# RTCP-Based Frame-Synchronized Feedback Control for IP-Video Communications over Multipath Fading Channels

H. Gharavi, K. Ban, and J. Cambiotis
National Institute of Standards and Technology
100 Bureau Drive, Gaithersburg, MD 20899-8920

*Abstract*— **This paper presents a packet-loss feedback tracking scheme for the transmission of video signals over mobile channels. The proposed feedback scheme is based on the real time transport control protocol (RTCP), which is designed to provide an end-to-end feedback assessment of transmitted packets on a frame-by-frame basis (video frame). In addition, the frame synchronized RTCP-based feedback scheme is designed to take care of losses of RTCP packets due to bad channels. The video encoder, upon receiving its feedback report, can identify the exact location of the missing packets in the transmitted video frame. The feedback scheme is then applied to transport H.264/RTP/UDP/IP packets in real-time. A packet-loss compensation strategy has been used to assess the quality of the received signal under multipath fading channel conditions.**

***Keywords-video streaming; RTP; RTCP; feedback control; H.264; IEEE 802.11; wireless; multimedia***

## I. INTRODUCTION

In recent years, there has been a tremendous interest in supporting IP-based services for mobile communication systems [1], [2]. Protection of video over mobile channels using FEC coding, automatic repeat request (ARQ), or a combination of both (hybrid) is normally supported by most recent wireless transmission standards. For instance, the IEEE 802.11 standard [3] provides ARQ (retransmissions) at its MAC layer, which may not always be suitable for delay-sensitive video conferencing applications. While packet protection schemes are very effective in reducing packet drops, feedback channel assessment can be utilized to improve the quality of video against transmission error by means of error resilient coding [4].

For RTP-based video streaming, a receiver feedback is readily provided by the RTP control protocol known as RTCP [5]. The RTCP, via its reception report, can provide end-to-end feedback information about delay jitter and packet-loss performance. Upon receiving such a report, the transmitting node may be able to adjust its coding strategy in order to reduce the effect of a missing packet at the receiver. Due to the round trip delay and the time interval in which the reception report is generated, the transmitter may not be able to respond to the channel variation in a timely manner. However, a more periodic feedback channel assessment would be at the expense of substantially increasing the channel traffic. In addition, for RTP streaming [5], [7] no guidelines have yet been specified that are specially defined for tracking packet losses for video communications.

Therefore, in this paper we present an RTCP-based scheme capable of reporting the channel performance on a synchronized frame-by-frame basis. The main feature of the proposed RTCP-based feedback scheme is its ability to identify the exact location of the missing data in a transmitted video frame. Based on such information, the encoder can change its coding strategy to compensate for the effect of missing data and thus avoiding distortion propagation. The proposed packet-loss tracking feedback scheme is designed to take into consideration the possibility of losing RTCP packets that are transmitted via the feedback back channel.

In the following, we first present our modified RTCP packet structure, which has been implemented for real time transmission of RTP/UDP/IP video packets using the H.264/AVC coding standard [6]. We then discuss our frame-synchronized packet-loss tracking scheme for the transmission of the H.264 coded video. In particular, we have considered a locally corrected reference frame to overcome the effect of the distortion propagation. For these experiment we have used our high-speed simulation testbed based on 802.11 WLAN [2] standard.

## II. FRAME-SYNCHRONIZED RTCP

RFC1889 presents the fundamental characteristics of RTP and RTCP [5]. The RTP protocol is used for transmitting real-time data information and RTCP for sending control information. The main function of RTCP is to provide a detailed representation of the packets exchanged during an RTP session. Its structure includes sender report (SR) and receiver report (RR), which are periodically transmitted to all participants in the session. It aims at providing a feedback on the quality of the transmission (e.g., delay jitter, average packet loss, etc.), where transmitters send "sender reports" and receivers send "receiver reports". While the SR includes transmission and reception information for active senders in the session, the RR would also contain reception information for non-active senders.

Although RTCP packets can be very effective in monitoring packet-loss statistics and calculating delay jitter of the nodes in an RTP session, they may not be suitable for low delay packet-loss control applications. This is mainly due to the

DARPA NMS Grant 8922385 supports this project.

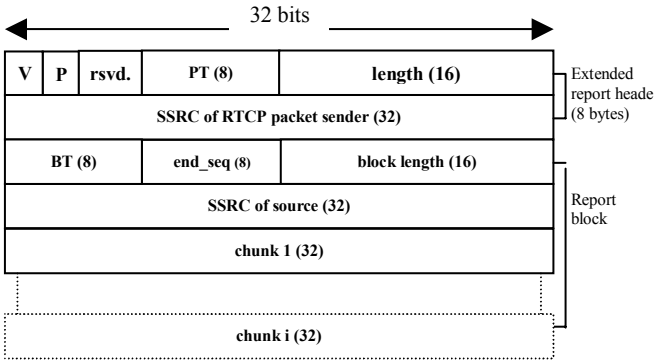


Figure 1. Modified XR packet format.

fact that for packet control and recovery applications, it would be essential to receive feedback assessment as quickly as possible. Since the current RTCP packet size is large, more frequent RTCP reports can significantly undermine the bandwidth availability.

We have developed an RTCP-based packet structure aimed at providing end-to-end feedback that can report packet losses in a timely manner. Our packet structure is similar to the RTCP extended report (RTCP-XR), which is primarily defined to provide more detailed statistics, particularly for multicast applications [7]. In our case, the RTCP-XR format was specifically designed to report losses on every frame of video signal.

### *A. Extended Report Packets*

The report block defined in [7] (Loss RLE Report Block) can be used for network monitoring or for applications that can make use of massive raw data. In our case, we need a compact XR packet that can report a relatively small number of packets within a short interval. This allows the encoder to set its coding strategy in a constructive manner and prevent distortion propagation. Thus, we have considered the same framework as in [7], but define our own XR block, which is similar in many respects to the "Loss RLE Report Block" in [7].

As will be discussed later, the information included in the proposed report block relies on the encoder's ability to track the transmitted packets by maintaining a transmitter table to identify the position of pixels in a corrupted video frame at the receiver.

Although the same XR packet type defined in [7] can also be used for our application, we slightly modified the packet format to reduce the overhead by 32 bits. Fig. 1 shows the modified XR packet format. The header remains the same as the XR packet defined in [7] where V (2 bits): identifies the RTP version, P (1 bit): is the padding bit and is set when the XR packet contains an additional padding octet at the end, resvd.: reserved (5 bits), PT (8 bits): packet type to identify this as an RTCP XR packet, length (16-bit): the length of the XR packet (in 32-bit word minus 1) including the header and any additional padding, SSRC (32-bit): the synchronized source identifier for the originator of this XR packet. Following the XR header is the report block, which begins with BT (block type) field that defines one of the seven report block options defined in [7].

The main difference between our XR and the one in [7] is the format of the remaining fields in the report block. For instance, we have used only the end sequence number, as the begin sequence number is not needed if the fixed length bit pattern is used (see the next paragraph). In addition, instead of the 16-bit RTP sequence number, only its 8 least significant bits are used. The next field is the block length, which specifies the length of the report for the bit pattern (including the header) in 32-bit words minus one.

The bit pattern, which presents the status of delivered/undelivered packets (i.e., 1/0), is included in the report and transported as a stack of 32-bit chunks (see Fig. 1). To exploit the inter-dependencies between the delivered and undelivered (e.g., bursts of packet drops) packets, the bit pattern can be runlength encoded. However, in our current implementation we have used uncoded fixed 32-bit chunks to transmit the bit pattern. In addition, as we will discuss next, we have developed an overlapping bit pattern reported by consecutive RTCP packets to allow the encoder to locate the missing data if the previously transmitted XR packets do not reach the transmitter.

## III. PACKET CONCEALMENT CONTROL

The source node, upon receiving the XR report, first identifies the position of the unsuccessfully transmitted packet within a coded frame by constructing a transmitter table. Note that the design of the transmitter table depends on the underlying compression technique as well as the encapsulation strategy. As we have considered the H.264/AVC coding standard [6], let's first provide a brief overview of the H.264 standard.

Before RTP streaming, the H.264 compressed video signal is first encapsulated into an appropriate packet size by taking into consideration the nature of the transmission system. H.264 was designed to provide a "network-friendly" packet-based video representation. It is based on the conceptual separation between a video coding layer (VCL) and a so-called network abstraction layer (NAL) [6]. Following the NAL header, the RTP payload is comprised of the slice output of the VCL and its header includes parameter set, picture structure (progressive frame picture, top field picture, bottom field picture, etc), slice type (Intra, Inter, B, etc.), address of the first macroblock (MB) in the slice, and so on. The first macroblock address provides useful information to find the number of macroblocks (MBs) that may have been lost in the preceding packets.

As an example, the operation of packet loss identification via the RTCP-XR feedback is described for a point-to-point transmission between node-A (the sender) and Node-B (the receiver). This example is depicted in Fig. 2 for the H.264 coding scheme. At Node-A, the H.264 encodes video frame at the slice-level with the option of a near-uniform packet size.

The coded slice is then encapsulated into an RTP packet with the NAL header [6]. Prior to the packet's transmission, specific information is stored at the transmitter table with

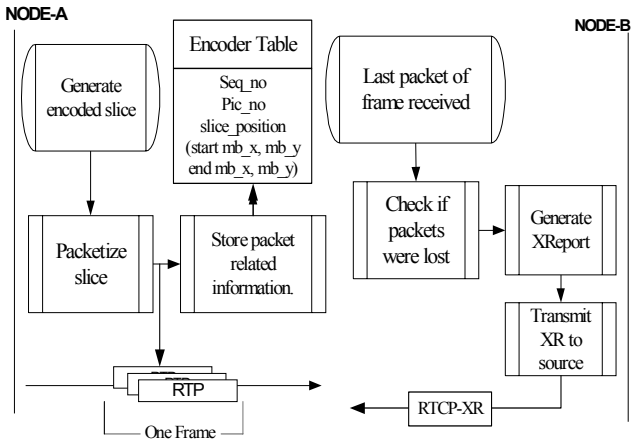


Figure 2. Video frame synchronized RTCP-based feedback for transmission of RTP packets from Node-A to Node-B.

respect to the RTP packet sequence number (see encoder table in Fig. 2). This information consists of: a) the picture number of the packetized slice, and b) the position of the slice within that picture (i.e., the first MB and the last MB of the slice).

The stored information in the transmitter table, with respect to the RTP sequence numbers reported by the XR, is also shown in Fig. 2.

At Node-B, the RTCP-XR packet is generated as soon as all the successfully received packets belonging to the same frame are detected. The receiver then generates its reception report (report block) at the end of each frame. This includes a bit pattern, which is packaged into one or more 32-bit chunks. Note that the criterion by which the receiver can detect the end of the frame is by checking both the M flag in the RTP header and the RTP timestamp.

For example, if the last packet in the frame (M flag packet) has also been lost, the receiver, via a change of RTP time-stamp, can easily detect that the newly arrived packet belongs to the next frame. We should also point out that under hostile channel environments, loss of packets may extend beyond a single frame. In this situation and with the aid of the stored sequence number at the transmitter table, the encoder can still identify the missing frames (including their corresponding packets). Fig. 3 shows an example of how the RTCP-XR report is generated. In this example it is assumed that all the packets are received in the correct sequential order.

As can be deduced from this example, the overlapping bit pattern in the XR report partially includes the bit pattern of the previously transmitted XR packet as long as it remains within its predefined one or multiple 32-bit chunk structure (e.g., 32, 64, 96, ..). With this arrangement, the source node receiving the XR report can still identify the missing packets that have been reported by the pervious RTCP-XR. This arrangement can be very effective when the back channel also suffers losses due to multipath fading.

It should be also noted that, depending on the status of the lost packets in the coded frame, the RTCP-XR might not be

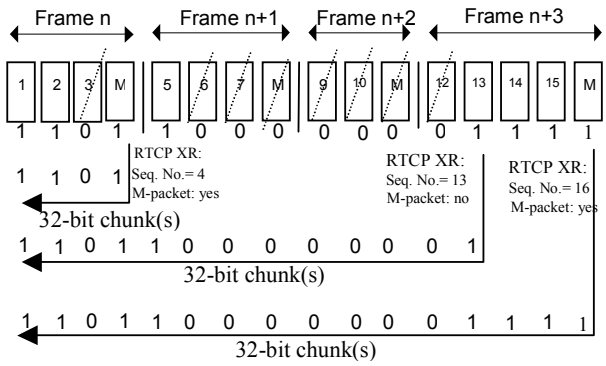


Figure 3. Identification of delivered/undelivered packets and the generation of the bit pattern, M: M flag packet.

able to report packet losses exactly at the end of each frame (i.e., fixed time interval). This is due to the fact that in the absence of the last packet (with the M flag), the XR is generated after the next successfully received packet, which belongs to the next frame.

### *A. Error Concealment*

After identifying those packets that have not reached their destination, there are a number of methods that can be considered to conceal their effects on the quality of the received video signal. The most straightforward option however, would be to retransmit the missing packets. This option introduces additional delay, which may not be acceptable for two-way video communications.

In this paper we implemented a simple, low-delay packet-loss concealment scheme that can prevent distortion propagation. Fig. 4 shows a simple scenario of the error concealment process where packets have been lost during transmission of the first frames (i.e., frame: $n$). In this example we have assumed the XR packet reaches the transmitting node before encoding the frame $n + 2$. According to this figure, as soon as the transmitter receives the XR packet, the encoder via its encoder table (see Fig. 2) can identify the frame and its specific regions, which had been affected by the missing packets. Having identified the frame, the encoder first replaces the pixel values covering these regions (from previously decoded frame) in the locally decode frame (i.e., frame $n$ in Fig. 4). This modified frame is then used as a new reference frame to locally reconstruct the next frame (frame $n$+1 in Fig. 4). This process continues until reaching the frame $n$+2, which has not yet been encoded. Assuming this frame reaches its destination without any loss of packets, the local decoder at the transmitter and the remote decoder at the receiver are now in full coordination and thus, the distortion can no longer effect this frame. It should be noted that the main objective behind this arrangement is to allow the distortion to propagate through the locally reconstructed frames (reference frames) in the same way as in the remote decoder. In this example, since the XR packet does not reach the transmitting node before the frame $n$+1, this frame could not be rescued from the distortion propagation.

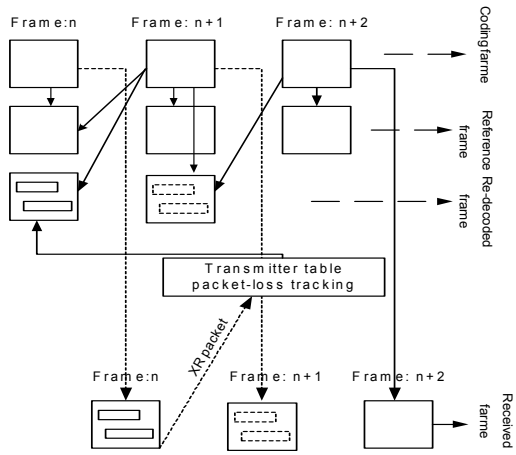


Figure 4. An example of distortion propagation and error compensation by correcting the reference frame at the encoder

Therefore, the number of infected frames depends entirely on the time interval in which the transmitter can receive its feedback report. However, under the proposed frame synchronized feedback scheme, there are a number of other factors that may cause delay in generating or receiving the XR report. The first one is a loss of a large number of packets that may cover the entire transmitted frame (s). The other factor is a loss of RTCP-XR packet itself. Fortunately, with the bit-pattern overlapping strategy, by receiving the next XR packet, the encoder should be able to identify the status of the transmitted packets reported in the pervious XR packets. Nevertheless, a loss of XR packet would prolong recovery.

## IV. EXPERIMENTAL RESULTS

A high-speed simulator to evaluate the performance of the feedback control scheme and its effect on the quality of video in real-time had been used. In these experiments the IEEE 802.11 WLAN technology was considered for a point-to-point communication with the bandwidth of 2 Mb/s and without a retransmission. The receiver buffer size was set at 5000-byte. The Ricean channel model using differing fading factors (K = 0 - 20) was considered. For the average channel SNR of 27 dB, the effect of feedback channel on the quality of the received video was then evaluated. For RTP/UDP/IP streaming, H.264 video coding has been used. In our experiments, XR packets were transmitted at the end of each frame. We should point out that for jitter control, standard RTCP packets were also transmitted at the 5 sec. nominal interval [5].

To carry out these tests under the same conditions, a pre-captured video sequence with the QCIF format at 10 frames/s with the fixed quantization parameter (i.e., QP= 11) was used instead of live video. In addition, the compressed video was encapsulated at a near fixed packet size of 500-byte.

In the following figures, we present the results of the feedback-tracking scheme (with the local decoder correction). For the purpose of comparisons these figures also included the results without the feedback control. Fig. 5 shows the average

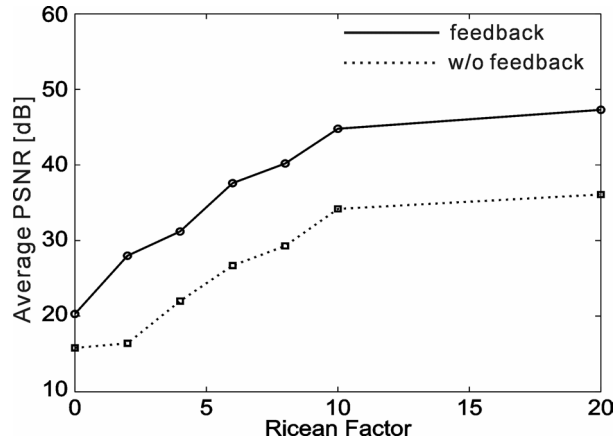


Figure 5. Average PSNR performance with and without feedback tracking.

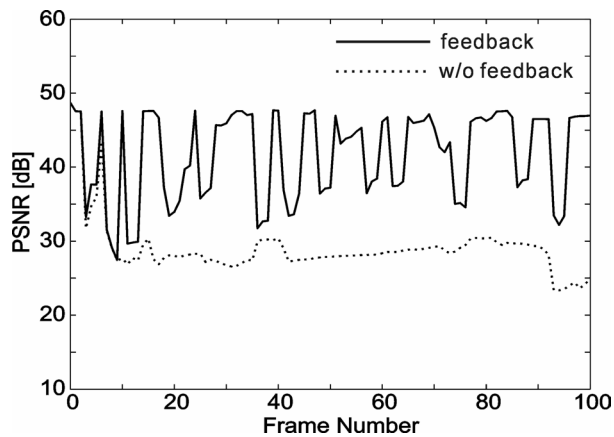


Figure 6. PSNR with and without XR report (Ricean factor K=8).

PSNR (peak-to-peak signal to r.m.s noise ratio) improvement when using the feedback-tracking scheme under various fading conditions.

Thanks to the RTCP-XR feedback report, substantial improvement can be achieved (more than 10dB) without even having to retransmit the missing packets. The contributions of the frame synchronized RTCP-XR packet-loss tracking scheme are shown in Fig. 6 and Fig. 7 with Ricean factors at K = 8 and K = 20, respectively.

In these figures, a dip in the PSNR values (with the local correction) is the result of a delay required for the XR report to reach the source node. The number of frames affected by the distortion propagation (a dip in PSNR values) also depends on the loss of XR packets due to a bad channel. Although the overlapping bit pattern strategy can take care of this situation, it can result in more uncorrected frames reaching the receiver.

Evaluating the effect of bit pattern length on the video recovery, Fig. 8 shows the comparison between the 32-bit (one chunk) and the 64-bit (two chunks) bit-pattern with the Ricean fading factor of 10. The improvement of the 64-bit chunks over

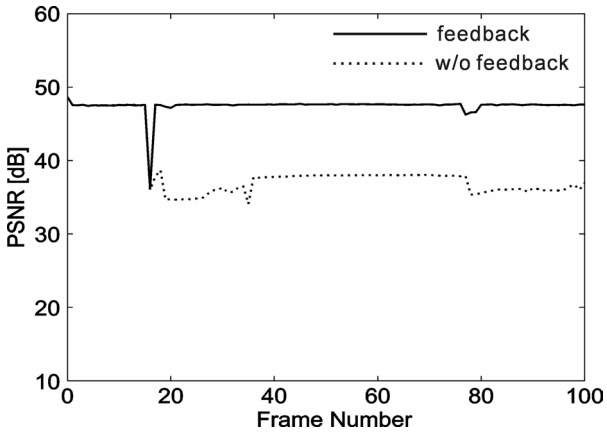


Figure 7. PSNR with and without XR report (Ricean factor K= 20).

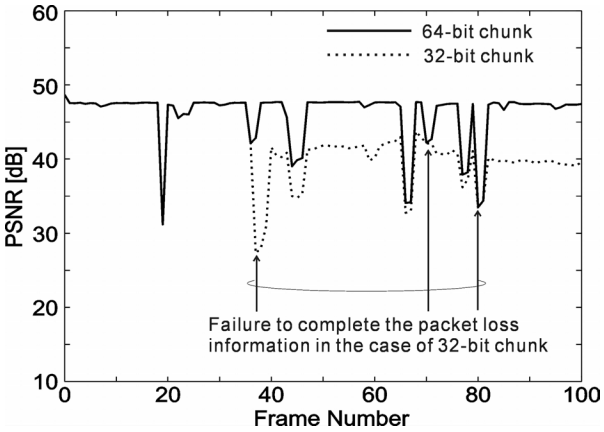


Figure 8. Effect of the bit-pattern chunk size on the feedback tracking performance (Ricean factor K= 10).

the 32-bit chunk depends on the XR-packet loss rate and the extent to which the bit pattern overlap each other. In the former case, we observed that under less severe fading conditions (i.e., higher values for K), the performance difference between the 32-bit and 64-bit bit-patterns becomes negotiable. With respect to the extent of their overlapping effect, this largely depends on the number of packets per consecutive frame, which tends to vary due to motion activities. Thus, the size of video packet (i.e., fixed length packet), the video coding rate, and the fading channel conditions are important factors in selecting a suitable length of the bit pattern.

Finally, we should point out that the main attribute of the proposed packet-loss tracking scheme is its synchronization with the RTP sampling frequency, as the number of packets per frame may change from frame to frame. This feature can regulate the transmission of XR packets and indeed, reduce the number of distorted frames before recovery.

## V. CONCLUSION

Our main objective was to develop a frame synchronized RTCP-XR feedback control scheme to improve the quality of the received video under multipath fading conditions. For transmission of RTP/UDP/IP packets we developed an RTCP-based feedback control scheme, which reports the status of the received packet on a frame-by-frame basis. In addition, by constructing an encoder table, the transmitter can identify the exact location of the undelivered data on the previously transmitted frames. We have shown that, based on the feedback information, the quality of the received video can be significantly improved.

## REFERENCES

[1] TR 25.933, "IP Transport in UTRAN Work Task Technical Report," 3GPP, May 2001.

[2] MTR-006, "IP in the RAN as a transport option in 3rd generation mobile systems," MWIF Technical Report, June 2001.

[3] ANSI/IEEE Std 802.11 1999 Edition, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Institute of Electrical and Electronic Engineers, August 1999.

[4] B. Girod and N. Farber, "Feedback-Based Error Control for Mobile Video Transmission," THE PROCEEDINGS OF THE IEEE, vol. 91, NO. 8, August 2003, pp. 1221-1234.

[5] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications, " Internet Engineering Task Force, RFC 1889, January 1996.

[6] S. Wenger and T. Stockhammer, "An Overview of the H.26L NAL Concept," Document JVT-B028, JVT 2nd meeting, January/February 2002.

[7] T. Friedman, R. Caceres, and A. Clark, "RTP Control Protocol Extended Report (RTCP-XR)," Internet Engineering Task Force, RFC 3611, November 2003.