the available network bandwidth. Therefore, at the sender side, the virtual segmentation scheme virtually divides the packet into segments instead of actually packetizing the packet into several small ones for network transmission over lossy channels. Then, a virtual segmentation (VS) header is added to indicate the number of segments in the packet and verify the contents of each segment. At the receiver side, the VS header can be used to locate the position of each segment in the packet, check the data corruption of each segment, and replace the corrupted segments with the correct ones of the retransmitted packet. Because the packet is virtually divided by adding a VS header, the packet size is almost the same as the original one. The throughput fairness of TFRC can be achieved.

As shown in Figure 2, the VS header contains two parts. The first part is the eight-bit ratio field. This field is used to indicate the number of segments within the packet. The value of the ratio field is the ratio of the packet length to the segment size. Another part is “checksum n” field which is used to verify the content of the corresponding segment. For example, the “checksum 2” field is used to verify the content of “segment 2”.

Figure 1. Wireless video transmission system.
We decide the segment size that maximizes the throughput efficiency as follows:

\[ E = \left( \frac{k}{k + h} \right) \times (1 - P_B) \]  

(1)

where \( k \) is the length of data field of the segment in bits, and \( h \) is the length of “checksum \( n \)” field in bits, and \( P_B \) is the segment corruption rate in wireless channel.

3. Performance results

We use the following network topology as shown in the Figure 3 to evaluate the performance of our proposed scheme. All of the devices are connected with wired links except the data receivers. The senders include two linux machines: one of them is a multimedia sender on which our proposed scheme is implemented, the other one is the TCP background traffic generator. At the receiver sides, TFRC receiver receives the packets from TFRC sender. TCP flows are produced as the background traffic to compete the available bandwidth with the TFRC flow. Along the packet traverse path, the intermediate node uses the NISTnet tool to generate the packet delay. The CISCO router limits the bandwidth in the wireless link.

Table 1 Experimental environment settings.

<table>
<thead>
<tr>
<th>Network settings</th>
<th>Traffic settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue discipline</td>
<td>Drop-tail queue</td>
</tr>
<tr>
<td>Bandwidth (wireless)</td>
<td>1Mb/s</td>
</tr>
<tr>
<td>End-to-end Delay</td>
<td>10 (ms)</td>
</tr>
<tr>
<td>Video sequence</td>
<td>Foreman</td>
</tr>
<tr>
<td>Resolution</td>
<td>CIF (352 x 288)</td>
</tr>
<tr>
<td>GOP length</td>
<td>9 frames</td>
</tr>
<tr>
<td>Packet size of TCP</td>
<td>1000 (bytes)</td>
</tr>
</tbody>
</table>

We estimate the fairness by using the normalized throughput equation as Eq. (2) where the “small pktsize” refers to the packet size which is smaller than 1000 bytes.

\[ \text{Normalized throughput} = \frac{\text{Throughput of TFRC}_{\text{small}}}{\text{Throughput of TFRC}_{\text{max}}} \]  

(2)

Figure 4 shows the normalized throughput comparison as static small packet size varies from 200 bytes, 400 bytes, 600 bytes, 800 bytes, to 1000 bytes. The line denoted as “No-TFRC” means that we actually use the small packets to transmit data based on TFRC; the line denoted as “VP-TFRC” is the same meaning as the line “NO-TFRC” except that the virtual packet scheme is used. From Figure 4, the throughput of TFRC should be reasonable fair while TFRC uses the same packet size as TCP to compete bandwidth. Because our proposed scheme virtually segments the packet and the packet size is close to the one of TCP flow, our proposed scheme outperforms the other two methods in the fairness comparison. From the results, the normalized
The throughput of our proposed scheme is close to 1 for every static packet size. Figure 5 shows the normalized throughput of VP-TFRC, NO-TFRC and the proposed scheme for different numbers of TCP flows. In Figure 5, both the VP-TFRC and NO-TFRC scheme determine the packet size by using Eq. (1) to maximize the throughput.
efficiency. In calculating Eq. (1), the parameters \(k\), \(h\), and \(P_P\) refer to the length of data field of the packet in bits, the length of header of the packet in bits, and the packet corruption rate, respectively. From Figure 4 and Figure 5, the smaller the packet size and the more number of the TCP flows can cause the throughput of original unmodified TFRC more unfair. From above experimental results, our proposed scheme achieves good throughput fairness. Figure 6 shows the PSNR for varied MAC retry limit. The line denoted as “Original” represents the results without packet size control. The line denoted as “Adaptive packet size” represents the results that the packet size is determined by using Eq. (1). From Figure 6, our proposed scheme achieves high PSNR values by utilizing the retransmitted packets.

4. Conclusions

This paper proposes an In-packet segmentation scheme to avoid the throughput unfairness induced by small-packet TFRC. The experiment results show that the proposed scheme achieves a significant improvement on video quality and throughput fairness.

References


