

# Multihop Rate Adaptive Wireless Scalable Video Using Syndrome-Based Partial Decoding

Yongju Cho, Hayder Radha, Jeongil Seo, Jungwon Kang, and Jinwoo Hong

The overall channel capacity of a multihop wireless path drops progressively over each hop due to the cascading effect of noise and interference. Hence, without optimal rate adaptation, the video quality is expected to degrade significantly at any client located at a far-edge of an ad-hoc network. To overcome this limitation, decoding and forwarding (DF), which fully decodes codewords at each intermediate node, can be employed to provide the best video quality. However, complexity and memory usage for DF are significantly high. Consequently, we propose syndrome-based partial decoding (SPD). In the SPD framework an intermediate node partially decodes a codeword and relays the packet along with its syndromes if the packet is corrupted. We demonstrate the efficacy of the proposed scheme by simulations using actual 802.11b wireless traces. The trace-driven simulations show that the proposed SPD framework, which reduces the overall processing requirements of intermediate nodes, provides reasonably high goodput when compared to simple forwarding and less complexity and memory requirements when compared to DF.

**Keywords:** Multihop wireless channel, scalable video, forward error correction (FEC).

Manuscript received Sept. 15, 2009; revised Feb. 4, 2010; accepted Feb. 10, 2010.

This work was supported by the IT R&D program of MKE/KCC/KEIT [KI002070, Development of Interactive View Control Technologies for IPTV].

Yongju Cho (phone: +82 42 860 6874, email: yongjucho@etri.re.kr), Jeongil Seo (email: seoji@etri.re.kr), and Jungwon Kang (email: jungwon@etri.re.kr) are with the Broadcasting & Telecommunications Convergence Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Hayder Radha (email: radha@egr.msu.edu) is with the Department of Electrical & Computer Engineering, Michigan State University, East Lansing, MI, USA.

Jinwoo Hong (email: jwhong@etri.re.kr) is with the Creative & Challenging Research Division, ETRI, Daejeon, Rep. of Korea.

doi:10.4218/etrij.10.1409.0072

## I. Introduction

In many wireless environments, deteriorated link conditions cause bit-corruptions within transmitted packets. These corrupted packets cause checksum failures and packet drops at wireless receivers. Thus, rate-adaptive video applications can benefit greatly from accurate channel capacity estimation, which can be employed to reduce the number of packet drops and hence provide improved quality of service.

To that end, many recent efforts incorporate cross-layer protocols that fully utilize corrupted packets to accurately estimate channel capacity and to improve throughput [1]-[8]. Among recent cross-layer efforts, one class of wireless multimedia protocols of particular interest is cross-layer-design with side-information (CLDS) [6] protocols, which relay corrupted packets with their side-information<sup>1)</sup> to a higher layer for further processing. CLDS protocols differ significantly from conventional (CON) [6] protocols, which drop any packet that has one or more residue errors.<sup>2)</sup> Prior studies [1]-[3], [6], [9] have shown that a very accurate channel capacity estimation, and hence significant improvement in wireless video throughput, can be achieved by CLDS. Therefore, we employ CLDS protocols in this study.

Some studies [10], [11] have focused on wireless multimedia communication over ad-hoc networks, which consist of multiple wireless hops from a server to a client. Each wireless channel between intermediate nodes suffers from impairment due to interference, fading, and multi-path effects. Hence, when a

<sup>1)</sup> The side information includes signal-to-silence ratio (SSR) indicators and MAC-layer checksum for packets, both of which can be used as parameters for channel estimation [7]. (SSR is a packet-level SNR parameter supported by 802.11 compliant devices).

<sup>2)</sup> Here, a residue error is an error that is not corrected by the physical layer; hence, it appears at the media access control (MAC) layer.

multihop wireless network is employed, channel capacity decreases over each hop; hence, the end-to-end (E2E) channel capacity can become very low. This leads to significant degradation of goodput<sup>3)</sup> and video quality for rate adaptation applications.

An ideal workaround to this problem is to employ decoding and forwarding (DF). Under DF, channel coding is employed over every hop of the end-to-end path. Hence, an intermediate node decodes the transmitted packets (or codewords) to suppress noise on the channel between two intermediate nodes and re-encodes the packets, potentially with a different codebook, for transmission towards the destination [12]. Thus, DF can achieve optimal performance with regard to network capacity. However, complexity and memory usage for DF are significantly high. Moreover, intermediate nodes, which may be participating by only forwarding the content toward a receiver further-down a multihop chain, do not have much incentive to perform full decoding-encoding of the channel-coded wireless video content. For such nodes, we need to minimize their burden in terms of the operation they need to perform toward the delivery of the video content to the final receiver.

These issues motivate the usage of a *partial processing framework* for rate-adaptive wireless video, which reduces the overall processing requirements of intermediate nodes. In this paper, we propose syndrome-based partial decoding (SPD) architecture. The proposed architecture employs CLDS protocols, which, among other things, accurately estimate the

channel capacity using a simple binary symmetric channel (BSC) [14] model. The two main contributions of this paper are the following:

- Syndrome-based partial decoding scheme: Under this scheme, each intermediate node only computes the syndromes of the received packet (that is, partial decoding) and relays the syndromes along with the packet (if the packet is corrupted) to the next hop. The client then fully decodes the packet by using the syndromes which are appended to the packet.
- Optimal packet size selection scheme: As explained later, SPD utilizes the channel packet error rate (PER) as an error parameter for its operation; hence, SPD goodput heavily depends on the size of packets. Thus, we derive the optimal packet size selection scheme to maximize the goodput in the proposed architecture.

The proposed SPD architecture is tested using a comprehensive set of wireless residual error traces collected at 2 Mbps, 5.5 Mbps, and 11 Mbps physical data rates of an operational 802.11b network. We compare the performance of SPD with DF, E2E decoding, and automatic repeat request (ARQ). We show that SPD outperforms E2E decoding and ARQ and provides relatively good goodput compared to that provided by DF.

The rest of this paper is organized as follows. Section II describes our wireless trace collection setup and then presents a preliminary analysis of the collected data. Section III presents the motivations for this study and develops the proposed SPD. Sections IV and V respectively evaluate the performance of the proposed framework and summarize key conclusions of this work.

## II. Collection and Empirical Analysis of Residual Wireless Traces

### 1. Data Collection

For this study, five wireless receivers were used to simultaneously collect error traces on an 802.11b LAN. The receivers were placed at various locations in a room, while the access point (AP) was placed in a room across a hallway from the receivers to simulate a realistic home/classroom/office setting. The receivers' media access control (MAC) layer device drivers were modified to pass corrupted packets to higher layers. To capture packets at high transmission rates, packet dissectors were implemented inside the device drivers. These packet dissectors ensured that only packets pertinent to our wireless experiment were processed, while all other packets were dropped. Each experiment processed one million packets with a payload of 1,000 bytes each. That is, each trace

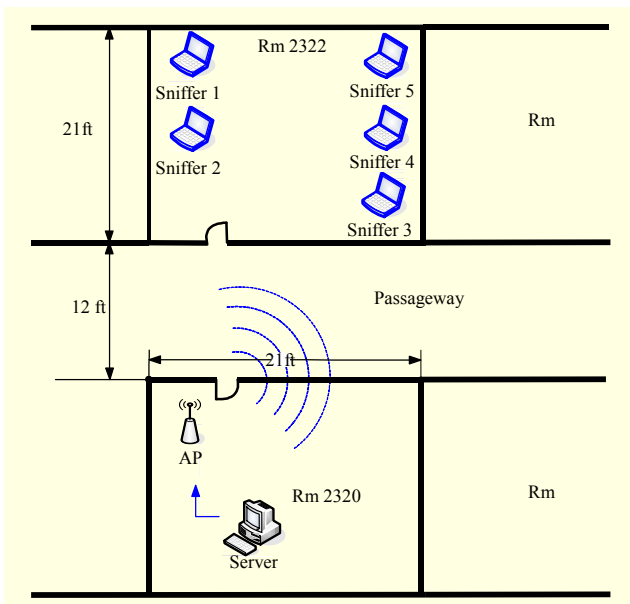


Fig. 1. Topologies used for wireless trace collection [13].

3) In this study the goodput is defined as the application level throughput, that is, the number of information bits per total number of bits forwarded by the network from a certain source address to a certain destination.

Table 1. Statistics of traces used in this study.

| Phy. data rate (Mbps) | Avg. PER (%) | Min. PER (%) | Max. PER (%) | Avg. BER (%) | Min. BER (%) | Max. BER (%) |
|-----------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| 2                     | 5.97         | 0.75         | 14.31        | 0.37         | 0.024        | 1.0          |
| 5.5                   | 9.79         | 0.61         | 22.74        | 0.6          | 0.024        | 1.31         |
| 11                    | 39.5         | 10.99        | 77.83        | 1.06         | 0.16         | 3.03         |

had approximately 1 GB of data.

A wired sender was used to send multicast packets with a predetermined payload on the wireless LAN. The use of multicasting disabled the MAC layer retransmissions. In addition to a packet's header and payload information, we logged the signal-to-silence ratio (SSR) for each packet. A packet's SSR is a one-byte number between 0 dB and 100 dB, representing an approximate measure of the SNR at which the packet was received. The sender used different transmission rates ranging from 500 kbps to 1 Mbps for each experiment. At the physical layer, the auto-rate-selection feature of the AP was disabled, and for each experiment, the AP was forced to transmit at a fixed data rate. Each trace collection experiment was repeated multiple times at 2 Mbps, 5.5 Mbps, and 11 Mbps physical layer data rates and at different times of day.

## 2. Average Statistics of the Traces

Table 1 provides some statistics of the traces collected for this study. Since the physical layer robustness decreases with an increase in data rate, the average packet error rate increases with an increase in the physical layer data rate. In particular, the average packet error rate increases from approximately 10% at 5.5 Mbps to almost 40% at 11 Mbps. Since the wireless receivers were placed at various locations, the receivers experienced different packet error rates. The overall minimum and maximum error rates in Table 1 show that the receivers under consideration were experiencing both good and bad link conditions.

For a detailed study of BER behavior at different SSR values, we refer the reader to section 2 in [13].

## III. Syndrome-Based Partial Decoding

In this section, we develop the rate adaptation architecture over an ad-hoc network using the two main contributions of this paper: SPD and optimal packet size selection. SPD is achieved by the following 3 steps. First, it only calculates a syndrome for a packet at each intermediate node. Second, it forwards a codeword for the syndrome of the packet (only

when the packet is corrupted at a hop) to the next hop. Third, it fully decodes the packet by using its syndromes at a client. SPD embodies BER as well as PER. That is, the goodput is a function of both BER and PER. Thus, the optimal goodput is achieved by finding an optimal packet size that provides the best goodput in the proposed architecture.

### 1. Architecture for Rate-Adaptation

The architecture of a multimedia streaming application depends heavily on the network over which it operates. Therefore, in this section, we define the proposed architecture for rate adaptation. Additionally, we outline the assumptions under which the proposed architecture is to be evaluated.

The proposed architecture, shown in Fig. 2, consists of a server, intermediate nodes, and a client that receives packets over a multihop wireless network. In the proposed architecture, a server, intermediate nodes, and a client are designed to support source and channel rate adaptation. The intermediate nodes and client support CLDS protocols that leverage residue-error-process and side information, which can be relayed to an FEC decoder for partial decoding at each intermediate nodes or full decoding at a client [15], to estimate the current channel capacity<sup>4)</sup> for a block of packets (or a rate adaptation period). The current channel capacity, which is estimated by the channel estimator with the entropy of the residue error process, is then transmitted to the server as feedback for rate adaptation. Using the feedback, the rate tuner at the server predicts the optimal source and channel rates for the next block of multimedia packets to be transmitted [14].

For this study, we focus on the performance of SPD; therefore, we assume that the system employs generic (arguably ideal) channel and source coding schemes. In particular, we consider the following simplifying assumptions. First, the channel code achieves the capacity (an ideal channel coder), and a block of packets can be successfully decoded at the client if the total rate source and channel coding rate does not exceed channel capacity, that is,  $R \leq C$  [16]. Note that rate  $R$  is predicted using (8) in [9]. Second, a video encoder provides a bit-stream having a bitrate that is exactly the same as the required bitrate, and the bit-stream renders the peak-to-peak signal-to-noise ratio (PSNR) value [17] according to the bitrate. With these two assumptions, we realize an architecture where, for video rates that cannot be supported by the underlying channel capacity, the video quality reduces to zero (zero PSNR value). Note that without these assumptions (that is, if, a

4) Here, the channel capacity is estimated as  $C_n = 1 - \frac{1}{m} \sum_{i=1}^m H_b(\epsilon_i)$ , where  $\epsilon_i$  represents the BER estimate for packet  $i$ ;  $\frac{1}{m} \sum_{i=1}^m H_b(\epsilon_i)$  is the instantaneous per-packet error-process entropy in a block of  $m$  packets; and  $n$  is a rate adaptation period [9].

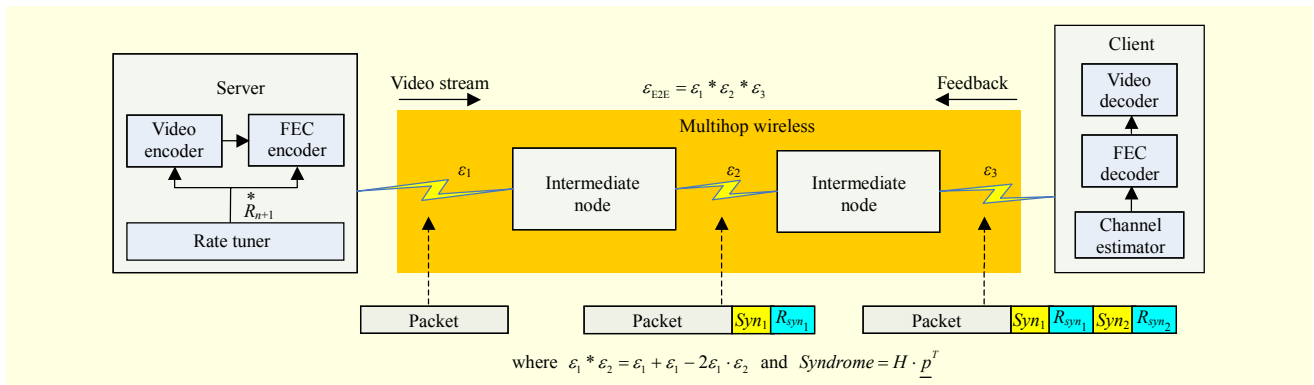


Fig. 2. Architecture of the proposed rate adaptation.

realistic channel coder or video encoder was used), we should carefully take the performance of the channel coder and video encoder into consideration in finding the rate. Therefore, these assumptions were made to focus only on the performance of the proposed SPD scheme.

## 2. SPD Overview

Each intermediate node calculates the packet syndromes ( $Syndrome_{rx1} = Check\_matrix_{n \times r} \cdot Packet_{1 \times n}^T$ ) from the server or the previous intermediate node and forwards the packet with the packet's syndromes (including the redundancy of the syndrome; that is, the codeword for the syndrome) only if the packet is corrupted. Note that when a packet is not corrupted (the syndromes consist of all 0s) only the packet is relayed to the next hop. Meanwhile, when a packet is corrupted, each node adds its syndrome, including the syndrome redundancy, to the packet in an embedded manner; hence, this process is repeated progressively until a client receives the packet. The client then uses the syndromes, which are coded and appended to the packet, to decode the packet. For example, under the scenario shown in Fig. 2, when a packet is corrupted at the first and second hops, the client receives the packet and two different syndromes appended to the packet. Note that a syndrome is used as the information needed to successfully decode the packet which is corrupted in a hop (or a channel link).

Note that a server uses the maximum channel BER ( $\epsilon_{max} = \max(\epsilon_1, \epsilon_2, \epsilon_3)$ ) to select the rates rather than the E2E BER ( $\epsilon_{E2E} \geq \epsilon_{max}$ ) (see Fig. 2) since each syndrome can be used to correct errors that occur at each hop. This leads SPD to suppress noise on each channel between a server and a client and to provide goodput performance close to that of DF.

Using Fig. 2 as an illustrative example, the packet received at a client is then decoded along with its syndromes as follows:

**Step 1.** Decode the codewords for the syndromes that are forwarded from the previous nodes.

**Step 2.** Decode the codeword for the message with each of

the decoded codewords for the syndromes.

Note that the above procedure is applicable to Hamming codes, and a similar procedure can be applied to Reed Solomon (RS) codes, for which the syndrome polynomial can be employed.

## 3. Goodput Evaluation

In this section, we develop the expression for the goodput of the proposed SPD and other schemes such as DF, end node decoding, and ARQ. As described in the previous subsection, for SPD, the total number of bits transmitted to a multihop wireless network increases as the PER and the number of hops increase. Therefore, the per-hop goodput for SPD,  $Goodput_{SPD}$ , can be expressed as

$$\begin{aligned}
 Goodput_{SPD} &= \frac{k}{n_{total}} = \frac{k}{\left[ \frac{(k+r) \cdot h + \sum_{i=1}^h (PER_i \cdot (k_{s_i} + r_{s_i}) \cdot h)}{h} \right]} \\
 &= \frac{k}{(k+r) + \sum_{i=1}^h (PER_i \cdot (k_{s_i} + r_{s_i}))} \\
 &= \frac{k}{\left( k + \frac{k \cdot H(\epsilon_{max})}{1 - H(\epsilon_{max})} \right) + \sum_{i=1}^h PER_i \cdot \left( r + \frac{r \cdot H(\epsilon_{max})}{1 - H(\epsilon_{max})} \right)} \\
 &= \frac{k}{k \cdot \left( 1 + \frac{H(\epsilon_{max})}{1 - H(\epsilon_{max})} \right) + \sum_{i=1}^h PER_i \cdot r \cdot \left( 1 + \frac{H(\epsilon_{max})}{1 - H(\epsilon_{max})} \right)} \\
 &= \frac{k}{\left[ k + \sum_{i=1}^h PER_i \cdot r \right] \cdot \Delta_{max}} = \frac{k}{\left[ k + \sum_{i=1}^h PER_i \cdot \frac{k \cdot H(\epsilon_{max})}{1 - H(\epsilon_{max})} \right] \cdot \Delta_{max}} \\
 &= \frac{1}{\left[ 1 + \sum_{i=1}^h PER_i \cdot \frac{H(\epsilon_{max})}{1 - H(\epsilon_{max})} \right] \cdot \Delta_{max}}, \tag{1}
 \end{aligned}$$

where  $k$  is the number of information (original source) bits,  $r$  is the number of redundant bits for  $k$ ,  $n(=k+r)$  is the total number of bits in a packet length,  $k_s$  is the number of syndrome (bits) of the packet,  $r_s$  is the number of redundant bits for the syndrome,  $h$  represents the number of hops from a server to a client, and

$$\Delta_{\max} = 1 + \frac{H(\varepsilon_{\max})}{1 - H(\varepsilon_{\max})}. \quad \text{Note that}$$

$$n_{\text{total}} = \frac{(k+r) \cdot h + \sum_{i=1}^h \left( PER_i \cdot (k_{s_i} + r_{s_i}) \cdot h \right)}{h},$$

which normalizes the total number of bits by defining the per-hop transmission cost; the length of  $k_s$  is the same as that of  $r$ , and  $r_{s_i} = \frac{k_{s_i} \cdot H(\varepsilon_{\max})}{C} = \frac{r \cdot H(\varepsilon_{\max})}{C} = \frac{r \cdot H(\varepsilon_{\max})}{1 - H(\varepsilon_{\max})}$ .

DF corrects error bits (full decoding) and re-encoding at every intermediate node; therefore, the goodput for DF,  $Goodput_{DF}$ , can be expressed as

$$Goodput_{DF} = \frac{k}{n_{\text{total}}} = \frac{k}{n}$$

$$= \frac{k}{\left( k + \frac{k \cdot H(\varepsilon_{\max})}{1 - H(\varepsilon_{\max})} \right)} = \frac{k}{k \cdot \left( 1 + \frac{H(\varepsilon_{\max})}{1 - H(\varepsilon_{\max})} \right)}$$

$$= \frac{k}{k \cdot \Delta_{\max}} = \frac{1}{\Delta_{\max}}, \quad (2)$$

where  $n_{\text{total}} = (n \cdot h) / h$ , normalizing the total number of bits by defining the per-hop transmission cost.

End node decoding, which we defined in this study, employs CLDS protocols but does not incorporate a partial or full decoding at any intermediate node and uses the E2E channel BER,  $\varepsilon_{E2E}$ , to find a source and channel coding rate. Thus, the goodput for end node decoding,  $Goodput_{END}$ , can be expressed as

$$Goodput_{END} = \frac{k}{n_{\text{total}}} = \frac{k}{(k+r)}$$

$$= \frac{k}{\left( k + \frac{k \cdot H(\varepsilon_{E2E})}{1 - H(\varepsilon_{E2E})} \right)} = \frac{k}{k \cdot \left( 1 + \frac{H(\varepsilon_{E2E})}{1 - H(\varepsilon_{E2E})} \right)}$$

$$= \frac{k}{k \cdot \Delta_{E2E}} = \frac{1}{\Delta_{E2E}}, \quad (3)$$

where  $\varepsilon_{E2E} = \varepsilon_1 * \varepsilon_2 * \dots * \varepsilon_n$ , and  $\Delta_{E2E} = 1 + \frac{H(\varepsilon_{E2E})}{1 - H(\varepsilon_{E2E})}$ .

Note that the BER for two cascaded BSCs is  $\varepsilon_1 * \varepsilon_2 = \varepsilon_1 + \varepsilon_2 - 2\varepsilon_1 \cdot \varepsilon_2$ .

For ARQ, a packet is re-transmitted until a defined time-out by a server if it is corrupted during its transmission from a server to a client. Therefore, the goodput for ARQ can be

expressed as

$$Goodput_{ARQ} = 1 - PER_{E2E}, \quad (4)$$

where  $PER_{E2E} = PER_1 * PER_2 * \dots * PER_n$ .

#### 4. Optimization of SPD

PER changes with respect to the size of a packet for a given channel BER, and  $Goodput_{SPD}$  is a function of BER and PER. Therefore, we capture this aspect to find the optimal packet size,  $pkt\_size$ , which maximizes  $Goodput_{SPD}$  as in [18] as

$$pkt\_size^* = \arg \max_{pkt\_size} Goodput_{SPD}$$

$$= \arg \max_{pkt\_size} \frac{1}{\left[ 1 + \sum_{i=1}^h PER_i \cdot \frac{H(\varepsilon_{\max})}{1 - H(\varepsilon_{\max})} \right]} \cdot \Delta_{\max}, \quad (5)$$

where  $l$  is the packet header<sup>5)</sup> (in bits). Note that for the goodput developed in the previous section, a packet header is not considered. Therefore, we consider the packet header for a realistic simulation. In addition, it is necessary to model the PER as a function of packet sizes to fully utilize (5). Consequently, we develop an empirical PER model to express the PER for packet sizes. To deduce the empirical model, we conduct the following measurement procedure:

**Step 1.** Calculate the BER of the underlying channel, that is, the BER of a collected trace.

**Step 2.** Calculate PER with respect to packet size.

**Step 3.** Repeat steps 1 and 2 with different traces.

From a comprehensive set of measurements, it can be observed that PER linearly increases as the packet size increases (see Fig. 3). Consequently, we derive the empirical model to express the PER for packet sizes as

$$PER = ax + b, \quad (6)$$

where  $a = 1.071 \times 10^{-5}$ ,  $b = 0.01597$ , and  $x$  is packet size. Note that  $a$  and  $b$  were found by linear fitting.

We leverage this empirical PER model to optimally select the packet size which maximizes  $Goodput_{SPD}$ . The performance of SPD, which fully employs this model, is described in the following section.

#### IV. Performance Evaluation

We now compare the performance of the proposed SPD

<sup>5)</sup> Header refers to supplemental data placed at the beginning of a block of data being stored or transmitted, which contain information for the handling of the data block. We use 56 bytes (28 bytes for MAC, 20 for IP, and 8 for UDP header) as a size of packet header.

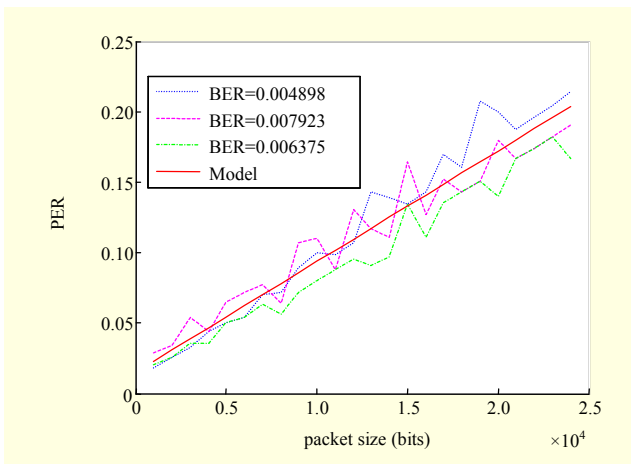


Fig. 3. Empirical PER model.

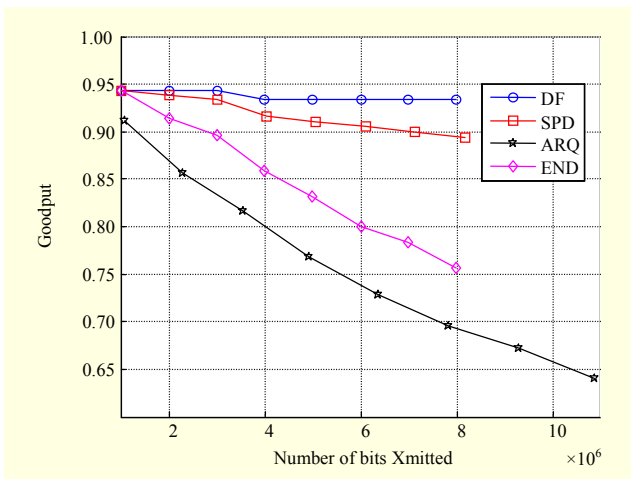


Fig. 4. Goodput comparison of four schemes in terms of per-hop transmitted bits.

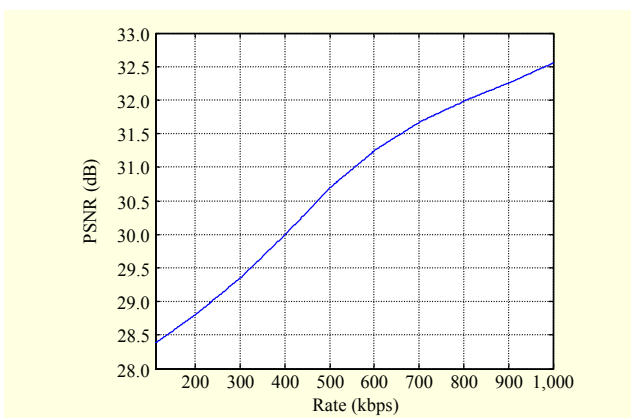


Fig. 5. RD (quality) function of SVC test video sequence [13].

with DF, end node decoding, and ARQ in terms of goodput and the total number of bits transmitted in an ad-hoc network. To compare the performance of the schemes, the experiments

Table 2. Performance comparison of schemes in terms of goodput (125 packets and each packet consists of 8,000 bits).

| Scheme |                      | Number of hops |       |       |       |       |
|--------|----------------------|----------------|-------|-------|-------|-------|
|        |                      | 1              | 2     | 3     | 4     | 5     |
| DF     | Goodput              | 0.944          | 0.944 | 0.944 | 0.933 | 0.933 |
|        | Xmitted bits (kbits) | 1,000          | 2,000 | 3,000 | 4,000 | 5,000 |
| SPD    | Goodput              | 0.944          | 0.939 | 0.934 | 0.916 | 0.911 |
|        | Xmitted bits (kbits) | 1,000          | 2,005 | 3,015 | 4,037 | 5,062 |
| END    | Goodput              | 0.944          | 0.913 | 0.897 | 0.858 | 0.832 |
|        | Xmitted bits (kbits) | 1,000          | 2,000 | 3,000 | 4,000 | 5,000 |
| ARQ    | Goodput              | 0.912          | 0.856 | 0.816 | 0.768 | 0.728 |
|        | Xmitted bits (kbits) | 1,088          | 2,288 | 3,552 | 4,928 | 6,360 |

Table 3. Performance of SPD with optimal packet size (Opt. pkt. siz) selection scheme in terms of goodput (Xmit rate=500 kbps).

| No. of hops | Actual channel      |         |           | Optimization        |         |           |
|-------------|---------------------|---------|-----------|---------------------|---------|-----------|
|             | Opt.pkt. siz (bits) | Goodput | PSNR (dB) | Opt.pkt. siz (bits) | Goodput | PSNR (dB) |
| 2           | 28,555              | 0.9126  | 30.38     | 26,700              | 0.9124  | 30.38     |
| 3           | 18,806              | 0.8999  | 30.34     | 19,100              | 0.8960  | 30.32     |
| 4           | 14,487              | 0.8760  | 30.26     | 14,456              | 0.8759  | 30.25     |
| 5           | 13,754              | 0.8669  | 30.23     | 12,649              | 0.8646  | 30.21     |

in this study were conducted as follows:

**Step 1.** Calculate the BER and PER of each channel (or hop) for a rate adaptation period. Note that we use 2 Mbps traces for this simulation and treat each trace as the channel between two nodes in the network.

**Step 2.** Find the source and channel coding rate, for a rate adaptation period. Note that we assume the use of an ideal channel code; hence, we use the capacity as the rate.

**Step 3.** Calculate the goodput for SPD, DF, end node decoding, and ARQ by using (1), (2), (3), and (4).

**Step 4.** Find the optimal packet size by using (5) which incorporates the empirical PER model (see (6)) and the corresponding goodput.

**Step 5.** Calculate the PSNR using the rate-distortion (RD) function of a scalable video coding (SVC) sequence,  $Q(\cdot)$  (see Fig. 5) [19], for the given goodput as  $PSNR=Q(Goodput_{SPD} \cdot T)$  where  $T$  represents a transmission (Xmit) rate in bits-per-second.

As for goodput, Fig. 4 and Table 2 show that SPD outperforms end node decoding and ARQ and its performance is a little inferior to that of DF over hops. However, note that

although DF provides the best performance in terms of goodput, the complexity and memory usage at each intermediate node are significantly high since each intermediate node performs full decoding and re-encoding of packets. Additionally, the results shown in Fig. 4 and Table 2 do not employ the optimal packet size selection scheme.

It is also observed in Table 2 that the total number of bits transmitted by all nodes in the network for SPD is a little larger than those of the DF and end node decoding schemes. However, the difference is very small and does not significantly impact the goodput of SPD.

The excellent performance of the proposed optimal packet size selection scheme for SPD can be seen in Table 3. The first column of Table 3 represents the number of hops. The second, third, and fourth columns represent the optimal packet size, goodput, and PSNR achieved by SPD under the actual channel, that is, the collected traces used for simulations in this study. The fifth column represents the optimal packet size estimated by (5) and (6). The sixth and seventh columns respectively represent the goodput and PSNR with the optimally estimated packet size under actual channel. As shown in Table 3, the optimal packet size selection scheme very accurately estimates the packet sizes that maximize both goodput and PSNR. In fact, they are very close to the optimum.

## V. Conclusion

In this paper, we proposed the new partial processing SPD framework for multihop rate-adaptive wireless video. From our realistic experiments, SPD reduces the overall processing requirements of intermediate nodes and provides reasonably high goodput compared to end node decoding and ARQ. Also, SPD provides less goodput than DF. However, the performance difference between DF and SPD is relatively small if we take complexity and memory requirements into consideration. Furthermore, if SPD incorporates entropy coding, such as Hoffman or run-length coding, for syndromes at each intermediate node, the goodput will be further improved.

## References

- [1] L. Larzon, M. Degermark, and S. Pink, "UDP Lite for Real Time Multimedia Applications," *IEEE Int. Conf. Communications (ICC)*, Vancouver, June 1999.
- [2] A. Singh, A. Konrad, and A.D. Joseph, "Performance Evaluation of UDP Lite for Cellular Video," *NOSSDAV*, 2001.
- [3] S.A. Khayam et al., "Performance Analysis and Modeling of Errors and Losses over 802.11b LANS for High-Bitrate Real-Time Multimedia," *Signal Processing: Image Communication*, vol. 18, no. 47, Aug. 2003, pp. 575-595.
- [4] F. Pauchet et al., "Conditions for SVC Bit Error Resilience Testing," Joint Video Team (JVT) 17th Meeting: Nice, France, Oct. 2005.
- [5] R. Riemann and K. Winstein, "Improving 802.11 Range with Forward Error Correction," Massachusetts Institute of Technology, Tech. Rep, Feb. 2005.
- [6] S. Karande and H. Radha, "Hybrid Erasure-Error Protocols for Wireless Video," *IEEE Trans. Multimedia*, vol. 9, no. 2, Feb. 2007, pp. 307-319.
- [7] S. Karande et al., "Utilizing Signal to Silence Ratio indications for improved Video Communication in presence of 802.11b Residue Errors," *ICME*, July 2006.
- [8] S. Karande, K. Misra, and H. Radha, "Survival of the Fittest: An Active Queue Management Technique For Noisy Packet Flows," *Journal of Advances in Multimedia - Special Issue on Multimedia Networking*, vol. 2007, no. 1, Jan. 2007.
- [9] Y. Cho et al., "On Channel Capacity Estimation and Prediction for Rate Adaptive Wireless Video," *IEEE Trans. Multimedia*, vol. 10, Nov. 2008, pp. 1419-1426.
- [10] Y.C. Hu and D.B. Johnson, "Design and Demonstration of Live Audio and Video over Multihop Wireless ad hoc Networks," *Proc. MILCOM*, 2002.
- [11] C.M. Cordeiro, H. Gossain, and D.P. Agrawal, "Multicast over Wireless Mobile ad hoc Networks: Present and Future Directions," *IEEE Network*, vol. 17, no. 1, 2003, pp. 52-59.
- [12] J.N. Laneman and G.W. Wornell, "An Efficient Protocol for Realizing Distributed Spatial Diversity in Wireless Ad-Hoc Networks," *ARL FedLab Symp. Advanced Telecommunications and Information Distribution (ATIRP-2001)*, College Park, MD, Mar. 2001.
- [13] Y. Cho et al., "A Multi-Tier Model for BER Prediction over Wireless Residual Channels," *CISS*, Mar. 2007.
- [14] Y. Cho et al., "A Rate-Distortion Empirical Model for Rate Adaptive Wireless Scalable Video," *CISS*, Mar. 2008.
- [15] S. Lin and D.J. Costello Jr., *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, NJ: Prentice Hall, 1983.
- [16] R.W. Yeung, *A First Course in Information Theory*, Kluwer Academic/Plenum Publishers, 2002.
- [17] P. Lambert et al., "Rate-Distortion Performance of H.264/AVC Compared to State-of-the-Art Video Codecs," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 16, no. 1, Jan. 2006, pp. 134-140.
- [18] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2006.
- [19] ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, *Advanced Video Coding for Generic Audiovisual Services*, version 8.9, 2007.



**Yongju Cho** received the BS and MS degrees from Iowa State University, Ames, in 1997 and 1999, respectively, and the PhD degree from Michigan State University in 2009, all in electrical and computer engineering. He joined ETRI, Daejeon, Korea, in 2001, and has been engaged in the development of a data

broadcasting system based upon MPEG-2/4 systems, IPMP and in the MPEG-21 area. His research interests include digital signal processing in the field of data broadcasting and adaptive wireless video communications.



**Hayder Radha** (F'09) received the BS degree (with honors) from Michigan State University (MSU) in 1984, the MS degree from Purdue University, West Lafayette, IN, in 1986, and the PhM and PhD degrees from Columbia University, New York, in 1991 and 1993, respectively, all in electrical

engineering. Currently, he is a professor of electrical and computer engineering (ECE) at MSU, the associate chair for Research and Graduate Studies of the ECE Department, and the director of the Wireless and Video Communications Laboratory at MSU. He was with Philips Research (1996–2000), where he worked as a principal member of the research staff and then as a consulting scientist in the Video Communications Research Department. He became a Philips Research Fellow in 2000. He was a distinguished member of the technical staff at Bell Laboratories, where he worked between 1986 and 1996 in the areas of digital communications, image processing, and broadband multimedia. His current research areas include wireless communications and networking, sensor networks, network coding, video coding, stochastic modeling of communication networks, and image and video processing. He has more than 100 peer-reviewed papers and 30 U.S. patents in these areas.



**Jeongil Seo** received the MS degree and PhD degree in electronics from Kyoungpook National University (KNU), Daegu, Korea, in 1996 and 2005, respectively. He worked as a senior member of engineering staff at the Laboratory of Semiconductor, LG-Semicon, Cheongju, Korea, from 1998 until 2000. He has

worked as a senior member of research staff with the Broadcasting Media Research Department, ETRI, Daejeon, Korea, since 2000. His research activities include digital audio processing, room acoustics, real-time audio codec systems, and interactive 3D audio broadcasting systems.



**Jung Won Kang** received the BS and MS degrees in electrical engineering in 1993 and 1995, respectively, from the Korea Aerospace University, Seoul, South Korea. She received the PhD in electrical and computer engineering in 2003 from Georgia Institute of Technology, Atlanta, GA. Since 2003, she has been a senior

member of research staff in the Broadcasting & Telecommunications Media Research Division of ETRI, Korea. Her research interests are in the areas of video signal processing, video coding, and video adaptation.



**Jin Woo Hong** received the BS and MS degrees in electronic engineering from Kwangwoon University, Seoul, Korea, in 1982 and 1984, respectively. He also received the PhD in computer engineering from the same university in 1993. Since 1984, he has been with ETRI, Daejeon, Korea, as a principal member

of engineering staff, where he is currently a director of the Broadcasting & Telecommunications Convergence Future Technology Research Department. From 1998 to 1999, he conducted research at Fraunhofer Institute in Erlangen, Germany, as a visiting researcher. His research interests include multimedia framework technology, broadcasting media and service, personal broadcasting, and realistic media.