Dynamic Adjustment Packet Control for Video Communications over Ad-hoc Networks

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

***Abstract*—This paper is concerned with transporting video information via multihop mobile ad-hoc channels. The major problem with transmitting real-time video information over these channels is the issue of link reliability. To improve the quality of the video reception we propose a cross layer feedback control mechanism that can allow the application layer to adapt itself to a dynamically changing network topology. We also present packet transmission strategies capable of recovering video signals under long bursts of packet drops, typical of a route change condition. This feedback control scheme has been developed for transmission of RTP/UDP/IP packets using the emerging H.264/AVC video-coding standard.**

Keywords-AODV; multihop; multimedia; video streaming; RTP; feedback control; H.264; IEEE 802.11; wireless

I. INTRODUCTION

As real-time multimedia communications find their way into military and commercial applications, efforts to support QoS under ad-hoc network environments are becoming increasingly important [1,2,3,4]. In particular, under best effort ad-hoc network conditions where routing and channel characteristics are expected to vary dynamically (e.g., high mobility conditions), achieving an acceptable video quality would be a challenging task. For instance, assuming a point-to-point communication, major factors affecting the ad-hoc channel performance are large delays due to the route discovery process in the event of a route change, as well as deterioration of the network throughput performance as the number of transmission hops increase [5]. This behavior is shown in Fig. 1, which depicts the maximum throughput performance as a function of hop-count (number of hops from the source to the destination). These results were obtained using our real-time simulator that has been developed for a peer-to-peer multihop ad-hoc network based on the IEEE 802.11b WLAN standard [6]. The maximum bitrate of all IEEE 802.11b WLAN nodes was set to 2 Mbit/s with a buffer size of 2000 bytes for every node. In these experiments, the Ricean multipath fading model using differing values for its parameter, K (Note that K = 0 corresponds to the Rayleigh fading), at a velocity of 10 Miles/hour has been used. We used a prerecorded sequence with slice size 600-byte. The AODV routing protocol [7] has been utilized for ad-hoc routing. These results indicate that the network's maximum throughput performance can suffer as the signal traverses over more hops. This is mainly the result of co-channel interference and an additional IP packet delay,

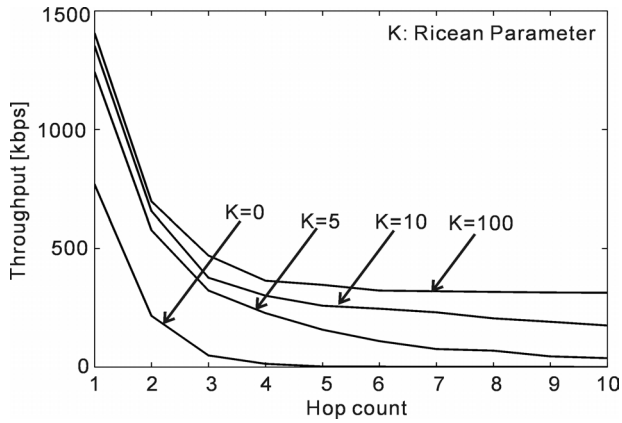

Figure 1. Hop count versus maximum throughput performance (UDP packet size is 612-byte).

which tends to increase as the signal hops through more intermediate nodes (e.g., sharing the same channel).

Therefore, to solve this problem, it would be essential to develop a feedback mechanism that can dynamically control the source bit rate in accordance with the characteristics of the multihop channel. Furthermore, having knowledge about a link failure can also help the application layer to control its packet transmission strategy accordingly.

In this paper we have proposed a cross-layer feedback approach to control the packet transmission policy capable of avoiding a substantial loss in video quality. In addition, to control the video transmission rate and compensate for the large number of packet drops, we present a number of coding strategies to improve the subjective quality of the received video in real-time. This includes a redundant packet transmission scheme to enhance packet recovery as well as reducing the discontinuity effect in the video reception (e.g., freeze-frame), in the case of link failure.

II. FEEDBACK CONTROL

The results shown in Fig. 1 verify the degree to which the network performance deteriorates with increased numbers of hops. This behavior suggests that to improve the perceived quality of video, it would be necessary to develop a feedback control mechanism that can allow the application layer to adapt

DARPA NMS Grant 8922385 supports this project.

itself to changes in the network characteristics. Such a mechanism could be based on a form of cross-layer feedback from the network layer to the application layer. In particular, information about the transmission path, such as hop-count or an occurrence of link breakage, can provide valuable information to the source node about the transmission link as far as ad-hoc routing is concerned. For instance, in ad-hoc routing protocols such as AODV (Ad-hoc On Demand Distance Vector) [7], each node maintains a routing table for an entry (destination) with the hop-count (number of hops from source to destination) and sequence number. In addition, a link breakage can also be detected by looking at the routing table information and/or control messages of the routing protocol. If a route change is the consequence of a link breakage, any intermediate node (between the source and destination) detecting the link breakage (to the next hop), will send the route error (RERR) message back to the source node. The source node therefore may use the reception of RERR as an indication of a link breakage. As an alternative way to detect a link breakage, the source node can simply monitor the sequence number for the destination in the routing table. In fact, the sequence number is updated whenever a node receives new information about the sequence number from the RREQ, RREP, or RERR messages related to the destination.

The delay in which the link breakage information can reach the source node depends on how quickly the intermediate node can detect the link breakage. The AODV may use a hello message-option to detect a link breakage. Another option is via the MAC layer [7], which has been considered in this paper. For instance, as IEEE 802.11b relies on positive acknowledgment, if the transmitter does not receive any acknowledgement after *n* retransmission attempts, a link breakage will be triggered. In this case, a delay associated with detecting a link failure depends on the value of *n*, which defines the maximum number of retransmissions setting, the acknowledgment timeout parameter, and the length of the packet (not including the propagation delay).

Fig. 2 illustrates the process of route change initiated by a link breakage at an intermediate node along the active routing path. As soon as the intermediate node detects the link breakage it will send an RERR message to the source node. Upon receipt of the RERR, the source node will reinitiate a route discovery process to seek a new route to the destination with an incremented sequence number.

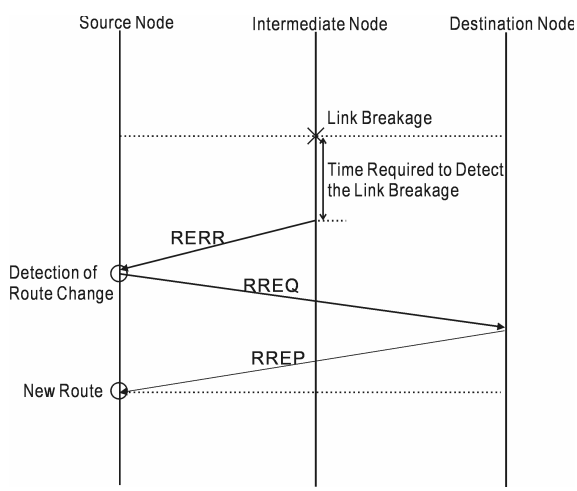

Figure 2. Process of route change

In the case of Linux operating system, the AODV protocol can share the routing table with other application programs through a virtual file in /proc directory or shared memory. This file system is used to pass the routing information to the application programs (e.g., video encoder). To keep a record of link breakages, the AODV protocol may create a new file that lists the destination together with the time of the link breakages. Every time the AODV protocol detects a link breakage (by receiving/initiating RREQ, RREP, or RERR), it writes the destination and time on the file so that other application programs can access and read the information on link breakage.

Each time the video encoder tries to send packets, it first checks its routing file system. If a route change is detected, the application layer upon receiving the hop-count information from the routing layer, should then adjust its bitrate in accordance with the permissible transmission rate. In the case of video communications, the bitrate can be adjusted by changing the value of the quantization parameter (QP). This parameter has been specifically defined in the syntax structure by all video coding standards as the means to control the video transmission rate. Its value, which can have a direct bearing on the video quality, is selected as a two-way compromise between the average transmission rate and the video quality. If a route change is the result of a link breakage, the video encoder terminates packet transmissions immediately after detecting a link breakage.

III. PACKET TRANSMISSION STRATEGIES

As soon as a new route is established, the application layer resumes its communication. Under this condition, we have considered three coding strategies for the resumption of packet transmission and encoding the incoming frame. Fig. 3 shows an example of the effect of a route change on consecutive coded video frames. The most straightforward option would be to switch to the I-frame mode (Intraframe) to encode the incoming frame as soon as the new route is established. Intraframe coding, despite its higher bitrate (compared with interframe prediction: P-frame), has the advantage of being independent of previously coded frames and can therefore expedite the re-synchronization of the video at the decoder in the absence of the previously dropped frames.

The second option would be to encode the incoming frame in a P-frame mode but using a frame that was encoded prior to the link breakage for prediction (i.e. frame n in Fig. 3). To make sure that the nearest reference frame has been selected for the prediction, it would be necessary to estimate the time interval between the actual occurrence of the link breakage and the formation of a new route. The main disadvantage of this approach however, is an unprecedented long interval, which may affect the interframe prediction accuracy. Consequently, this may

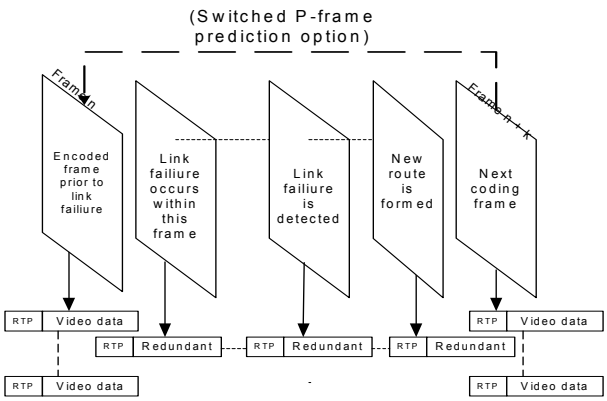

Figure 3. Effect of a route change on a consective coded video frame and packet transmission strategy.

cause an increase in the bitrate as most blocks are expected to be encoded in the intraframe mode. Nevertheless, selecting a large tracking range for motion estimation can improve the prediction performance given that the number of missing frames is limited and/or the motion activities remain mostly within the search window during a route change period.

At this stage, we should point out that both the above strategies have the advantage of not affecting the end-to-end delay, which is an essential factor for a two-way communication. However, assuming that the receiver can tolerate a longer delay, the third approach is based on retransmission of the undelivered packets as soon as the new route is established. Unfortunately, retransmitting a large number of packets can cause a sudden increase in the bitrate, consequently overloading the network. This arrangement can be particularly harmful when the new hop-count is higher than before the link breakage. Thus, we propose a data re-encapsulation technique aimed at reducing video data for retransmissions (at the expense of a graceful reduction of video quality). This scheme, which will be referred to as a redundant packet, was originally designed for lossy recovery of packets lost under fading conditions (i.e., without a link breakage).

A. Redundant Packet

In this scheme, which is based on the RTP streaming protocol [8], the entire video information in each frame, after being reduced in size, is re-encapsulated into a single RTP packet. This type of packet, which is generated in a per frame basis, is transmitted together with other video packets for the purpose of packet recovery under multipath fading conditions (e.g., a small burst of packet drops). In the event of a link breakage however, redundant packets can also be used to replace the missing frame (see Fig. 3).

We should point out that the motivation behind a redundant packet is to transport only the most sensitive information in a coded video frame. Its structure is based on removing the less sensitive transform coefficients in every coded block with the object of minimizing the propagation of distortion. Note that a propagation of distortion is mainly caused by a lack of coordination between the local decoder (at the transmitter) and the remote decoder (at the receiver).

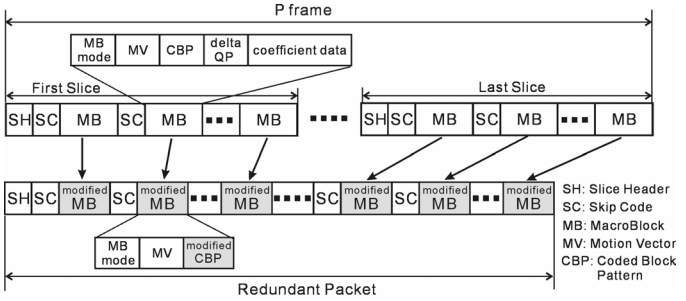

Figure 4. Format of the redundant packet

The specific format of a redundant packet depends on the underplaying compression technique as well as its encapsulation strategy. As we have considered the recent H.264/AVC coding standard [9] for RTP/UDP/IP video streaming, the procedure of generating the redundant packet will be presented after a brief overview of the H.264 video coding.

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 is 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) [10]. 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.

The design of a redundant packet is based on the fact that recovering only the motion vectors can have a profound effect on concealing the propagation distortion, despite a loss of data coefficients. In this strategy, a redundant packet is generated for every frame using the same header as the first slice. Fig. 4 shows the arrangement for generating a redundant packet for an entire P frame. As shown, its payload consists of runlength codes for skip macroblocks, the macroblock mode, components of the motion vector, and finally, the coded block pattern (CBP).

In the case where a macroblock contains coded coefficients, the value of the CBP is changed accordingly, signifying that the macroblock contains no data coefficients. In addition, if a frame contains any intra-coded macroblocks, their entire data will be included in the redundant packet. Note that in this arrangement intra-coded macroblocks, before being transferred into the redundant packets, are entropy-encoded without using neighboring inter-coded macroblocks. However, depending on the degree of a packet-drop's burstiness, the redundant packet, which is generated for every coded frame, can be transmitted a number of times per frame (interleaved within the data packets) by using the same RTP time-stamp. In addition, a redundant packet belonging to the previous frame may also be transported via the current frame to increase its span against a very long burst of packet drops. At the decoder a successfully received redundant packet is then utilized to regenerate the missing data

packet(s). This initially requires identifying the first and last macroblock addresses in the missing packet (i.e., slice), which can be easily obtained from the successfully received neighboring packets. Based on this information the relevant data is then extracted from the redundant packet to regenerate only the missing packet. Obviously, in this arrangement, as long as the redundant packet is received successfully, it is possible to repair any number of missing packets in the frame.

In the case of link breakage (see Fig. 3), these redundant packets, which are stored for every video frame, will be transmitted as soon as a feedback from the network layer signals the formation of a new routing path. Bear in mind that the number of frames that can be affected by a loss of large numbers of packets depends on how quickly the application layer receives a link-loss notification from the underplaying network layer. To ensure that an appropriate number of redundant packets (i.e., one packet per frame) will be transmitted to compensate for undelivered packets, it would be necessary to estimate the link loss time interval. As discussed in section 2, this depends on AODV parameters such as Retry-limit, ACK timeout, packets size, and hop-count from the node that has detected a link-loss. After completing the transmission of the redundant packet, the next incoming frame will be coded in a P-frame or I-frame mode but at a different bitrate in accordance with the new hop-count. A change of bitrate would require changing the quantization parameter (QP) in such a way that the target bitrate shown in Fig.1 can be met.

IV. EXPERIMENTAL TESTBED

An experimental testbed for a peer-to-peer mobile ad-hoc network was constructed to demonstrate the performance of video communication under a change of routing condition. In these experiments, a pre-recorded video sequence (QCIF format) has been encapsulated into 612-byte RTP packets (including the RTP header) before being transmitted in real-time over a multihop ad-hoc channel. All the mobile nodes were implemented using IEEE 802.11. For ad-hoc routing we deployed the AODV routing protocol. In our experiments, the bit rate for all the IEEE 802.11 devices was set to 2 Mb/s. In addition, the maximum number of retransmissions, *n*, on all WLAN devices was set to 2. Note that a higher number of retransmissions can also help avoid an unnecessary link breakage. For RTP/UDP/IP streaming, an H.264/AVC encoder was considered. One redundant packet per frame was included in every frame to help lossy recovery of the missing packets at the receiver. A simulation testbed was constructed by importing our H.264/RTP/UDP/IP streaming into the Qualnet. The testbed is capable of grabbing live video for transmission in real-time. This testbed was used to evaluate the cross-layer network performance using the Ricean channel model. To carry out our measurements under differing fading conditions, a pre-captured video sequence (QCIF) was used instead of live video.

To investigate the impact of the feedback control scheme on the network performance, our next experiment was based on a scenario where the destination node moves away from the source node. In this arrangement, transmitted packets can go through

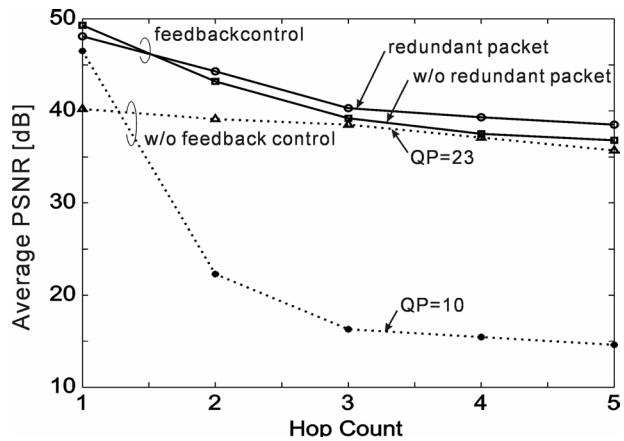

Figure 5. Hop count versus PSNR (Ricean factor K=10).

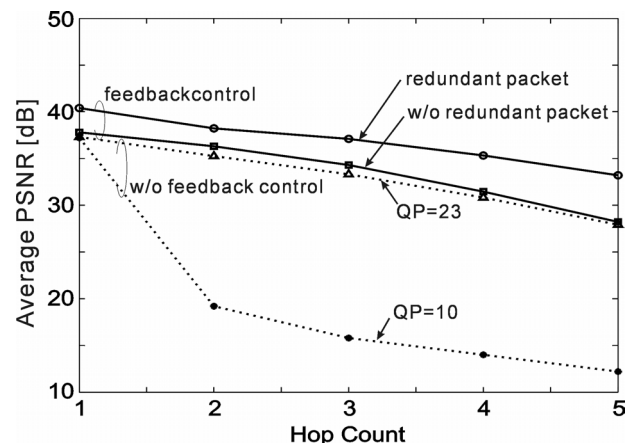

Figure 6. Hop count versus PSNR (Ricean factor K=5).

one to six hops before reaching their destination. The Ricean channel model with different fading factors: K = 10 and K=5, was used.

Fig. 5 and Fig. 6 show the average PSNR (peak-to-peak signal to r.m.s. noise ration) for different hop counts. In these figures the PSNR value is averaged over all the frames within each hop-count period. In order to mitigate the effect of packet drops caused by fading, redundant packet transmission has been considered. In these figures, we also included the results in the absence of a feedback control (i.e., using a fixed QP value) for the purpose of comparison.

As can be seen from these figures, the feedback control scheme can significantly improve the quality of received video signals.

In the case of less severe multipath fading (K = 10 in Fig. 5), the contribution of a redundant packet does not appear to be significant, as packets are not expected to drop in a large quantity. Looking at Fig. 6, we can easily deduce that as the Ricean factor decreases, the effect of redundant packets becomes more apparent.

In our final experiment, we compared all three strategies discussed earlier under the same route change scenario. Bear in

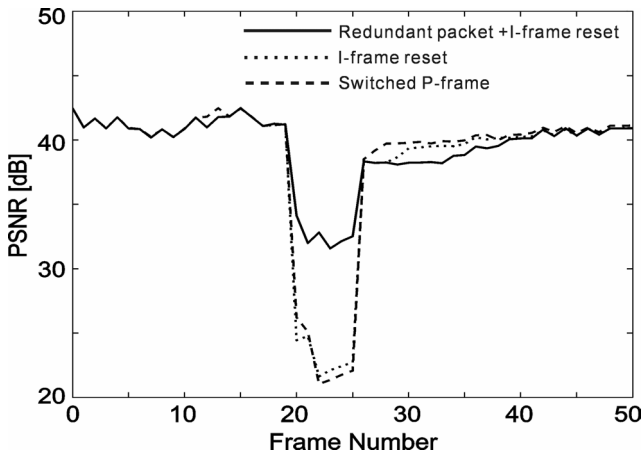

Figure 7. PSNR in a route change scenario with link breakage

mind that these schemes consisted of an intra-reset scheme (I-frame), switched P-frame scheme [11], and the proposed redundant packet transmission scheme.

In the I-frame reset scheme, as soon as the new route is established, the source node switches to the intraframe mode to help re-synchronize the video stream at the receiver. For the other two schemes the number of frames that were affected by a route change was first estimated. By using a conservative estimate, we first identified the last coded frame (based on the recoded time stamp of the RTP stored packets). In the case of switched P-frame, the last decoded frame prior to the link breakage was used for prediction. For the redundant packet transmission scheme, all the RTP redundant packets, which were readily available (i.e., stored in the transmitter buffer for several consecutive frames), were then transmitted before encoding the next frame. To avoid distortion propagation after the new route is established, the next frame was encoded as an I-frame. This is mainly because of the lossy recovery of the previous frames using redundant packet.

Fig. 7 shows the PSNR performance under Ricean fading (K = 100) with a link breakage and a change of hops from 2 to 3. As can be seen from this figure, there is no significant difference between the switched P-frame and I-frame strategies. This is mainly due to the fact that for the switched P-frame, the prediction accuracy is not sufficiently good enough to ensure interframe coding for majority of the blocks in the coding frame. At the same time, with the redundant packet strategy, the PSNR does not drop as abruptly as that of the I-frame reset and switched P-frame.

In terms of subjective evaluation, the redundant packet scheme provides continuity in the video reception. However, the main drawback of the redundant packet transmission is its excessive end-to-end delay requirement.

V. CONCLUSION

The main difficulty in supporting video services in ad-hoc environments is the throughput performance, which tends to deteriorate as the number of hops increase. In addition, the effect of a route change, due to a link failure, can cause the loss of a large number of packets, which can undermine the video resynchronization process at the receiver. To combat this situation it was necessary to develop a feedback mechanism that can dynamically control the packet transmission rate in accordance with the number of transmission hops. In addition, it was shown that generating a redundant packet, which includes only the most sensitive information in a coded video frame can help repair the missing packets under severe fading conditions. Another important contribution of the cross-layer feedback was detection of a link failure. On this basis, we evaluated a number of coding strategies to mitigate the effect of a long burst of packet drops.

REFERENCES

[1] R. L. Lin and J. Liu, "QoS Routing in ad Hoc wireless networks," IEEE Journal on Selected Areas in Communications," Vol.17, No.8, pp. 1426-1438, Aug. 1999.

[2] F. Eshghi and A. K. Elhakeem, "Performance analysis of ad hoc wireless LANs for real-time traffic," IEEE Journal on Selected Areas in Communications, Vol. 21, No. 2, pp. 204-215, Feb. 2003.

[3] S. Mao, S. Lin, S. S. Panwar, and Y. Wang, "Reliable transmission of video over ad-hoc networks using automatic repeat request and multipath transport," IEEE Vehicular Technology Conference, VTC 2001 Fall, Vol. 2, pp. 615 –619, Oct. 2001.

[4] H. Gharavi and K. Ban, " Vision-based ad-hoc sensor networks for tactical operations," World Wireless Congress, 3Gwireless'2002, San Francisco, May 28-31, 2002, pp. 469-474.

[5] H. Gharavi, K. Ban, "Multihop sensor network design for wideband communications," THE PROCEEDINGS OF THE IEEE, vol. 91, NO. 8, August 2003, pp. 1221-1234.

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

[7] C. E. Perkins, E.M. Royer, S.R. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing, " Internet Engineering Task Force, RFC 3561, July 2003.

[8] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications, " RFC1889, Jan. 1996.

[9] T. Weigand, "Document: JVT-G050, "Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), March 2003.

[10] S. Wenger and T. Stockhammer, "An overview of the H.26L NAL concept," Document JVT-B028, JVT 2nd meeting, Jan/Feb 2002.

[11] 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.