Error Resilient FEC Video Transmission Based on Optimal FEC Code Rate Decision

Tae-jun Jung, Joowon Lee, and Kwang-deok Seo
Yonsei University
Wonju, Gangwon, South Korea
e-mail: kdseo@yonsei.ac.kr

Yo-Won Jeong
Samsung Electronics Corporation
Suwon, Gyeonggi, South Korea
e-mail: suance88@hanmail.net

***Abstract*—To support high-quality wireless video transmission over IP, error-resilience techniques are important because wireless video has more stringent requirements than general video transmission for packet loss, latency and jitter. The optimal forward error correction (FEC) code rate decision is a crucial procedure to determine the optimal source and channel coding rates to minimize the overall picture distortion when transporting video packets over packet loss channels. The conventional FEC code rate decision schemes using an analytical source coding distortion model and a channel-induced distortion model are usually complex and typically employ the process of model parameter training, which involves potentially high computational complexity and implementation cost. To avoid the complex modeling procedure, we propose a simple but accurate joint source-channel distortion model to estimate the channel loss threshold set for optimal FEC code rate decision.**

Keywords- error-resilient video transmission; FEC code rate decision; joint source-channel coding; forward error correction.

I. INTRODUCTION

The problem of error control for packet loss in wireless video communication is becoming increasingly important because of the growing interest in video delivery over unreliable channels such as wireless networks and the Internet. With the growth of wireless communication infrastructure, such as Wi-Fi, more and more IP networks are taking a mixed form consisting of wireless and wired channels. One inherent problem in the wireless communications system is that information may be altered or lost during transmission due to channel noise. The effect of such information loss can be devastating for the transport of compressed video because any damage to the compressed bit stream may lead to objectionable visual distortion at the decoder. Moreover, high-quality image and high scene fidelity are extremely important in IP based video transmission to build a complete security [1] [2]. Thus, it is necessary for the video sender to provide adequate error resilience features to protect the video data from the channel errors. Two effective approaches for error resilience and protection are automatic repeat request (ARQ) and forward error correction (FEC) [3] [4] [5]. Due to the stringent delay constraint imposed by real-time video transmission, it is often considered more beneficial to use FEC than ARQ [6] [7]. Because the packet loss rate in the channel changes dynamically, FEC coding with a fixed code rate either wastes channel bandwidth at low packet loss rates or is insufficient to recover video information at high packet loss rates [8]. Thus, it is more efficient to adapt the code rate in response to time-varying packet loss dynamics in order to ensure consistent optimal video quality.

Several previous studies have focused on determining the optimal code rate for joint source-channel coding (JSCC). In [9] and [10], a source coding distortion model and a channel-induced distortion model were proposed as a means of determining a combination of the optimal intra-refresh rate and code rate. These models enable the optimal code rate to be estimated fairly accurately. However, these model equations have numerous parameters that depend on the characteristics of the input video sequences.

In this paper, we propose a practical code rate decision scheme based on a joint source-channel distortion model. The joint source-channel distortion model is used to estimate an accurate channel loss threshold set. With the accurate channel loss threshold set, we can determine the optimum code rate for various channel conditions and therefore adjust the code rate to maintain the maximum video quality under the given channel condition. In order to efficiently apply the joint source-channel distortion model to general live video transmission, we present a practical test run scheme to train the scene-dependent model parameters (SDMP) of the proposed model in real time with acceptable computational complexity. With extensive simulations, we verify the performance of the proposed model and the accuracy of the obtained code rate in various packet loss environments.

This paper is organized as follows. In Section II, we introduce the fundamental video transmission system. Section III describes the loss threshold sets. In Section IV, we present the proposed joint source-channel distortion model. In Section V, we show the experimental results, and finally concluding remarks are presented in Section VI.

II. FUNDAMENTAL VIDEO TRANSMISSION SYSTEM AND RESIDUAL PACKET LOSS RATE

The fundamental structure of the video transmission system over the Internet that supports both source and channel codings is presented in Fig. 1. The video

 ISBN: 978-1-61208-499-2

transmission system consists of a video encoder/decoder, a channel encoder/decoder, and a channel adaptation block. The channel adaption block optimizes the video quality and maintains the end-to-end delay within a given maximum delay bound. The sender periodically receives information about the instantaneous channel status from the receiver through a RTP control protocol (RTCP) feedback report [11]. From this information, the block for the channel characteristic estimation calculates the current status of the channel packet loss rate, p_L, and the available total transmission rate, R_T.

In video coding and transmission over noisy channels, the Reed-Solomon (RS) code is one of the widely used FEC schemes [4]. The channel encoder in Fig. 1 generates n FEC packets for every k video packets by the RS code, denoted as $RS(n,k)$ where n denotes the number of total packets produced by each RS coding. The code rate of RS code is defined as

$$r = \frac{k}{n}. \quad (1)$$

Let R_S and R_C be the source and channel coding bit rates, respectively. If we define the total transmission rate as R_T, then R_S and R_C can be expressed as

$$R_S = R_T \cdot r, \quad R_C = R_T \cdot (1-r). \quad (2)$$

The rate of the packet loss of all data, including video data and FEC redundant data, is generally called the residual packet loss rate (RPLR) [12]. If we discard all the packets in a transmission group when more than $(n-k)$ packets are lost, the RPLR is a good parameter to represent the channel distortion [12]. The RPLR is simply formulated as

$$RPLR = \sum_{i=n-k+1}^{n} \binom{n}{i} p_L^i (1-p_L)^{n-i}. \quad (3)$$

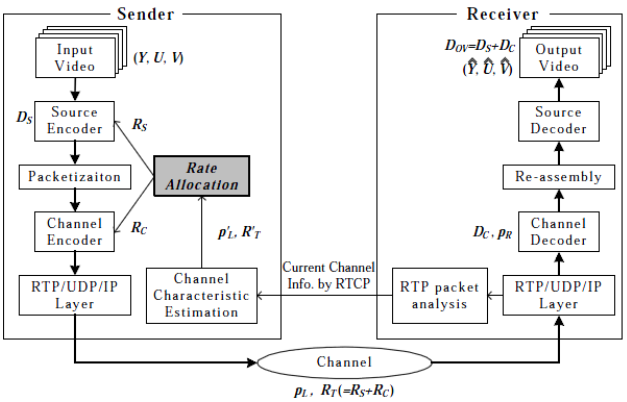

Figure 1. Fundamental transmission structure of the IP video transmission system including source and channel codings.

The overall distortion, D_{OV}, is defined as the mean square error (MSE) of the luminance values (Y component in Fig. 1) of all the pixels between the input video frame of the sender and the decoded video frame of the receiver. The overall distortion can also be represented as the sum of the source-coding distortion, D_S, and the channel-induced distortion, D_C [12].

III. Loss Threshold Sets

To analyze the overall distortion characteristics, we measure overall distortion values for a wide range of channel packet loss rates and various code rates. For this measurement, we employ an H.264/AVC codec [13] and assume that the number of slices per frame is adjusted to three or more in order to prevent the sizes of slices from exceeding the maximum transmission unit (MTU) size. Each slice then becomes one packet. The packet loss characteristic of the channel is assumed to be independent and random. The experimental results of the overall distortion values, $D_{OV}(r,p_L)$, for the test sequence *Foreman* (CIF) are shown in Fig. 2. We can see that the packet loss rates, which serve as thresholds to change the optimal code rate are crucial information. We call the set of these values a *channel loss threshold set* (*CLTS*). In Fig. 2, the CLTS is $\{PL_2, PL_3, PL_4, PL_5\}$.

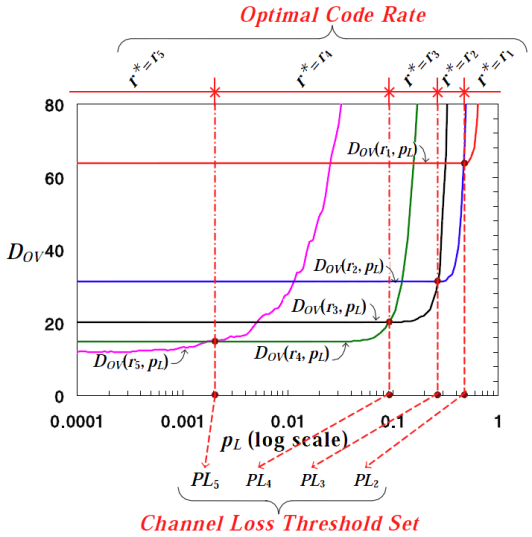

Figure 2. Relationship among overall distortion, code rate, and CLTS.

It is difficult to directly calculate the CLTS because it is impracticable to draw all necessary distortion curves in practical video surveillance applications. In order to efficiently obtain the CLTS, an analytic model is needed. Since the RVPLR is more directly coupled with video quality than the channel packet loss rate, for facile modeling, we introduce the concept of the residual loss threshold set (RLTS), a loss threshold set based on the RVPLR. We define the RLTS as

$$RLTS = \{PR_k \mid PR_k = p_R(n_0, k, PL_k),\ k = 2, 3, \ldots, n_0\}, \quad (4)$$

where $p_L=PL_k$. Note that we can directly obtain the CLTS from the RLTS using (4). The RLTS can be obtained more easily than the CLTS by the proposed joint source-channel distortion model which is explained in the next section.

IV. JOINT SOURCE-CHANNEL DISTORTION MODEL

Obtaining the CLTS by plotting all distortion curves as shown in Fig. 2 whenever the characteristics of the input video change, is extremely time-consuming. To address this problem, we propose a model for the RLTS that can derive all elements of the RLTS, and the CLTS can be easily obtained from the RLTS. This model is based on the following observation: the difference of source coding distortion between r_k and r_{k-1} is likely to be equal to the channel distortion at $r=r_k$ and $p_L=PL_k$. To illustrate this, the above statement is represented in Fig. 3 for k = 2, 3, 4. Note that in Fig. 3 the notation $D_C(r,p_L)$ refers to the channel distortion when the code rate is r and the channel packet loss rate is p_L. This observation is attributed to the geometric characteristic that the $D_{OV}(r_k,p_L)$ curve abruptly decreases and the $D_{OV}(r_{k-1},p_L)$ curve maintains a nearly constant value at their intersection. We can formulate the observation as follows:

$$D_{OV}(r_{k-1},0)-D_{OV}(r_k,0)\approx D_C(r_k,PL_k), \qquad k=2,3,\ldots,n_0. \tag{5}$$

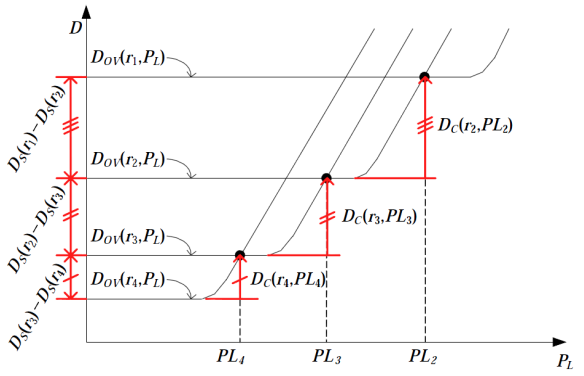

Figure 3. Conceptual distortion curves.

We can see that no ratio values between $|D_{OV}(r_{k-1},0)-D_{OV}(r_k,0)|$ and $|D_C(r_k,PL_k)|$ fall out of the range between 1 and 0.75. From this result, (5) can be represented as

$$0.75\le\left|\frac{D_{OV}(r_{k-1},0)-D_{OV}(r_k,0)}{D_C(r_k,PL_k)}\right|\le 1 \qquad k=2,3,\ldots,n_0. \tag{6}$$

Because the ratio of $|D_{OV}(r_{k-1},0)-D_{OV}(r_k,0)|$ to $|D_C(r_k,PL_k)|$ is narrowly bounded, (6) can be approximated as a constant value

$$\left|\frac{D_{OV}(r_{k-1},0)-D_{OV}(r_k,0)}{D_C(r_k,PL_k)}\right|\approx c, \quad 0.75\le c\le 1, \qquad k=2,3,\ldots,n_0. \tag{7}$$

The relationship between the source coding rate R_S and the distortion yielded by the source coding D_S is called the rate-distortion model (R-D model). In this study, we employ an inverse proportional model because it is simple and accurate [12]. $D_{OV}(r_k,0)$ can then be represented as

$$D_{OV}(r_k,0)=\frac{w_1}{R_T r_k-w_2}+w_3, \qquad k=2,3,\ldots,n_0. \tag{8}$$

where $R_T r_k$ denotes the source coding rate, and w_1, w_2, and w_3 are SDMPs that can be varied according to input video characteristics. Note that, unlike w_1 and w_2, w_3 can be varied by the RVPLR of the previous state even if the input video characteristic is unchanged because the current distortion can be affected by the previous distortion. We can see that the distortion of the received video was proportional to the RVPLR regardless of k. Therefore, we can represent $D_C(r_k,PL_k)$ as

$$D_C(r_k,PL_k)=\tilde{D}_C(k,PR_k)=w_4 PR_k, \qquad k=2,3,\ldots,n_0, \tag{9}$$

where w_4 is an SDMP. If we substitute (8) and (9) into (7) and rearrange for PR_k, we finally obtain the set of PR_ks as

$$PR_k=\frac{\beta}{(k-(\alpha+1))(k-\alpha)},\ \alpha=\frac{w_2 n_0}{R_T},\ \beta=\frac{w_1 n_0}{c\cdot w_4 R_T}, \quad k=2,3,\ldots,n_0, \tag{10}$$

where α and β are the final SDMPs of the joint source-channel model.

If we equivalently express (10) as

$$PR_k=\frac{\beta}{k\left(1-\left(\frac{\alpha}{k}+\frac{1}{k}\right)\right)\cdot k\left(1-\frac{\alpha}{k}\right)}, \qquad k=2,3,\ldots,n_0, \tag{11}$$

and assume that k is much larger than α, namely $|\alpha/k|\ll 1$, we can approximate (11) to (12).

$$PR'_k=\frac{\beta}{k(k-1)}, \qquad k=2,3,\ldots,n_0. \tag{12}$$

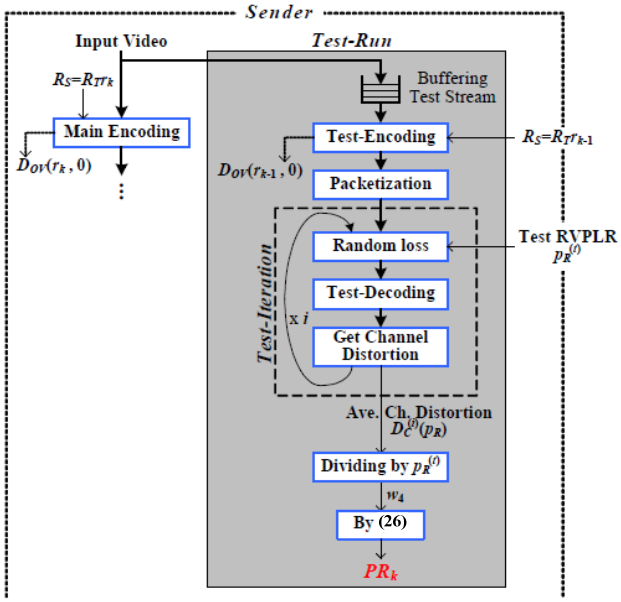

Figure 4. The overall process of the test run in the sender.

To obtain the SDMPs, α and β of (11), we need PR_i and PR_j where i and j are different values among 2, 3, …, n_0. In the case of the approximate model of (12), we need only one PR_k. The test run is a real-time process for obtaining the sample PR_ks. Because the test run should not disturb the real-time performance of the basic process of the sender of Fig. 1, it is performed using residual computing resources whenever the basic process of the sender is in an idle state. Since such residual computing resources are generally

 ISBN: 978-1-61208-499-2

insufficient, the computational load of the test run should be reduced.

The overall process of the test run for obtaining a sample PR_k is based on

$$\left|\frac{D_{OV}(r_{k-1},0)-D_{OV}(r_k,0)}{w_4 PR_k}\right| \approx c, \quad 0.75 \le c \le 1, \qquad k=2,3,...,n_0, \qquad (13)$$

which is derived by substituting (9) into (7). To obtain a sample PR_k in (13), we have to obtain three values, $D_{OV}(r_{k-1},0)$, $D_{OV}(r_k,0)$, and w_4, in advance. The overall process of the test run, including the processes for obtaining these three values, is shown in Fig. 4.

Since there is a high possibility that the characteristics of an input video will change significantly when scene-changes occur or a scene is long, the test run needs to be performed periodically. We buffer the test stream whenever a test run is started. The additional encoding by the test run generally requires a huge computational load, because the video encoding includes various complex operations, such as a motion estimation, a discrete cosine transform (DCT), and an inverse DCT. Of these operations, motion estimation takes the most time in the encoding process [14]. To reduce the computational load of motion estimation, we reuse the motion vector information evaluated via main encoding. The main encoding and the test encoding are performed using the same input frames, and the source coding rates $R_T \cdot r_{k-1}$ and $R_T \cdot r_k$ are so similar that motion vectors of the same macroblock position resulting from the main encoding and the test encoding are very similar [15].

V. EXPERIMENTAL RESULTS

The experiment was performed using a desktop computer system with a Quad-Core 4.0 GHz CPU and 2 GByte memory. For input video, we used eight test sequences with 720p HD (1280x720) resolution, *Stockholm*, *Sunflower*, *Tractor*, *Troy*, *Blue Sky*, *Rush Hour*, *Park Joy*, *Citybus*, for which the frame rates are 15 fps. The basic parameters of the experiment are as follows: n_0 is 10 and R_T is 1,536 Kbps. For the test run, we stored each test sequence for the first second as a test stream. We assumed that the main encoding was performed with k=8. The test RVPLR was fixed at 1/15. For each experiment, we compare the three types of systems discussed below.

1) Optimal system (OS): In this system, the optimal code rate is considered to be the code rate that minimizes the sum of D_S and D_C, which are formulated as in Eqs. (16) and (18), respectively. In the test encoding, we perform two additional encodings to obtain three SDMPs of (16) for the test stream with r_k=0.2 (k=2) and 0.9 (k=9). In the test-iteration, the loss rate is uniformly set to 1/15 regardless of class, and the number of iterations is set to 30.

2) RLTS model-based system (RMS): In this system, we calculate the optimal code rate through the proposed RLTS model in (19). To obtain two SDMPs of (19), we need two PR_k samples. One sample can be obtained by test-encoding with r_k=0.7 (k=7) because the main encoding is performed with r_k=0.8. To obtain another sample, we perform two test-encodings with r_k=0.3 and 0.4.

3) Approximate model-based system (AMS): In this system, we calculate the optimal code rate through the approximate RLTS model. To obtain the SDMP, we need one PR_k sample. It can be obtained by test-encoding with r_k=0.7. Other settings for the test-iteration are the same as RMS.

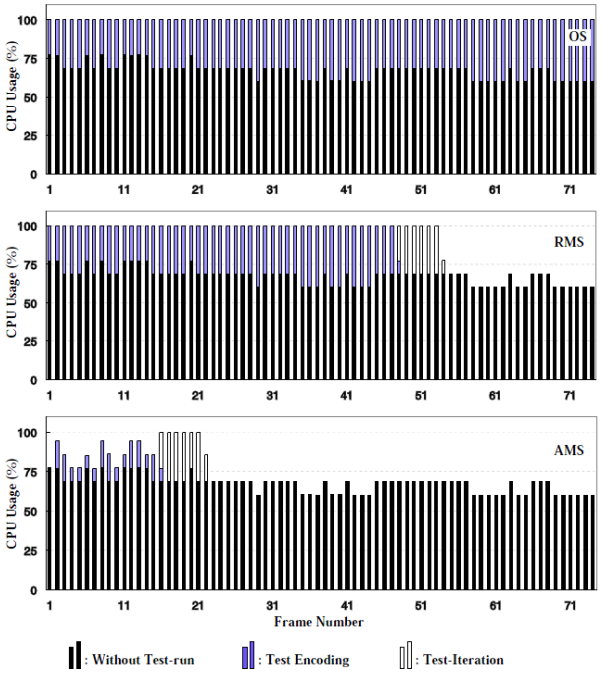

Figure 5. The Comparison of processing overheads of test runs for the three types of systems: OS, RMS, and AMS. The *Stockholm* sequence of HD resolution is used.

The experimental results of processing overheads required for the test run for each frame of the *Stockholm* sequence are shown in Fig. 5. The black bar represents the CPU usage of each frame for the entire processing without the test run. The remaining CPU resource can be used for the test run. The blue and white bars represent the CPU usage for the test-encodings and test-iteration, respectively. In the case of an OS (optimal system), the test-encodings are not completed at the end of the sequence because the OS requires two additional test-encodings that introduce a heavy computational load. In the case of RMS, the test run is completed before the end of the sequence. However, because the test run is completed too late, the period during which the output of the test run can be used is very short. On the other hand, the AMS requires only one test-encoding, and we apply all the speed-up methods described in Section 4. Thus, the test run is consequently completed very rapidly, and the output of the test run can be used over a substantial time period

The experimental results of the PSNR performance for the channel packet loss rate are shown in Table I. The PSNR values are obtained as follows. First, we perform a test run

 ISBN: 978-1-61208-499-2

via three methods: the OS, the RMS, and the AMS. Next, we determine the optimal code rate for the channel packet loss rate through the model and its SDMPs obtained from each test run method. After applying this code rate, we simulate the video transmission system of Fig. 1 and calculate the average PSNR of the total frames of the test sequence. Finally, all the PSNR results are obtained using 50 runs of the above procedure in order to obtain statistically meaningful average values [16]. In Table I, AMS-PR and AMS-UR are the PSNR results when applying the pseudo-random loss and uniform-random loss, respectively.

TABLE I. SUMMARY OF THE PSNR RESULTS FOR TEST SEQUENCES

Test sequence	Test-Run	Avg. PSNR (dB)	Max. PSNR gap for OS
Stockholm	OS	28.653	
	RMS	28.509	1.631
	AMS-PR	28.288	2.574
Sunflower	OS	33.258	
	RMS	33.249	0.283
	AMS-PR	33.026	3.150
Tractor	OS	32.257	
	RMS	32.197	0.877
	AMS-PR	31.980	2.150
Troy	OS	30.513	
	RMS	30.456	0.567
	AMS-PR	30.173	3.198
Blue Sky	OS	33.278	
	RMS	33.222	0.575
	AMS-PR	32.932	2.746
Rush Hour	OS	29.171	
	RMS	29.105	0.472
	AMS-PR	28.836	2.199
Park Joy	OS	24.547	
	RMS	24.497	0.771
	AMS-PR	24.210	3.008
Citybus	OS	31.106	
	RMS	31.007	0.184
	AMS-PR	29.772	3.207

In Table I, the performance of the AMS-PR is very close to the performance of the OS except for the high loss rate. At about p_L=0.6, the maximum PSNR gap between the AMS-PR and the OS is 2.57dB (*Stockholm*) and 3.15dB (*Sunflower*). The range of p_L having a high PSNR gap is not long enough to be considered a serious drawback. The gap between the average values of the PSNR curve of the AMS-PR and the OS is just 0.36dB (*Stockholm*) and 0.23dB (*Sunflower*). For other test sequences, similar results are observed.

VI. CONCLUSIONS

In this paper, we propose a simple but accurate joint source-channel distortion model to estimate the channel loss threshold set for optimal FEC code rate decision. The proposed joint model exempts us from a complex training procedure for obtaining many model parameters for separate source and channel models, which is usually required in the conventional code rate decision approach. Since the proposed model is expressed as a simple closed form and has a small number of SDMPs, a video sender using the model can be easily implemented. For training the SDMPs in real time, we propose a practical test run procedure. This method increases the speed of the test run while maintaining its accuracy for training the SDMPs. When compared to the previous methods, by using the proposed simple model and practical test run method, the video sender can determine the optimal code rate for JSCC whenever there is a change in the packet loss condition in the channel. The experimental results confirm that the proposed model and its test run procedure can accurately estimate the channel loss threshold set in real time, resulting in an optimal FEC code rate with low computational complexity.

ACKNOWLEDGMENT

This work was supported by a grant 'Biotechnology & GMP Training Project' from the Korea Institute for Advancement of Technology (KIAT), funded by the Ministry of Trade, Industry and Energy (MOTIE) of the Republic of Korea (N0000961).

REFERENCES

[1] Y. Abudoulikemu, Y. Huang, and C. Ye, "A scalable intelligent service model for video surveillance system based on RTCP," *Int. Conf. on Signal Processing Systems*, vol. 3, pp. 346-349, 2010.

[2] D. Chu, C. Jiang, Z. Z. Hao, and W. Jiang, "The design and implementation of video surveillance system based on H.264, SIP, RTP/RTCP and RTSP," *Int. Symp. on Computational Intelligence and Design (ISCID)*, vol. 2, pp. 39-43, 2013.

[3] D. Wu, Y. T. Hou, and Y. Q. Zhang, "Transporting real-time video over the Internet: Challenges and approaches," *Proc. of the IEEE*, vol. 88, no. 12, pp. 1855-1877, Dec. 2000.

[4] Y. Wang and Q. Zhu, "Error control and concealment for video communication: A review," *Proc. of the IEEE*, vol. 86, no. 5, pp. 974-997, May 1998.

[5] S. Jalil, M. Abbad, and R. Azouzi, "Hybrid FEC/ARQ schemes for real-time traffic in wireless networks," *Int. Conf. on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1-6, 2015.

[6] S. Soltani, K. Misra, and H. Radha, "Delay constraint error control protocol for real-time video communication," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 742-751, June 2009.

[7] S. Jalil, M. Abbad, and R. Azouzi, "Hybrid FEC/ARQ schemes for real-time traffic in wireless networks," *Int. Conf. on Wireless Networks and Mobile Communications (WINCOM)*, pp. 1-6, 2015.

[8] N. Abdelhamid, T. Taleb, and L. Murphy, "Forward error correction strategies for media streaming over wireless networks," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 72-79, 2008.

[9] K. Stuhlmuller, N. Farber, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 1012-1032, June 2000.

[10] P. Frossard and O. Verscheure, "Joint Source/FEC Rate Selection for Quality-optimal MPEG-2 Video Delivery," *IEEE Trans. on Image Proc.*, vol.10, no.12, pp.1815-1824, Dec. 2001.

[11] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Real-time transport protocol," IETF *RFC 3550*, July 2003.

 ISBN: 978-1-61208-499-2

[12] K. Stuhlmuller, N. Farber, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 1012-1032, June 2000.

[13] Joint Video Team (JVT), H.264/AVC Reference Software Version JM 19.0, available from http://iphome.hhi.de/suehring/tml/ download/, retrieved: June 2016.

[14] J. Osterman *et al.*, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7-28, First Quarter 2004.

[15] H. Zhou, J. Zhou, and X. Xia, "The motion vector reuse algorithm to improve dual-stream video encoder," in Proc. *Int. Conf. Signal Process*., pp. 1283-1286, Oct. 2008.

[16] Q. Qu, Y. Pei, and J. W. Modestino, "An adaptive motion-based unequal error protection approach for real-time video transport over wireless IP networks," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 1033-104, Oct. 2006.

 ISBN: 978-1-61208-499-2