This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2010 proceedings

Joint Video Source Coding and Distributed Link Adaptation in Wireless Mesh Networks

Antonios Argyriou
Department of Electrical and Computer Engineering
University of Thessaly, Greece

Haiyan Luo, Dalei Wu and Song Ci
Department of Computer and Electronics Engineering
University of Nebraska-Lincoln, USA

***Abstract*—Real-time video communication over wireless multi-hop networks has gained significant interest in the last few years. In this paper, we focus our attention on the problem of source coding and link adaptation for packetized video streaming in wireless multi-hop networks when network nodes are media-aware. We consider a system where source coding is employed at the video encoder by selecting the encoding mode of each individual macro-block, while error control is exercised through application-layer retransmissions at each media-aware network node. For this system model, the contribution of each communication link on the end-to-end video distortion is considered separately in order to achieve globally optimal source coding and ARQ error control. To reach the globally optimal solution, we formulate the problem of Joint Source and Distributed Error Control (JSDEC) and devise a low-complexity solution algorithm based on dynamic programming. Extensive experiments have been carried out on the basis of H.264/AVC codec to demonstrate the effectiveness of the proposed algorithm over the existing Joint Source and Channel Coding (JSCC) algorithm in terms of PSNR perceived at the decoder under time-varying multi-hop wireless links.**

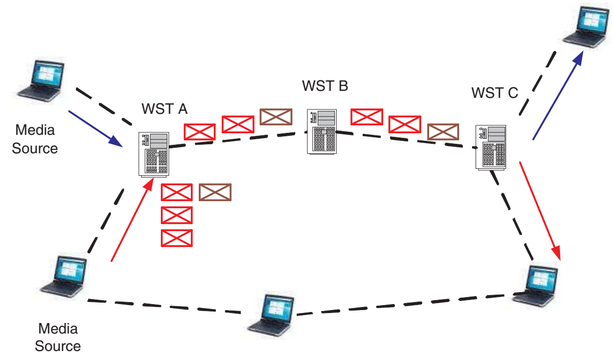


Fig. 1. The general scenario addressed by this paper consists of multiple wireless stations (WSTs) that generate real-time encoded unicast bitstreams and they are subsequently forwarded by the WSTs in the end-to-end wireless path.

I. Introduction

Real-time video communication services such as video telephony, video conferencing, video gaming, and mobile TV broadcasting, are considered very important applications for wireless multi-hop networks. However, the dissemination of pre-compressed or real-time video over wireless networks is characterized by several problems [1]. A particular characteristic of existing and emerging networks that is usually overlooked by video streaming applications, is that an end-to-end path consists of several interconnected physical networks which are heterogeneous and have asymmetric properties in terms of throughput, delay, and packet loss. Congested last-mile wireline links (e.g., DSL links) or wireless access networks (e.g., WiFi) are some of the real-life examples. In such networks, employing measurements or congestion control in the end-to-end fashion might not provide end-systems with a correct status of the network characteristics [2], [3]. For example, some wireless hops in the end-to-end path as shown in Fig. 1 may face bad channel conditions. For a video streaming application, a pure end-system implementation might unnecessarily limit the choices of the video encoder or the streaming algorithms with respect to the optimal streaming rate and the error control strategy along the end-to-end path. Therefore, proxy-based solutions are usually proposed to deal with this situation. However, for real-time video encoding systems, the majority of the current proxy-based streaming approaches have mainly focused on wireless networks only at the last hop [1]. Moreover, even for the wireline networks, most of the current research only makes use of a single proxy [4].

We have recognized the importance of this problem in [2] where we considered pre-compressed video streaming over a wireline network. In this paper, we address the aforementioned problems in a systematic fashion by proposing a novel mechanism that integrates into a joint optimization framework the parameters that affect the encoding and transmission of a video stream in a multihop mesh network. Toward this end, we focus on generalizing the approach on the basis of Joint Source and Channel Coding (JSCC) which can integrate both network/transport parameters and the source characteristics for improving the system performance [5], [6]. The objective of JSCC is to distribute the available channel rate between source and channel bits so that the decoder distortion is minimized. However, the situation that we described will essentially translate into sub-optimal JSCC allocation when it is exercise over several physical channels [3].

II. Proposed System

Both the distortion estimation and resource allocation algorithms are implemented at the streaming server and encoder. The encoder executes the algorithm for joint source and channel allocation by striving to minimize the global distortion. To achieve this, the encoder calculates for each macroblock the estimated source distortion and channel distortion. The average channel distortion is calculated for each packet erasure channel h, based on the reported feedback from the corresponding proxy (packet erasure rate ϵ_h). By using these estimates, the encoder calculates the optimal encoding mode μ for each individual macroblock (source coding), while the optimal number of retransmissions σ for the corresponding transport packet (error control) is calculated locally by each proxy. Therefore, source-channel coding is applied jointly, i.e. the encoder decides the encoding mode while it also indirectly determines the optimal rate dedicated to channel coding with ARQ. An important advantage of the JSDEC algorithm is that the proxy does not require the notification of the optimal rate for channel coding. Furthermore, each proxy decides individually the maximum number of allowed retransmissions for a particular video packet, since it can deduce the optimal channel coding rate.

A. End-to-End Distortion Estimation

The Recursive Optimal Per-pixel Estimate (ROPE) algorithm has been adopted to calculate distortion recursively across frames [7], meaning that the estimation of the expected distortion for a frame currently being encoded is derived by considering the total distortion introduced in previous frames.

We consider a N-frame video clip $\{f_1, ..., f_n, ..., f_N\}$. During encoding, each video frame is divided into 16 $\times$ 16 macroblocks (MB), which are numbered in the scan order. In our implementation, the packets are constructed such that each packet consists of a row of MBs and is independently decodable. The terms of row and packet sometimes are interchangeably used in this paper. When a packet is lost during transmission in the network, we use the temporal-replacement error concealment strategy. Therefore, the motion vector of a missing MB is estimated as the median of motion vectors of the nearest three MBs in the preceding row. If the previous row is lost too, the estimated motion vector is set to zero. The pixels in the previous frame, which are pointed by the estimated motion vector, are used to replace the missing pixels in the current frame.

Let I be the total number of packets in one video frame, and J the total number of pixels in one packet. Let us denote $f^j_{n,i}$ the original value of pixel j of the ith packet in frame n, $\hat{f}^j_{n,i}$ the corresponding encoder reconstructed pixel value, $\tilde{f}^j_{n,i}$ the reconstructed pixel value at the decoder, $E[d^j_{n,i}]$ the expected distortion at the receiver for pixel j of the ith packet in frame n. Then the total expected distortion $E[D]$ for the entire video sequence can be calculated by summing up the expected distortion of all the pixels

$$E[D] := \sum_{n=1}^{N}\sum_{i=1}^{I}\sum_{j=1}^{J} E[d^j_{n,i}] \tag{1}$$

where the Mean-Squared Error (MSE) is

$$\begin{aligned} E[d^j_{n,i}] &= E[(f^j_{n,i} - \tilde{f}^j_{n,i})^2] \\ &= (f^j_{n,i})^2 - 2f^j_{n,i}E[\tilde{f}^j_{n,i}] + E[(\tilde{f}^j_{n,i})^2] \end{aligned} \tag{2}$$

Since $\tilde{f}^j_{n,i}$ is unknown to the encoder, it can be thought as a random variable. To compute $E[d^j_{n,i}]$, the first and second moments of $\tilde{f}^j_{n,i}$ are are calculated similar to [7].

III. The Error Control Proxy

The functionality realized at the proxy is an application-layer delay-constrained ARQ error control algorithm for video packets. ARQ is exercised on a local scope only between two successive nodes (i.e., proxy, sender, or receiver). A negative acknowledgment scheme is used for this purpose. In practice, the proxy is a network node that terminates RTP sessions [8]. This is necessary since it has to monitor the sequence number space of each RTP flow. Part of the available bandwidth at the proxy is allocated to forward incoming video packets. The remaining of the available bandwidth is used for retransmitting packets locally stored at the proxy.

A. Packet Loss Rate

Based on the above design, we can calculate the impact of the ARQ algorithm at each proxy on the aggregate packet loss rate and the latency experienced by the transmitted video packets. The communications network considered in this work is packet-switched, and the adopted packet loss model is an "erasure channel". We consider a Markov chain for characterizing transitions from good to bad states of each individual channel. For this channel model, if the packet erasure rate of hop h at the network layer is ϵ_h and the average number of retransmissions for a particular video transport packet i is $m_{h,i}$, then the resulting residual packet loss rate is $\epsilon_h^{m_{h,i}}$. Sometimes, packet loss is not caused by packet erasure ($\epsilon_h^{m_{h,i}}$) rather the excessive delay τ_h for the hop h. If the transmission delay for hop h is L_h, the above probability can be expressed as $P_r\{L_h > \tau_h\}$. Then the overall packet loss rate due to packet erasure and packet delay is:

$$\rho_{h,i} = \epsilon_h^{m_{h,i}} + (1 - \epsilon_h^{m_{h,i}})P_r\{L_h > \tau_{h,i}\} \tag{3}$$

An important parameter of this formula is the actual distribution of retransmissions for a given permissible range. This value can be calculated using the adopted channel model. The probability of k retransmissions for a successful delivery of packet i is given by:

$$\pi(k, \sigma_{h,i}) = \frac{(1-\epsilon_h)\epsilon_h^k}{1-\epsilon_h^{\sigma_{h,i}+1}}. \tag{4}$$

where $\sigma_{h,i}$ is the maximum transmissions for source packet i on link h. Given $\sigma_{h,i}$ and ϵ_h, we can also calculate the average number of retransmissions per packet i for hop h as:

$$m_{h,i} = \frac{1-\epsilon_h^{\sigma_{h,i}+1}}{1-\epsilon_h} \tag{5}$$

Equation (4) accounts for the two possible reasons of packet loss in the network, namely excessive delay and channel erasure. In literature, this method of calculating the residual PER was first introduced in the seminal work of [9]. Regarding Equation (5), it captures the channel erasure effect as Bernoulli distribution as well as the effect of truncated ARQ by $\sigma_{h,i}$.

Therefore, the *average* packet transmission delay for each hop h is given by

$$L_h(i) = \sum_{k=0}^{\sigma_{h,i}} \pi(k, \sigma_{h,i})[k * RTT_h + FTT_h], \quad (6)$$

where FTT_h and RTT_h are the forward and the round trip delay of hop h, respectively. Thus, within the proposed framework, error control (i.e. channel coding) is exercised locally at each proxy by enforcing the maximum number of retransmissions for each specific packet and hop $\sigma_{h,i}$.

Therefore, we can express the overall end-to-end packet loss rate for H tandem proxies as

$$\rho_i = 1 - \prod_H (1 - \rho_{h,i}). \quad (7)$$

The only parameter which needs to be estimated is the forward delay of each of the tandem-connected links. In this work, we model the one-way network delay as a Gamma distribution, since this distribution captures the main reason of network packet delays, which is due to buffering in the wireless nodes [10].

IV. Optimization Framework for Source Coding and Distributed Error Control

In this section, we jointly optimize the source coding and distributed error control parameters within our proposed framework. The goal is to minimize the perceived video distortion for given available link capacity and packet error rate. First, we formulate the problem as a constrained minimum distortion problem. Then, we give the optimal solution using dynamic programming.

A. Problem Formulation

In the following, we will define the rate-distortion optimization problem. Let $\vec{\mu}(n)$ and $\vec{\sigma}(n)$ denote the vectors of source coding and channel error control parameters for the n-th frame, respectively. Then, the end-to-end distortion minimization problem is:

$$\min_{\mu \in \mathbf{U}, \sigma \in \mathbf{\Sigma}} \sum_{n=1}^{N} \sum_{i=1}^{I} E[D_{n,i}](\vec{\mu}(n), \vec{\sigma}(n))$$
$$\text{s.t. } R_S + R_{C_h} \leq R_{T_h} \quad \forall h \in H \quad (8)$$

where $\mathbf{U}$ and $\mathbf{\Sigma}$ are the entire sets of available vectors of $\vec{\mu}(n)$ and $\vec{\sigma}(n)$ respectively, and H is the number of tandem-connected nodes. R_S, R_C denote the rates allocated to source coding and error control, respectively. R_T is the total available channel rate. Furthermore, the available channel rate R_{T_h} can be denoted in terms of channel bandwidth W_{T_h}:

$$R_{T_h} = W_{T_h} log_2(1 + \xi) \quad (9)$$

where ξ is the time-varying channel SNR. Therefore, the fluctuating channel data rate is reflected. The rate constraint for retransmitted packets must be satisfied by every proxy in the end-to-end path. Recall that the average number of retransmissions at each proxy is $m_{h,i}$ and the residual network packet loss rate is $\rho_{h,i}$. Therefore, the bit rate constraint that needs to be satisfied for frame n at each proxy h is:

$$\sum_{i=1}^{I} R_s(\vec{\mu}(n)) \cdot m_{h,i} + \sum_{i=1}^{I} S_i \cdot m_{h,i} \leq R_{T_h} \quad \forall h \in H \quad (10)$$

where S_i is the size parity bits of packet i. The first term in Equation (10) denotes the source coding rate for frame n, and the second term is the channel rate estimation for the retransmitted packets of frame n. What we have achieved in (10) is to decouple the transmission rate constraints for the paths connected in tandem.

B. The Optimal Solution for the Minimum Distortion Problem

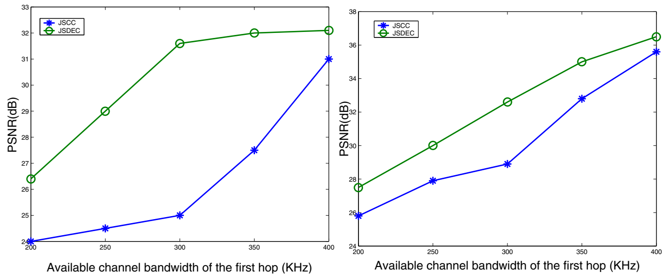
For simplicity, let us denote the parameter vector of packet i in video frame n as $\mathcal{V}_w := \{\vec{\mu}(n), \vec{\sigma}(n)\}$, where $w(1 \leq w \leq N \times I)$ is the index of the packet i of the whole video clip. Clearly, any selected parameter vector $\mathcal{V}_w$ resulting in the total bit rate of source and channel coding to be greater than R_{T_h} is not in the optimal parameter vector $\mathcal{V}_w^* := \{\vec{\mu}^*(n), \vec{\sigma}^*(n)\}$. Therefore we set the average distortion of a packet with bit rate larger than the maximum allowable bit rate to infinity, meaning that a feasible solution will not result in any bit rate greater than R_{T_h}. Therefore, the minimum distortion problem with rate constraint can be transformed into an unconstrained optimization problem. Without losing generality, we assume that the current packet will depend on the latest a packets $(a \geq 0)$, due to the concealment strategy. To solve the optimization problem, we define a cost function $\mathcal{C}_i(\mathcal{V}_{i-a}, \ldots, \mathcal{V}_i)$ which represents the minimum average distortion up to and including the packet i, given that $\mathcal{V}_{i-a}, \ldots, \mathcal{V}_i$ are the decision vectors for the packets $(i-a), \ldots, i$. Let $\mathcal{P}$ be the total packet number of the video clip, and we have $\mathcal{P} := N \times I$. Therefore, $\mathcal{C}_{\mathcal{P}}(\mathcal{V}_{\mathcal{P}-\mathbf{a}}, \ldots, \mathcal{V}_{\mathcal{P}})$ represents the minimum total distortion of the whole video clip. Thus solving (8) is equivalent to solve

$$\underset{\mathcal{V}_{\mathcal{P}-a}, \ldots, \mathcal{V}_{\mathcal{P}}}{\text{minimize}} \ \mathcal{C}_{\mathcal{P}}(\mathcal{V}_{\mathcal{P}-a}, \ldots, \mathcal{V}_{\mathcal{P}}). \quad (11)$$

The key observation for deriving an efficient algorithm is the fact that given $a+1$ control vectors $\mathcal{V}_{i-a-1}, \ldots, \mathcal{V}_{i-1}$ for the packets $(i-a-1), \ldots, (k-1)$, and the cost function $\mathcal{C}_{i-1}(\mathcal{V}_{i-a-1}, \ldots, \mathcal{V}_{i-1})$, the selection of the next control vector $\mathcal{V}_i$ is independent of the selection of the previous control vectors $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_{i-a-2}$. This means that the cost function can be expressed recursively as

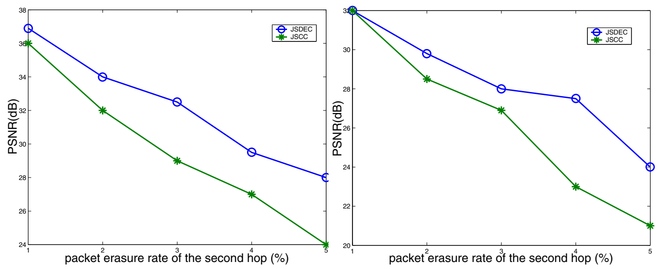
$$\mathcal{C}_i(\mathcal{V}_{i-a}, \ldots, \mathcal{V}_i) = \underset{\mathcal{V}_{i-a-1}, \ldots, \mathcal{V}_{i-1}}{\text{minimize}} \{\mathcal{C}_{i-1}(\mathcal{V}_{i-a-1}, \ldots, \mathcal{V}_{i-1}) + E[D_{n,i}]\}. \quad (12)$$

This recursive representation of the cost function makes the future step of the optimization process independent of its past steps, which consists of the foundation for dynamic programming. Therefore, the problem can now be converted into a graph theory problem of finding the shortest path in a directed acyclic graph (DAG).



(a) $W_{T2} = 0.3MHz$ (b) $W_{T2} = 0.6MHz$

Fig. 2. Frame PSNR comparison between the proposed JSDEC system and the JSCC system with constant packet loss rates. ($\epsilon_1 = \epsilon_2 = 2\%$)



(a) $\epsilon_1 = 2\%$ (b) $\epsilon_1 = 5\%$

Fig. 3. Frame PSNR comparison between the proposed JSDEC system and the JSCC system with constant bandwidth. ($W_{T1} = W_{T2} = 0.3MHz$)

V. Experimental Results

In this section, we design experiments to evaluate the performance of the proposed JSDEC system. A topology similar to the one shown in Fig. 1 is used for experiments in this paper. The QCIF (176 $\times$ 144) sequence "Foreman" is adopted for real-time encoding with the H.264/AVC JM12.2 codec [11]. The network topology is simulated using NS-2. All frames except the first are encoded as P frames. The target frame rate is set to 30 frames/second. The channel is assumed to be frequency-flat and time-invariant during the transmission of a packet but may vary from packet to packet. The average SNR of each hop in a given path is $\bar{\xi}$, then the random link quality ξ experienced by each packet is generated according to Rayleigh distribution, whose Probability Density Function (PDF) is:

$$p_\xi(\xi) = \frac{1}{\bar{\xi}} e^{-\frac{\xi}{\bar{\xi}}} \quad (13)$$

where $\bar{\xi} := E\{\xi\}$ is the average received SNR.

The propagation delay of each link is set to $10\mu s$. In the simulations, quantization step size (QP) q is considered for source coding parameter. PSNR is adopted to evaluate the distortion between the original sequence and the reconstructed sequence at the receiver. Since the duration of the given video sequence is short (300 frames), the initial startup delay at the decoder playback buffer is set to five frames. In the case of a buffer underflow, re-buffering is performed until two more frames are received. We compare the proposed JSDEC system with a typical JSCC system where each proxy employs error control individually, without coordination with the encoder. In the adopted JSCC system, channel quality information (packet loss rate) is fed back to the source node for the optimization of the encoder behavior by choosing the optimal source coding parameters. However, this existing JSCC system is not able to control the ARQ behavior. Both systems adopt the same packetization scheme under the same network environment to ensure fairness. In all experiments, we set the number of frames stored at each proxy equal to four.

First, we present the experimental results for the proposed JSDEC system comparing with the existing JSCC system, when the packet loss rate is constant. The results for two tandem connected hops (i.e. one ARQ proxy) when $W_{T2} = 0.3MHz$ and $R_{T2} = 0.6MHz$ are shown in Fig. 2(a) and Fig. 3(b), respectively. We compare the performance difference of the JSCC and JSDEC systems while the feedback sent from the proxy back to the encoder is subject to a delay equal to the transmission delay of one frame. The two links attached to the same proxy are configured with the same packet erasure rate of $\epsilon_1 = \epsilon_2 = 2\%$. The y-axis corresponds to PSNR (dB) while the x-axis corresponds to the available channel bandwidth on the first hop of the path (i.e., W_{T1}).

In Fig. 2(a), the available channel bandwidth of the second hop is set to $0.3MHz$, (i.e., $W_{T2} = 0.3MHz$) while the available channel bandwidth of the first hop W_{T1} changes from $0.2MHz$ to $0.4MHz$ at a step size of $50KHz$. Our simulation results indicate that the proposed JSDEC system achieves the largest performance gain (about 6dB) at around $W_{T1} = W_{T2} = 0.3MHz$. When W_{T1} is either increased or decreased from $0.3MHz$, the performance gain is lower, but still significant compared with the JSCC system. However, if the bandwidth difference between the two available channels become even larger, the performance gain will decrease significantly. For instance, when W_{T1} is at either $0.2MHz$ or $0.4MHz$, there is only 1-2dB PSNR performance increase. To explain this, when W_{T2} is fixed at $0.3MHz$, the much lower value of W_{T1} constrains the upper bound of the source encoding rate. This indirectly allows most of the available rate on the second channel R_{T2} to be used for error control and thus leads to minimal channel distortion. Similarly, if the value of W_{T1} is much higher than $0.3MHz$, the channel on the first hop introduces little channel distortion because of the significant spare bandwidth available for error control. Nevertheless, when the network does not operate in such highly asymmetric as far as channel rates are concerned, the performance improvement of decoded video quality of JSDEC over JSCC is more considerable because the sender can calculate an optimal source rate from a wider range, not limited by any of the two channels. That is to say, our proposed system is more proactive in allocating part of the available channel rate for error control since it can lead to improvement in the overall video quality. Therefore, the resources used by source coding and error control are globally optimized in a joint and distributed way to achieve the best user-perceived quality. This conclusion is also supported by the simulation

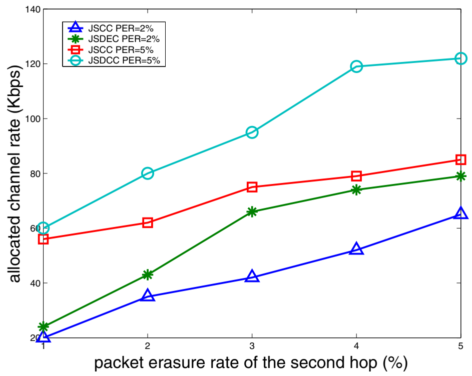


Fig. 4. Frame allocated channel coding rate comparison between the proposed JSDEC system and the JSCC system with constant channel rates. ($W_{T1} = W_{T2} = 0.3MHz$)

results that we will provide in the following paragraphs.

Similarly, in Fig. 3, W_{T2} is set to $0.6MHz$, while W_{T1} varies from $0.2MHz$ to $0.4MHz$ at a step size of $50KHz$. In this figure, the performance gain achieved by using JSDEC over JSCC is much lower compared with that shown in Fig. 2(a). This is because in Fig. 2(b) the available channel bandwidth of the channel of the second hop is always much higher than that of the first one. Thus, there is always enough rate on the channel of the second hop to be allocated for error control, which mitigates channel distortion.

To examine the system performance under different parameters, we then present in Fig. 3(a) and Fig. 3(b) experimental results for a configuration where the packet loss rates vary within the range of $1-5\%$, while the channel bandwidths are kept constant at $0.3MHz$. Similarly, we also assume that the two links in tandem are attached through a single proxy. In both figures, we set the average PER of the first hop ϵ_1 to 2% and 5%, respectively, while ϵ_2 changes from 1% to 5%.

Based on the results shown in these two figures, we can conclude that if either ϵ_1 or ϵ_2 is low, the performance does not differ much. To explain, low PER corresponds to few packet losses and thus little channel distortion. However, when either of these packet erasure rates increase, channel distortion will also increase. More interestingly, when the discrepancies in the packet erasure rates are significant, and especially under high PER, the performance gain achieved by using the proposed system is higher when compared with the JSCC system. This means that it is more crucial to use JSDEC when the two hops undergo significant difference in link quality, which also explains that the proposed system is particularly useful for multi-hop scenarios.

In Fig. 4 we present results for the case that W_{T1} and W_{T2} are both set to $0.3MHz$. We measure the bit rates allocated for error control under the JSDEC and JSCC systems when ϵ_1=2% and ϵ_1=5% on average, respectively, while ϵ_2 changes from 1% to 5% at a step size of 1% for both cases. It can be observed from the figure that with the increase of packet erasure rate, the resources allocated for error control always increase. More importantly, even under the same PER value, the proposed JSDEC always allocates more resources for error control, which explains the performance gains observed in Fig. 2. However, the JSCC system is unable to account for these situations on the second hop, and as a result it can not select a source coding rate that enables the optimal error control on the second hop.

VI. Conclusions

In this paper, we have presented a system for joint error resilient source coding and distributed application-layer error control (JSDEC) suitable for video streaming in wireless multi-hop networks. The proposed system employs distributed error control by using a lightweight application-layer ARQ scheme residing in the intermediate nodes. We have demonstrated that in order to achieve globally optimal joint source and channel coding decisions for packet video transmission over wireless multi-hop networks, the end-to-end video distortion estimate must consider the contribution of each communications link. We have derived a model for a path that consists of multiple packet erasure channels connected in tandem. The formulated problem has then been solved based on dynamic programming, incurring moderate complexity at the streaming server. Experiments carried out with NS-2 and H.264/AVC have validated the efficacy of the proposed scheme under different channel conditions.

Acknowledgements

This work was partially supported by NSF under Grant No. CCF-0830493 and CNS-0707944. It was also partially supported by Layman Foundation.

References

[1] M. van der Schaar and P. Chou, *Multimedia over IP and Wireless Networks*. Academic Press, 2007.

[2] A. Argyriou, "Distributed resource allocation for network-supported FGS video streaming," in *Packet Video Workshop*, Lausanne, Switzerland, November 2007.

[3] J. Yan, K. Katrinis, M. May, and B. Plattner, "Media- and TCP-friendly congestion control for scalable video streams," *IEEE Transactions on Multimedia*, vol. 8, no. 2, April 2006.

[4] L. Gao, Z.-L. Zhang, and D. Towsley, "Proxy-assisted techniques for delivering continuous multimedia streams," *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, pp. 884–894, December 2003.

[5] L. Kondi and A. Katsaggelos, "Joint source-channel coding for scalable video using models of rate-distortion functions," in *ICASSP*, 2001.

[6] F. Zhai, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Rate-distortion optimized hybrid error control for real-time packetized video transmission," *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 40–53, January 2006.

[7] Z. He, J. Cai, and C. W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 511–523, June 2002.

[8] M. Westerlund and S. Wenger, "RTP Topologies," draft-ietf-avt-topologies-05.txt, July 2007.

[9] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *Microsoft Research Technical Report MSR-TR-2001-35*, 2001.

[10] A. Mukherjee, "On the dynamics and significance of low frequency components of Internet load," University of Pennsylvania, Tech. Rep. MS-CIS-92-83, Tech. Rep., 1992.

[11] "JVT reference software," Available from http://bs.hhi.de/ suehring/tml/download/.