

RESEARCH

Open Access

Modulation level allocation for MGS streaming over a multihop wireless channel

Daeyeon Kim^{1*}, Takeo Fujii¹ and Kyesan Lee²**Abstract**

This article introduces a method for efficiently transmitting medium grain scalable video packets over a transmission path consisting of multiple wireless links. Medium grain scalability provides bit rate adaptation according to the available bit rate by dropping a number of video packets in the compressed bit stream. In other words, rate-distortion control can be achieved by means of packet transmission control. The available bit rate and the spectral efficiency are determined by the bandwidth and the modulation level, respectively. Accordingly, the number of packets available for transmission is affected by the modulation level of the packets. However, if we consider modulation levels with higher spectral efficiency in order to increase the number of packets and reduce the expected video distortion, the packet error rate of the transmitted packets can also be increased because the spectrally efficient modulation levels are sensitive to channel noise. This is another reason for the increment in expected video distortion, because the erroneous received packets cannot be used for video reconstruction. Therefore, this article considers the minimization of expected video distortion by the optimization of two factors—packet extraction for transmission and modulation level allocation for the extracted packets. Packet extraction is optimized for the path between the source and destination nodes, whereas the modulation level for each extracted packet is optimized for each link along the transmission path.

Keywords: cross-layer optimization, medium grain scalability, adaptive modulation.

1. Introduction

Scalable video coding (SVC), as standardized by the joint video team of the international telecommunication union—telecommunication standardization sector (ITU-T) and the international organization for standardization/international electro-technical commission (ISO/IEC) [1], is a video compression method that can bandwidth-efficiently support multiple spatial-temporal resolutions for a single video. It also supports a multi-bit rate feature that can be adapted to network or channel variations. These standard SVC properties can be utilized in diverse applications, such as multi-user video streaming services, distributed video streaming multihop networks, or scalable on-demand services, as discussed in [2]. This article considers medium grain scalability (MGS), which is one of the standard bit rate scalable coding methods [3]. MGS provides network abstraction layer (NAL) packets (MGS packets in

this article) that can be dropped without causing a decoding violation. To efficiently utilize the multi-bit rate feature, unequal protection (UEP) strategies can be considered for MGS packets, as each packet has a different priority in terms of its rate distortion (RD) attribute. For example, a priority index was developed in [3,4] to indicate the priority that can be used for UEP. UEP has been considered in video transmission systems, as in [5-15]. In [5-7], an application layer resource type, such as parity data, was considered. In contrast, the studies [8-15] considered physical layer (PHY) optimization for SVC or MGS video streams, as in the proposed method. In [8], multiple code division multiple access (CDMA) channels were proposed, with a different processing gain for the SVC quality layers. The suggested optimization problem can be simplified by separately transmitting each SVC layer to each CDMA channel, so that as many CDMA channels as SVC layers are required to fully utilize the method. In [9,10], frequency diversity was utilized by orthogonal frequency division multiple access systems. Modulation and channel coding was designed to guarantee

* Correspondence: daeyeon@awcc.uec.ac.jp

¹Advanced Wireless Communication Research Center, University of Electro-Communications, Tokyo, Japan

Full list of author information is available at the end of the article

the same target bit error rate (BER) in [9,10], whereas [11,12] considered a flexible packet error rate (PER) according to the RD attribute of the video packets. The algorithms were designed to find the transmission modes of multi-rate transmitters, which minimize the expected video distortion. In other words, studies [11,12] jointly addressed UEP by only transmitting video packets with a higher priority, and by allocating more transmission time to higher priority packets. This article considers the same approach for a wireless transmission path consisting of multiple links. For multihop wireless channels, the cross-layer optimization (CLO) designs of [13-15] are introduced for a video streaming service. These designs, including optimal path selection, assume a sufficient number of intermediate nodes, so that if the quality of a link in one path degrades, an alternative path can be substituted. A predefined transmission time is reserved for each node, and the remaining time for each node is an important factor for these CLO designs. However, in the case where the number of intermediate nodes is too small, and only one feasible transmission path is available, this path selection diversity cannot be achieved. Therefore, the flexible allocation of transmission time for links on the selected path must be considered. In this article, such time allocation is achieved by allocating the adaptive modulation levels for the links. This article also assumes that the available transmission power for each link is limited, in order to prevent interference to surrounding communication systems. Therefore, we focus on the modulation levels of the links according to their channel state information (CSI) in terms of the received signal-to-noise ratio (SNR).

This rest of the article is organized as follows. Section 2 introduces the proposed system, outlines the problem statements, and formulates an optimization problem. Section 3 provides three levels of algorithm for solving this problem. The performance of the proposed method is demonstrated in Section 4, and we present our conclusions in Section 5.

2. Introduction to the proposed system

2.1. Configuration of the proposed system

The proposed method efficiently transmits MGS packets over a wireless transmission path connected by multiple links. It is assumed that the density of the nodes is sufficiently low that there exists only one feasible path between the source and destination nodes. A predefined transmission time for the path is allocated prior to the proposed optimization, and the total time for the path can flexibly be distributed between the links on the path. Therefore,

$$\sum_{i=0}^{\mathcal{P}-1} \sum_{h=0}^{\mathcal{H}-1} \tau_{i,h} \leq T,$$

where $\tau_{i,h}$ is the time required to transmit the i th packet over the h th link, T is the predefined time that is determined according to the required frame rate of the video for real time streaming, and \mathcal{P} and \mathcal{H} are the number of video packets and links, respectively. We assume that the transmission power is fixed over the nodes in a path to prevent interference to surrounding communication systems. The proposed method is designed to optimize packet transmission and modulation level allocation according to the quality of each link. We assume that CSI concerning each link is fed back to the source nodes via a backward control channel. The source node extracts those packets available for transmission, and finds the modulation level of every link for each packet scheduled to be transmitted. These modulation levels are signaled to the corresponding intermediate nodes before the packet is transmitted. Each intermediate node demodulates the received packet, remodulates it according to the modulation level information, and forwards it to the next node.

2.2. Problem statements

2.2.1. Expected distortion analysis

Video distortion of decoded frames in a group of pictures (GOP) [1] is affected by the combination of packets available for the video reconstruction. Therefore, it is necessary to predict the combination at the destination node in order to control and reduce the distortion. The combination at the destination node for each packet in a GOP results from two factors—the packet drop rate (PDR) (decided by the transmitter) and the PER (influenced by channel noise). We define the packet loss rate (PLR), which is the probability that the packet is not available at the destination node, as $\phi = 1 - (1 - X)(1 - Y)$ where X and Y are the PER and PDR, respectively. For \mathcal{P} packets in a GOP, the number of combinations is $2^{\mathcal{P}}$, as two cases (that of being used and unused for decoding) can be considered for each packet. Therefore, the expected distortion of the k th frame is

$$\bar{d}_k = \sum_{c=0}^{2^{\mathcal{P}}-1} \Phi_c d_{k,c}, \quad (1)$$

where Φ_c and $d_{k,c}$ are the probability that the c th combination occurs at the destination node and the distortion of the k th frame in the c th combination, respectively. Equation (1) implies that $2^{\mathcal{P}}$ decoding simulations are required to calculate \bar{d}_k , because $d_{k,c}$ must be measured for $0 \leq c < 2^{\mathcal{P}}$ by the decoding simulations. Φ_c can be expressed in terms of the PLR of \mathcal{P} packets (ϕ_i for $0 \leq i < \mathcal{P}$) as

$$\Phi_c = \prod_{i=0}^{\mathcal{P}-1} (\alpha_{i,c} (1 - \phi_i) + (1 - \alpha_{i,c}) \phi_i)$$

where $\alpha_{i,c}$ denotes whether P_i is to be used ($\alpha_{i,c} = 1$) or not ($\alpha_{i,c} = 0$). Video distortion is largely affected by the reference structure, which can be established in various ways. A multiple reference structure [1] is considered to improve coding efficiency and error resilience. However, according to [16], such improvements are largely dependent on the temporal characteristics of the videos, and, indeed, the improvements are not especially large, in most cases, compared to the complexity of this type of structure. In addition, both the video encoder and decoder require a large amount of memory to store the multiple frames to reference. Therefore, this article considers hierarchical B, with its single-reference coding structure. In this case, the number of decoding simulations can be reduced from 2^P , as discussed at the end of Section 2.2.1. Furthermore, by considering only combinations with a high probability of occurrence, the number of simulations can be reduced to a practical level.

A. Simple examples of expected distortion Let us assume that there are three frames, each of which is coded to one packet, as depicted in Figure 1. In this figure, the packet reference [1] is expressed by the arrows. F_k is the k th frame in temporal order, and P_k is its coded packet. In this article, we define the number of effective packets (NEP) for each frame. The resulting distortions of the three frames are decided by the combination of NEP (CNEP) of the frames. For this coding structure, four CNEPs can be considered, as shown in Table 1. If a referenced packet is unavailable, any packet referencing it is also unavailable. For example, if P_0 is unavailable, P_1 becomes unavailable, so that P_2 also becomes unavailable. Therefore, if NEP for F_0 is 0, all of the packets of F_1 and F_2 become unavailable for decoding. Hence, NEPs for F_1 and F_2 also become 0, as shown by the CNEP of 0. CNEP = 1 denotes the case where P_0 is available and P_1 is unavailable. Therefore, the NEP for F_2 is also 0, because P_2 becomes ineffective for distortion. The probability of occurrence of each CNEP is shown in the table in terms of the PLR of the packets. If P_0 is unavailable, P_1 and P_2 are also unavailable, so that CNEP = 0. In other words, the probability of CNEP = 0 is the same as the PLR of P_0 , ϕ_0 . As CNEP = 1 in the case where P_0 is available and P_1 is unavailable, the probability of CNEP = 1 is $(1-\phi_0)\phi_1$. In this way, the

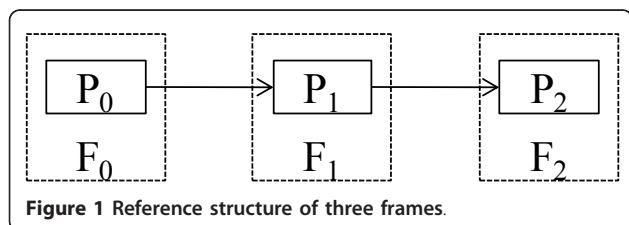


Figure 1 Reference structure of three frames.

Table 1 Possible CNEPs of the reference structure in Figure 1

CNEP	NEP			Probability
	F_0	F_1	F_2	
0	0	0	0	ϕ_0
1	1	0	0	$(1-\phi_0)\phi_1$
2	1	1	0	$(1-\phi_0)(1-\phi_1)\phi_2$
3	1	1	1	$(1-\phi_0)(1-\phi_1)(1-\phi_2)$

probability of the four CNEPs can be calculated, so that the expected distortion of the three frames can be obtained from (1). Therefore, the expected distortion of F_0 is $\bar{d}_0 = \phi_0 d_{0,0} + (1-\phi_0)\phi_1 d_{0,1} + (1-\phi_0)(1-\phi_1)\phi_2 d_{0,2} + (1-\phi_0)(1-\phi_1)(1-\phi_2)d_{0,3}$. As the distortion of F_0 is not affected by the quality of F_1 or F_2 , $d_{0,1} = d_{0,2} = d_{0,3}$. Therefore, $\bar{d}_0 = \phi_0 d_{0,0} + (1-\phi_0)d_{0,1}$. In this way, the expected distortion of F_1 can be written as $\bar{d}_1 = \phi_0 d_{1,0} + (1-\phi_0)\phi_1 d_{1,1} + (1-\phi_0)(1-\phi_1)d_{1,2}$, where $d_{1,2} = d_{1,3}$. If we assume SVC, we can consider multiple layers [1]. Therefore, as shown in Figure 2, each frame can be coded to multiple packets, where each packet of a frame represents a spatial or quality layer. In this article, we focus on quality scalability and consider MGS coding. Therefore, in the rest of this article, the term “packet” means base quality layer packet or MGS packet. In Figure 2, $P_{k,l}$ is the l th quality layer packet of F_k , where the 0th quality layer means the base layer. The associated CNEPs are given in Table 2. In the same way as the previous coding structure, the expected distortion of F_0 can be obtained as

$$\bar{d}_0 = \phi_{0,0} d_{0,0} + (1-\phi_{0,0})\phi_{0,1} d_{0,1} + (1-\phi_{0,0})(1-\phi_{0,1})d_{0,4}, \quad (2)$$

where $d_{0,1} = d_{0,2} = d_{0,3}$ and $d_{0,4} = d_{0,5} = d_{0,6}$. In this way, the expected distortion of each frame can be expressed in terms of PLRs. The CNEPs in Table 2 can be categorized into three groups according to the number of frames requiring error concealment, where a number of error concealment techniques [17] can be considered for any frame that NEP = 0. For example, CNEP = 0 requires concealment for both of the frames, whereas CNEP = 1 or 4 requires concealment for F_1 . Neither frame requires any error concealment for the remaining four CNEPs. Therefore, \mathcal{C} (the number of CNEPs) for Figure 2 is $2^0 + 2^1 + 2^2 = 7$. \mathcal{C} can be generalized as

$$\mathcal{C} = \sum_{t=0}^{\mathcal{T}} \mathcal{Q}^t, \quad (3)$$

where \mathcal{T} and \mathcal{Q} are the number of frames and quality layers, respectively. This analysis can be extended to more complicated coding structures, such as the hierarchical B structure [1].

B. Expected distortion of hierarchical B structure For further coding efficiency and temporal scalability, SVC is designed to provide the hierarchical B structure shown in Figure 3, where four temporal layers are presented. To simplify the distortion analysis, this article defines reference groups (RGs), so that the hierarchical B structure in Figure 3 can be analyzed as depicted in Figure 4. In the figure, frames in an RG are independent of each other. Therefore, one RG can be considered as one frame in order to simplify the analysis of the reference structure and calculate the video distortion (detailed discussion is given with Table 3). Although some of the references in Figure 3 are omitted in Figure 4, this representation can sufficiently describe the reference relations. For example, we can see from Figure 4 that F_4 cannot be decoded if F_0 is not decoded, although the reference arrow from F_0 to F_4 is omitted in the figure. The CNEP of a frame unit was introduced previously. In this section, the CNEP of an RG unit is introduced in order to analyze the expected distortion of the hierarchical B coding structure. If the number of quality layers (including the base layer) is 3, the number of CNEPs is 364 according to (3), as \mathcal{T} (the number of RGs in this case) is 5. Therefore, 364 decoding simulations, as listed in Table 3, must be accomplished to obtain \bar{d}_k for $0 \leq k \leq 8$, where the NEPs of frames in an RG are the same as the NEP of the RG. For example, CNEP = 362 means that 2 packets are available for every frame in RG_4 (F_1, F_3, F_5 , and F_7) and 3 packets are available for the remaining frames. As each of the frames in an RG is independent of other frames in the same RG, these frames can independently be analyzed. For example, in order to calculate \bar{d}_1 (the expected distortion of F_1), the probability of each CNEP must be calculated. Therefore, the NEPs of RG_0, RG_1, RG_2, RG_3 , and RG_4 can be considered as the NEPs of F_0, F_8, F_4, F_2 ,

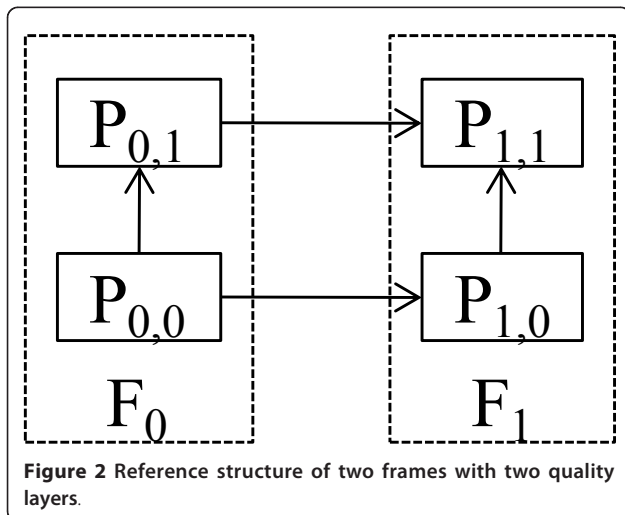


Figure 2 Reference structure of two frames with two quality layers.

Table 2 Possible CNEPs of the reference structure in Figure 2

CNEP	NEP		Probability
	F_0	F_1	
0	0	0	$\varphi_{0,0}$
1	1	0	$(1-\varphi_{0,0})\varphi_{0,1}\varphi_{1,0}$
2	1	1	$(1-\varphi_{0,0})\varphi_{0,1}(1-\varphi_{1,0})\varphi_{1,1}$
3	1	2	$(1-\varphi_{0,0})\varphi_{0,1}(1-\varphi_{1,0})(1-\varphi_{1,1})$
4	2	0	$(1-\varphi_{0,0})(1-\varphi_{0,1})\varphi_{1,0}$
5	2	1	$(1-\varphi_{0,0})(1-\varphi_{0,1})(1-\varphi_{1,0})\varphi_{1,1}$
6	2	2	$(1-\varphi_{0,0})(1-\varphi_{0,1})(1-\varphi_{1,0})(1-\varphi_{1,1})$

and F_1 , respectively. The probability of each CNEP is then calculated based only on the PLRs of packets in F_0, F_8, F_4, F_2 , and F_1 , because \bar{d}_1 is not affected by the PLRs of packets in the other frames. The distortion of the k th frame of the c th CNEP, $d_{k,c}$, can be calculated from the simulation of the c th CNEP. For example, to calculate $d_{1,362}$, one packet (the highest quality layer packet) for each frame in RG_4 is eliminated, and all of the remaining packets are decoded. The resulting distortion of F_1 can be obtained for $d_{1,362}$. Note that the NEPs of frames F_3, F_5 , and F_7 in the same RG do not affect $d_{1,362}$. However, performing 364 decoding simulations to optimize 9 frames is impractical. Therefore, a more efficient version of this expected distortion analysis is required, as discussed in Section 3.5.

2.2.2. Increment of expected distortion

The increment in the expected distortion due to every packet loss must be obtained in order to minimize the expected distortion, as discussed in Section 3. As we have mentioned, the expected distortion can be expressed in terms of the PLR. Consequently, the increment can also be expressed in terms of the PLR. In Figure 2, for example, if the PLR of $P_{0,0}$ increases, the expected distortion of F_0 (\bar{d}_0 in (2)) increases according to

$$\frac{\partial \bar{d}_0}{\partial \phi_{0,0}} = d_{0,0} - \phi_{0,1}d_{0,1} - (1 - \phi_{0,1})d_{0,4}.$$

On the other hand, the PLR of $P_{1,0}$, $\phi_{1,0}$, is irrelevant to the expected distortion \bar{d}_0 given by (2), as no packet in F_0 references $P_{1,0}$. Therefore,

$$\frac{\partial \bar{d}_0}{\partial \phi_{1,0}} = 0,$$

as (2) does not contain $\phi_{1,0}$. In this way, the increment of the expected distortion can also be calculated for the hierarchical B structure, where the structure can be

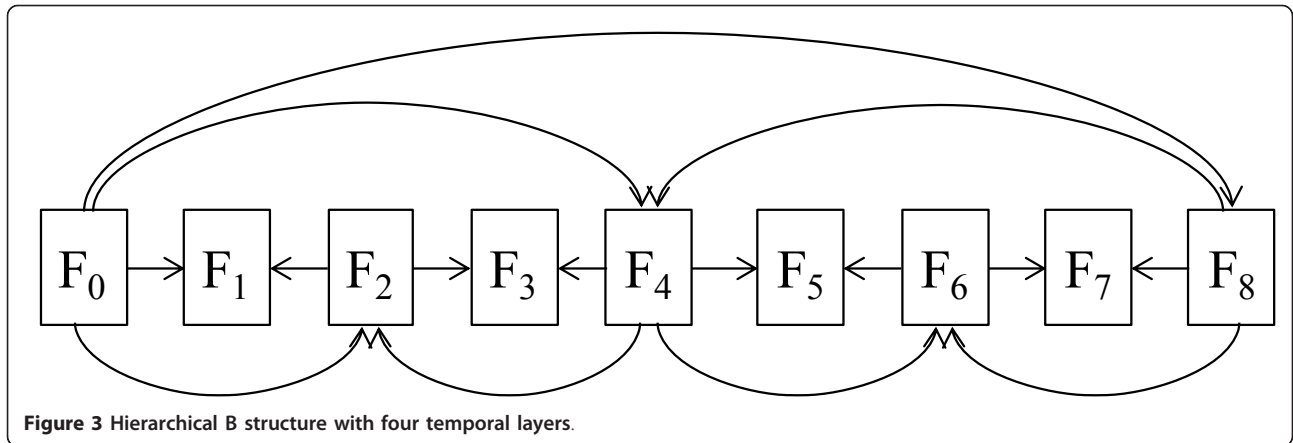


Figure 3 Hierarchical B structure with four temporal layers.

simplified by using the reference group concept discussed in Section 2.2.1.

2.2.3. Expected delay

For P_i , the number of bits that can be transmitted per second over the h th link is $r \log_2(\mu_{i,h})$, where r and $\mu_{i,h}$ are the bandwidth and modulation level allocated to P_i over the h th link, respectively. Therefore, the expected time required to transmit P_i over the h th link is

$$\bar{\tau}_{i,h} = \frac{(1 - Y_i) L_i}{r \log_2(\mu_{i,h})}, \quad (4)$$

where L_i and Y_i are the number of bits and the PDR of P_i , respectively. Note that $\bar{\tau}_{i,h}$ is the expected value, as Y_i is a probability. Therefore, the delay constraint is

$$\bar{\tau} = \sum_{P_i \in \mathbf{P}} \bar{\tau}_{i,h} \leq T, \quad (5)$$

where $\mathbf{P} = \{P_i | i = 0, 1, \dots, \mathcal{N} - 1\}$ is the set of packets in a GOP ($\mathcal{N} = \mathcal{F} \times \mathcal{Q}$ is the number of packets in a GOP, where \mathcal{Q} is the number of quality layers employed).

2.2.4. Optimization

The purpose of the proposed method is to minimize the average of the expected distortion of each frame in a GOP. Therefore,

$$\bar{d} = \frac{1}{\mathcal{F}} \sum_{k=0}^{\mathcal{F}-1} \bar{d}_k,$$

where \mathcal{F} is the number of frames in a GOP. The PLR of packet P_i in \mathbf{P} is

$$\phi_i = 1 - (1 - X_i)(1 - Y_i), \quad (6)$$

where X_i is the PER of P_i . Here, the Lagrange optimization formula can be written as

$$(\mu^*, \mathbf{Y}^*, \lambda^*) = \arg \min (\bar{d} + \lambda (\bar{\tau} - T)), \quad (7)$$

where $\bar{\tau}$ and T are the required and allowed delay in transmitting \mathbf{P} , respectively. λ is the Lagrange multiplier. As shown in (7), the optimized modulation level set μ^* , PDR set \mathbf{Y}^* , and Lagrange multiplier λ^* must be found in order to minimize $\bar{d} + \lambda (\bar{\tau} - T)$.

3. Implementation of the proposed system

3.1. Resource distortion attribution

As discussed in [3], each additional MGS coded video packet drop results in an increment in the received video distortion and a decrement in the required bit rate. For each MGS packet, the study [3] defines

Table 3 CNEPs of the RG structure in Figure 4

CNEP	NEP				
	RG ₀	RG ₁	RG ₂	RG ₃	RG ₄
0	0	0	0	0	0
1	1	0	0	0	0
...					
362	3	3	3	3	2
363	3	3	3	3	3

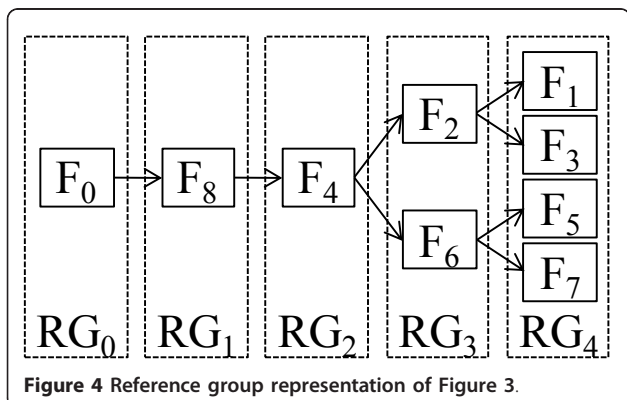


Figure 4 Reference group representation of Figure 3.

For bit rate control, the packets are prioritized according to the RD attribute. In this article, we modify this to consider the required time resource (delay). Any increment in the PDR or modulation level results in an increment in the expected distortion and a decrement in the required delay. Therefore, this article defines the resource-distortion (RsD) attribute, which is the distortion increment/delay decrement, as

$$\begin{aligned} \lambda_{i,h}^{\text{PHY}} &= \frac{\partial \bar{d}}{\partial \mu_{i,h}} / \left(-\frac{\partial \bar{\tau}}{\partial \mu_{i,h}} \right) \\ \lambda_i^{\text{MAC}} &= \frac{\partial \bar{d}}{\partial Y_i} / \left(-\frac{\partial \bar{\tau}}{\partial Y_i} \right). \end{aligned} \quad (8)$$

For $\lambda_{i,h}^{\text{PHY}}$, the dividend is

$$\frac{\partial \bar{d}}{\partial \mu_{i,h}} = \frac{\partial x_{i,h}}{\partial \mu_{i,h}} \frac{\partial X_{i,h}}{\partial x_{i,h}} \frac{\partial X_i}{\partial X_{i,h}} \frac{\partial \phi_i}{\partial X_i} \frac{\partial \bar{d}}{\partial \phi_i}. \quad (9)$$

$X_{i,h}$ is the PER of the hop, which is $1 - (1 - x_{i,h})^{L_i}$, where $x_{i,h}$ is the BER of the h th hop. According to [18], $x_{i,h}$ can be approximated as

$$x_{i,h} \approx 0.2 \exp \left[\frac{1.6\sigma_{i,h}}{1 - \mu_{i,h}} \right] \quad (10)$$

for multi-level quadrature amplitude modulation (MQAM), where $\sigma_{i,h}$ is the SNR of the hop. In (9), X_i is the PER at the destination node, which is

$$X_i = 1 - \prod_{h=0}^{\mathcal{H}-1} (1 - X_{i,h}),$$

where \mathcal{H} is the number of hops in the path. Therefore, Equation (9) is

$$\frac{\partial \bar{d}}{\partial \mu_{i,h}} \approx \frac{\theta_{i,h} \bar{\delta}_i x_{i,h} L_i (1 - x_{i,h})^{L_i - 1} \ln^2 [5x_{i,h}] (1 - Y_i)}{1.6\sigma_{i,h}},$$

where

$$\theta_{i,h} = \frac{\partial X_i}{\partial X_{i,h}} = \prod_{\substack{g=0 \\ g \neq h}}^{\mathcal{H}} (1 - X_{i,g}) \quad (11)$$

and

$$\bar{\delta}_i = \frac{\partial \bar{d}}{\partial \phi_i} = \frac{1}{\mathcal{F}} \sum_{k=0}^{\mathcal{F}-1} \frac{\partial \bar{d}_k}{\partial \phi_i}. \quad (12)$$

From (4), the divisor in (8) is

$$-\frac{\partial \bar{\tau}}{\partial \mu_{i,h}} = \frac{(1 - Y_i) L_i \ln [2]}{r \mu_{i,h} \ln^2 [\mu_{i,h}]}.$$

Therefore,

$$\lambda_{i,h}^{\text{PHY}} = \begin{cases} \frac{r \theta_{i,h} \bar{\delta}_i x_{i,h} L_i (1 - x_{i,h})^{L_i - 1} \ln^2 [5x_{i,h}] \ln^2 [\mu_{i,h}]}{1.6\sigma_{i,h} \ln [2]}, & Y_i < 1 \\ \text{any value,} & Y_i = 1 \end{cases} \quad (13)$$

Note that if Y_i is 1, both the dividend and divisor of $\lambda_{i,h}^{\text{PHY}}$ are 0, so that $\lambda_{i,h}^{\text{PHY}}$ cannot be specified. In other words, $\lambda_{i,h}^{\text{PHY}}$ can take any value. On the other hand, the dividend of λ_i^{MAC} in (8) can be written, using (6) and (12), as

$$\frac{\partial \bar{d}}{\partial Y_i} = (1 - X_i) \bar{\delta}_i.$$

From (4), the divisor is

$$-\frac{\partial \bar{\tau}}{\partial Y_i} = \tau_i$$

Therefore,

$$\lambda_i^{\text{MAC}} = \begin{cases} \frac{(1 - X_i) \bar{\delta}_i}{\tau_i}, & 0 < Y_i < 1 \\ \text{any value,} & Y_i \in \{0, 1\} \end{cases} \quad (14)$$

Note that Y_i is a probability, and can be controlled to take an edge value of 0 or 1 by the transmitter. If it is one of these two edge values, neither the dividend nor the divisor of λ_i^{MAC} can be specified. Consequently, λ_i^{MAC} cannot be specified, which means it can take any value.

3.2. Algorithm I-searching continuous modulation level and PDR

If μ^* and Y^* minimize $\bar{d} + \lambda (\bar{\tau} - T)$ in the optimization formula (7) differentiated with respect to any element $\mu_{i,h}$ in μ^* and Y_i in Y^* becomes 0, so that

$$\frac{\partial \bar{d}}{\partial \mu_{i,h}} + \lambda \frac{\partial \bar{\tau}}{\partial \mu_{i,h}} = 0 \text{ and } \frac{\partial \bar{d}}{\partial Y_i} + \lambda \frac{\partial \bar{\tau}}{\partial Y_i} = 0.$$

The equations above can be reformed and substituted with RsD attributes as

$$\lambda = \lambda_{i,h}^{\text{PHY}} = \lambda_i^{\text{MAC}}. \quad (15)$$

As Y_i affects $\lambda_{i,h}^{\text{PHY}}$ and λ_i^{MAC} , as shown in (13) and (14), the following three settings can be considered in order to satisfy (15).

- *Setting 1:* Set $\mu_{i,h}$ to satisfy (15) and Y_i to $0 < Y_i < 1$.
- *Setting 2:* Set $\mu_{i,h}$ to satisfy $\lambda = \lambda_{i,h}^{\text{PHY}}$ and Y_i to 0.
- *Setting 3:* Set Y_i to 1.

The target service of the proposed method is real-time streaming that requires a more stringent delay constraint than that on the expected delay given in (5). However, Y_i is not 0 or 1 for Setting 1, which means that P_i may or may not be sent. Therefore, by excluding Setting 1 from our consideration, Equation (5) can be modified to

$$\tau = \sum_{P_i \in \mathcal{P}} (1 - Y_i) \tau_{i,h} \leq T, \text{ where } Y_i \in \{0, 1\}. \quad (16)$$

Therefore, Algorithm I considers only Settings 2 and 3. Algorithm I is developed to search $\mu_{i,h}$ so as to make $\lambda_{i,h}^{\text{PHY}}$ approaches λ . From (10), $\mu_{i,h}$ can be evaluated in terms of $x_{i,h}$ as $1 - 1.6\sigma_{i,h}/\ln[x_{i,h}]$. Therefore, $\mu_{i,h}$ satisfying (15) can be determined by obtaining $x_{i,h}$ satisfying (15). $x_{i,h}$ is the BER value, which is usually close to 0. Thus, it is convenient to consider the logarithmic values $\beta_{i,h} = \ln(x_{i,h})$ and $\Lambda_{i,h} = \ln(\lambda_{i,h}^{\text{PHY}})$ as

$$\Lambda_{i,h} = \beta_{i,h} + \ln[\mu_{i,h}(1 - x_{i,h})^{L_i-1} \ln^2(5x_{i,h}) \ln^2(\mu_{i,h})] + \Psi_{i,h}, \quad (17)$$

where

$$\psi_{i,h} \triangleq \frac{r\theta_{i,h}\bar{\delta}_i}{1.6\sigma_{i,h} \ln[2]} \text{ and } \Psi_{i,h} \triangleq \ln(\psi_{i,h}) \quad (18)$$

is the term independent of $\beta_{i,h}$ and its logarithm, respectively. Figure 5 provides an example of $\Lambda_{i,h}$, where L_i and $\Psi_{i,h}$ are set to 5000 and 6, respectively. To calculate $x_{i,h}$ according to (10), $\sigma_{i,h}$ is set to 15 dB. If input Λ (the value that $\Lambda_{i,h}$ should approach) is G_1 , the solution of $\beta_{i,h}$ is g_1 . However, if input Λ is G_2 , there are two candidate solutions for $\beta_{i,h}$, which are $g_{2,1}$ and $g_{2,2}$. In this case, the lower value $g_{2,1}$ is selected, as the purpose of the algorithm is to minimize $\bar{\delta}$. To find the intersection, Algorithm I is developed as shown in Figure 6. The second term in (17) is $\ln(\mu_{i,h}) + (L_i-1) \ln(1-x_{i,h}) + 2 \ln(\ln(5x_{i,h})) + 2 \ln(\ln(\mu_{i,h}))$. Therefore, $\Delta_{i,h}$ (the slope of $\Lambda_{i,h}$) in Algorithm I can be obtained as

$$\Delta_{i,h} = 1 - \frac{x_{i,h}(L_i-1)}{1-x_{i,h}} + \frac{1}{\ln(5x_{i,h})} \left(\frac{1-\mu_{i,h}}{\mu_{i,h}} \left(1 + \frac{2}{\ln \mu_{i,h}} \right) + 2 \right) \quad (19)$$

by differentiating each term in (17) with respect to $\beta_{i,h}$. Note that $\Psi_{i,h}$ has vanished, as it is independent of $\beta_{i,h}$. First, Algorithm I checks whether or not $\psi_{i,h}$ is 0, whereas Algorithm II calculates $\theta_{i,h}$ and $\bar{\delta}_i$ according to (11) and as discussed in Section 2.2.2, respectively, then inputs those values to Algorithm I. If at least one of the packets referenced by P_i is dropped, $\bar{\delta}_i$ becomes 0 because a change in φ_i cannot contribute to \bar{d} . Consequently, $\psi_{i,h}$ becomes 0 and $\Lambda_{i,h}$ becomes $-\infty$, regardless of $\beta_{i,h}$, according to (18). This means that a value

of $\beta_{i,h}$ satisfying (15) does not exist. Therefore, Algorithm I sets Y_i to 1, i.e., Setting 3, in order to drop the packet and satisfy (15) regardless of $\beta_{i,h}$. Otherwise, if \bar{d} is not 0, $\Psi_{i,h}$ is calculated from $\psi_{i,h}$ according to (18) in order to obtain the intersection shown in Figure 5. It can be seen from Figure 5 that the intersection of the curve $\Lambda_{i,h}$ and the solution, e.g., g_1 or $g_{2,1}$ (rather than $g_{2,2}$), must be somewhere at which $\Delta_{i,h} \geq 0$ (note that the slope at the intersection of $\Delta_{i,h}$ and $g_{2,2}$ is negative, and that the solution $g_{2,2}$ maximizes the video distortion, which is not desired). Therefore, $\beta_{i,h}$ should initially be set to a sufficiently small value. We found that $\beta_{i,h} = -10$ ($x_{i,h} = 10^{-10}$) was sufficiently small to make the slope positive for various settings of L_i , $\Psi_{i,h}$, and $\sigma_{i,h}$. To reduce the computational power, the algorithm utilizes $\Delta_{i,h}$. If $\Delta_{i,h}$ is negative, Y_i is set to 1 and the process is terminated. Otherwise, $\beta_{i,h}$ is increased by $(\Lambda - \Lambda_{i,h})/\Delta_{i,h}$ to allow $\Lambda_{i,h}$ to approach Λ . If $\Lambda_{i,h}$ is within a predefined tolerance of Λ , we consider $\beta_{i,h}$ satisfying (15) to have been found. Otherwise, we repeat the iterations of calculating $\Delta_{i,h}$ and $\beta_{i,h}$ until $\Lambda_{i,h}$ becomes close to Λ . A small number of such iterations are sufficient to approach the solution, because $\Delta_{i,h}$ has an almost constant value of 1 for low $\beta_{i,h}$, e.g., lower than approximately -4.5 in Figure 5. This is due to the second and third terms in (19) becoming very small compared to 1 as $x_{i,h}$ approaches 0. However, if the input Λ is higher than the possible maximum of $\Lambda_{i,h}$, e.g., $\Lambda > G_1$ in Figure 5, the solution does not exist. In this case, $\Delta_{i,h}$ becomes negative while the $\Delta_{i,h}$ and $\beta_{i,h}$ calculations are repeated. Table 4 shows an example where the input Λ is set to 6 for Figure 5.

As shown in Table 4, $\beta_{i,h}$ was initially set to -10, as depicted in Algorithm I in Figure 6. In the first iteration of Algorithm I, $\Delta_{i,h}$ and $\Lambda_{i,h}$ are calculated to be 0.9935 and -0.6443 according to (17) and (19), respectively. $\Lambda_{i,h} = -0.6443$ is not close to $\Lambda = 6$, so Algorithm I determines $\Delta_{i,h}$, and finds that it is negative.

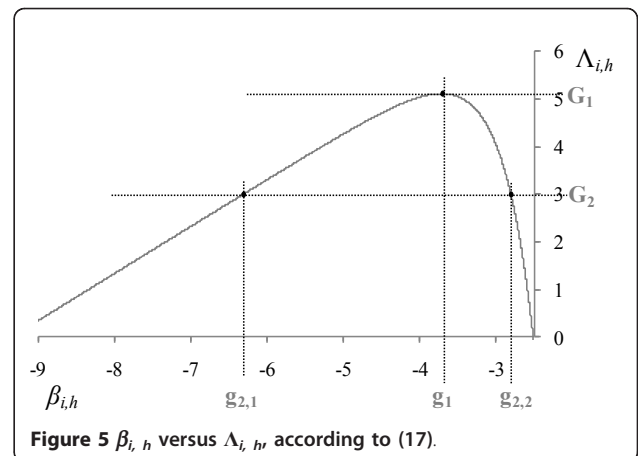
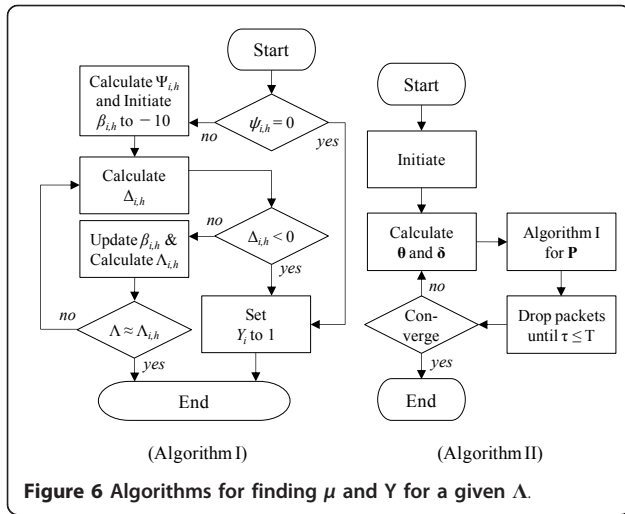


Figure 5 $\beta_{i,h}$ versus $\Lambda_{i,h}$ according to (17).



Therefore, Algorithm I is terminated by setting the PDR Y_i to 1, which means that P_i will be dropped.

3.3. Algorithm II-refining parameters

In Algorithm I, $\bar{\delta}_i$, and $\theta_{i,h}$ are used to calculate $\psi_{i,h}$ according to (18). $\bar{\delta}_i$ is affected by the PLR of packets referenced by P_i and the PLRs of packets referencing P_i . In addition, $\theta_{i,h}$ is affected by the PERs of P_i of other links, according to (11). However, δ and θ (which are the set of $\bar{\delta}_i$ and $\theta_{i,h}$ for \mathbf{P} , respectively) are not available unless Algorithm I has been accomplished for the related packets and links so far. Therefore, this article proposes Algorithm II, which initializes the PDR and PER of every packet over every link to 0 in order to initiate δ and θ . Therefore, Algorithm I can calculate μ and Y over the packets. However, if μ and Y do not

satisfy (16), a real-time streaming service cannot be guaranteed. Therefore, Algorithm II drops packets with Setting 2 until (16) is satisfied, as shown in Figure 6. Equation (14) can be used as the criterion for deciding which packets will be dropped, as it quantifies the attribute of transmitting P_i . Therefore, it drops packets with a lower λ_i^{MAC} . In this way, the set Y obtained by Algorithm I over \mathbf{P} is modified according to the packet dropping procedure. Using the obtained PDR set Y and the PERs that can be calculated by μ , the algorithm can renew δ and θ . μ and Y are then calculated again via the same procedure. If every element pair in the previously calculated μ and Y is close to corresponding with those in the new μ and Y , we consider (15) to be satisfied. Therefore, Algorithm II is terminated. Otherwise, the algorithm repeats this procedure until μ and Y become stable.

3.4. Algorithm III-satisfying the resource constraint

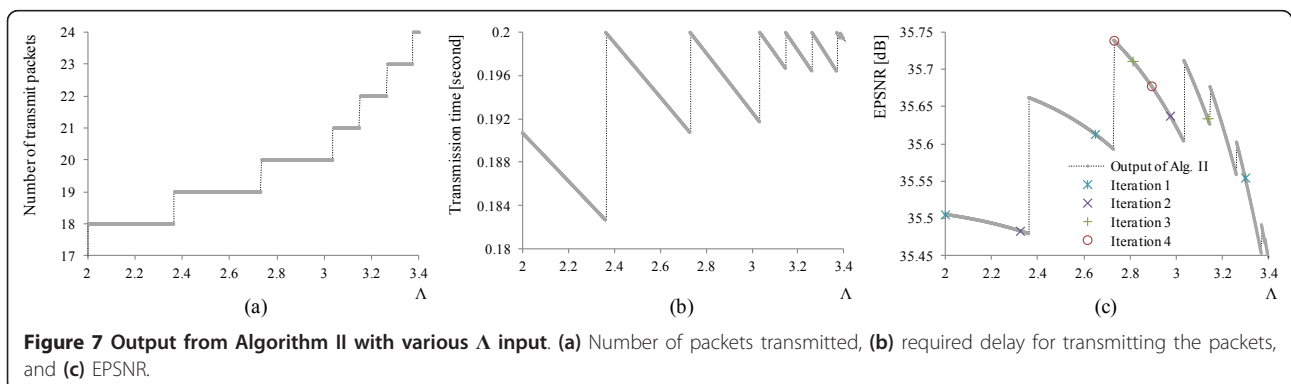
Figure 7 shows some examples of output from Algorithm II in terms of (a) the number of packets transmitted, (b) the required delay for transmitting the packets, and (c) the expected peak SNR (EPSNR), where

$$\text{EPSNR} = 10 \log_{10} \left(\frac{255^2}{\bar{d}} \right).$$

The expected distortion \bar{d} is calculated in terms of mean squared error (MSE). The allowed delay T in (16) is set to 0.2 s. Using the example of Figure 7, this section discusses Algorithm III, which is designed to find the value of Λ that maximizes EPSNR, where the EPSNR performance of Algorithm II for various input Λ is shown in Figure 7c. The reason for the local maxima of EPSNR in Figure 7c is as follows. If $\Lambda = 2.6$, Algorithm II allocates 19 packets to transmit and a 0.194-s transmission time (by modulation setting for the 19 packets), as shown in Figure 7a, b. As the allowed delay is set to 0.2 s, more transmission time can be allocated by adjusting Λ . By reducing Λ from 2.6, the BER for

Table 4 Calculations for $\Lambda_{i,h}$ to approach $\Lambda = 6$ in Figure 5

Iteration	$\beta_{i,h}$	$\Lambda_{i,h}$	$\Delta_{i,h}$
1	-10	-0.6443	0.9935
2			-1.4876



each transmitted packet is lowered (as shown in Figure 5), which means that a lower modulation level is allocated. Therefore, the transmission time becomes larger by reducing Λ from 2.6, as shown in Figure 7b. As Λ approaches 2.4, the transmission time approaches its maximum of 0.2 s. However, if Λ is less than 2.4, the number of transmitted packets is reduced to 18, as shown in Figure 7a, to keep the transmission time below 0.2 s. Thus, the transmission time is reduced discontinuously, as shown in Figure 7b. These discontinuities in radio resource (transmission time) result in the discontinuities and local EPSNR maxima observed in Figure 7c. Algorithm III is designed to find the Λ value that maximizes EPSNR. In the rest of this section, Algorithm III is explained by reference to the EPSNR performance of Figure 7c. In order to avoid the local maxima in the figure, Algorithm III iteratively reduces the search range. In the first iteration (Iteration 1 in Figure 7c), three equally spaced initial Λ values are selected. In the figure, these three values are 2, 2.65, and 3.3, where the values are chosen for the purpose of visual convenience. (If the EPSNR maximum does not exist between 2 and 3.3, the maximum cannot be found with these initial values. Therefore, a sufficiently wide range for the initial values of -10, 0, and 10 is considered in the simulations of Section 4.) Of the three EPSNRs with these initial values, that with $\Lambda = 2.65$ is the greatest. Therefore, in the next iteration (Iteration 2), two more Λ values (2.325 and 2.975, which make the interval $0.65/2 = 0.325$ for the five Λ values of Iterations 1 and 2) are chosen around 2.65, and EPSNRs for the two new Λ values are determined by Algorithm II. In Iteration 3, two more Λ values (2.8125 and 3.1375) are chosen around $\Lambda = 2.975$, which gave the greatest EPSNR among the five Λ values of Iterations 1 and 2, and EPSNRs for these two values are determined. In this way, two more Λ values are tested in the next iteration, and the same procedure is repeated until the predefined number of iterations is accomplished. The Λ value with the greatest EPSNR among all tested values is then chosen for the packet transmission.

3.5. Practical expected distortion and increment

The expected distortion and its increment are discussed in Section 2. The expected distortion increment $\bar{\delta}_i$ is utilized to find the optimal packet extraction and modulation level allocation, and is updated while Algorithm II is performed. To obtain the exact amount of the expected distortion, the number of error patterns given by (3) must be simulated, which is impractical as \mathcal{T} and \mathcal{Q} grow. Therefore, the number of simulations must be reduced for practical implementation. For example, we can simulate only the $\mathcal{T}\mathcal{Q} + 1$ CNEP that are most

likely to occur at the destination node. Table 5 shows an example of the CNEPs where the number of temporal layers and quality layers are 2 and 2, respectively. If the NEPs in the temporal level immediately before the current level is the same as or 1 greater than that of the current level, the CNEP is chosen in order to calculate δ (for Algorithm II) and \bar{d} (for Algorithm III). In this case, $\mathcal{T} = 3$, so that the number of required CNEPs is $7 = \mathcal{T}\mathcal{Q} + 1$, as in Table 5. In the case of $\mathcal{T} = 5$ and $\mathcal{Q} = 3$, it is 16, which is much more practical than the 364 simulations discussed in Section 2.2.1.

3.6. Complexity of the proposed method

The proposed method consists of three algorithm levels. As shown in Figure 6, Algorithm II runs Algorithm I as many as \mathcal{HP} times until the convergence criterion for Algorithm II is achieved. Therefore, the complexity of Algorithm II is $\kappa_2 = \mathcal{HP}(\kappa_1 + \kappa_{\text{MAC}})\eta_2$, where κ_1 and κ_{MAC} are the complexity of Algorithm I and the calculation of (14) for the packet dropping module in Algorithm II, respectively. η_2 is the number of iterations required for convergence. The complexity of Algorithm I is $\kappa_1 = \kappa_0\eta_1$, where κ_0 is the complexity of calculating (17) and (19) and η_1 is the number of calculations. As Ψ_i is fixed for each iteration of Algorithm I, the calculation of the second term in (17) is the main cause of the complexity. The computational power required to calculate this term is mainly dependent on calculating $10^{\beta_{i,h}}$ to obtain $x_{i,h}$, $\ln(5x_{i,h})$, $\ln(\mu_{i,h})$, and $(x_{i,h}^*)^{L_i-1}$. Once its components have been calculated, Equation (19) can easily be obtained. As κ_{MAC} is trivial compared to κ_1 , κ_2 can be considered as $\mathcal{HP}\kappa_1\eta_2$. Consequently, the total complexity for the proposed method for optimizing \mathbf{P} is $\kappa = \mathcal{HP}\kappa_0\eta_1\eta_2\eta_3$, where η_3 is the predefined number of samples taken to find the optimal Λ , as discussed in Section 3.4. Therefore, κ can be considered as $\mathcal{O}(\mathcal{HP})$.

3.7. Algorithm I–discrete-searching discrete modulation level and PDR

Section 3.2 introduced Algorithm I for finding a continuous modulation level and PDR for a packet. Algorithm I finds a continuous modulation level satisfying $\lambda = \lambda_{i,h}^{\text{PHY}}$. However, it is not practical to realize this continuous modulation value. Therefore, this section considers discrete modulation levels affordable by MQAM. For instance, five modulations, such as 4-, 16-, 64-, 256-, and 1024-QAM, can be considered. By reforming (10), $\beta_{i,h}$ can be written as

$$\beta_{i,h} = \mathcal{C}_1 + \frac{\mathcal{C}_2\sigma_i}{1 - \mu_{i,h}}, \quad \text{where } \mu_{i,h} \in \mathbf{M}. \quad (20)$$

Table 5 Reduced number of CNEPs considered for practical implementation

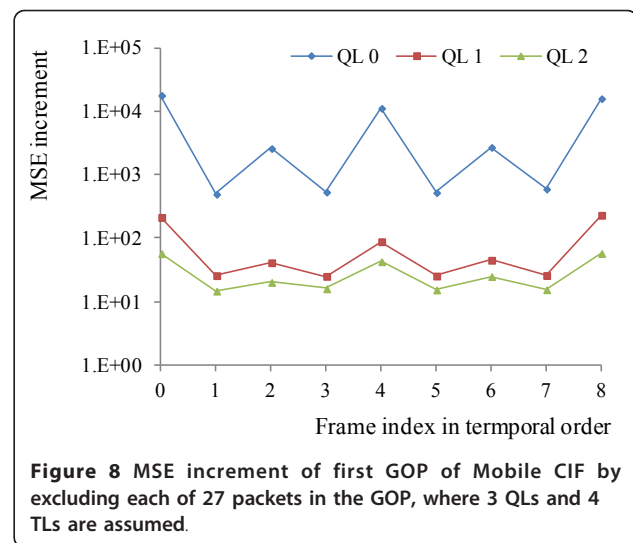
CNEP	NEP		
	F_0	F_1	TL1
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	2	1	1
5	2	2	1
6	2	2	2

Constants C_1 and C_2 are $\log_{10}[0.2]$ and $1.6/\ln[10]$, respectively. As $M = \{4,16,64,256,1024\}$, the number of possible $\beta_{i,h}$, and consequently the number of possible $\Lambda_{i,h}$, is also five. Hence, it is difficult to satisfy $\lambda = \lambda_{i,h}^{PHY}$. Therefore, an alternative method (Algorithm I–Discrete) is considered, which finds the $\Lambda_{i,h}$ closest to Λ from among the five possible candidates. Consequently, modulation levels with $\Delta_{i,h} \geq 0$ are selected from the five candidates, and the level whose $\Lambda_{i,h}$ value is closest to Λ is found. However, if each of the five candidates is less than 0, it can be considered that none of the modulation levels is adequate. Therefore, the algorithm sets the PDR to 0 and terminates. By substituting Algorithm I (which is called by Algorithm II, as in Figure 6) with Algorithm I–Discrete, an efficient packet extraction and discrete modulation level scheme can be obtained by Algorithm III.

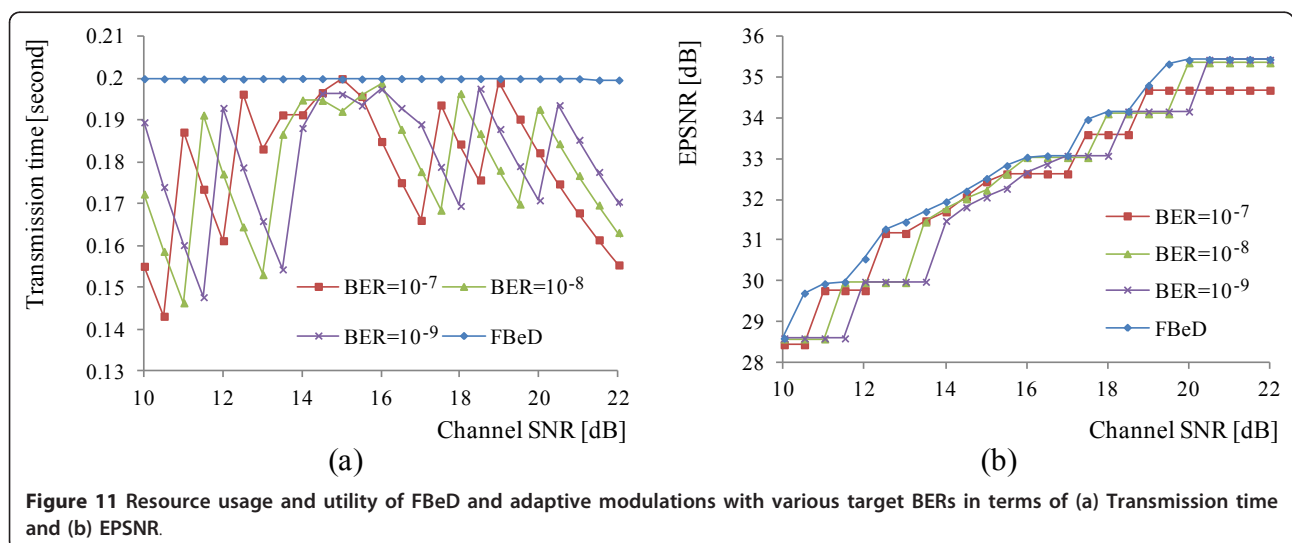
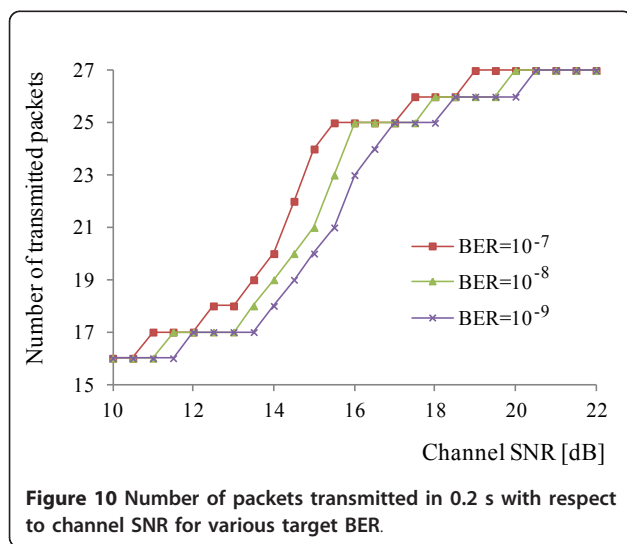
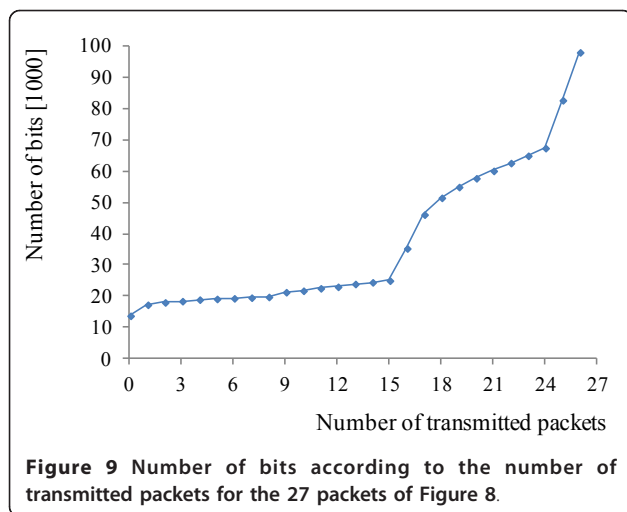
4. Simulation results

4.1. Transmission path configured with a single link

Prior to a discussion of multi-link cases, the performance of the proposed method for a single link is discussed. The proposed method is designed to find the optimal set of transmitted packets, and the modulation level of each transmitted packet, for a transmission path configured with multiple links. Therefore, if the proposed method is applied to a single link, it operates similarly to the method in [12], which is designed to determine the optimal set of transmitted packets and transmission modes. In the rest of this article, the two methods are termed Single Link Optimization (SLO) and Multi-Link Optimization (MLO). In this section, JSVM 9.19.7 [17] is considered for the simulations. For the simulations of Sections 4.1 and 4.2, Mobile common intermediate format (CIF) is tested at 30 frames per second (FPS) with a bandwidth of 150 kHz for each link, where the GOP size is 8 (which results in three temporal layers (TLs) in the hierarchical B structure, as in Figure 3) and the number of quality layers (QLs) is 3. Figure 8 shows the total MSE increment of the first



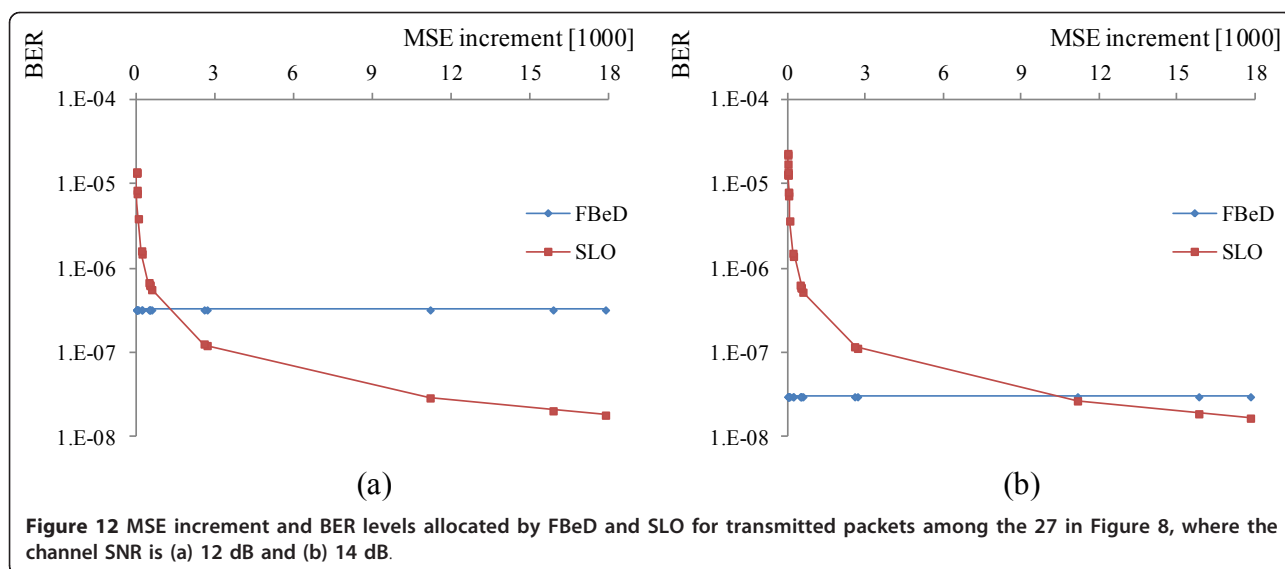
GOP by excluding each of the 27 packets configuring 9 frames and 3 QLs (QL 0, QL 1, and QL 2), where QL 0 is the base quality layer and the ninth frame is shared by the next GOP. If we consider an adaptive modulation strategy to guarantee a fixed BER level for every link, the number of transmitted packets for the limited transmission time (0.2 s in this section) is determined by the BER level. If the BER level is lowered to reduce channel error, the bit rate is reduced, so that the number of packets transmitted is reduced. Figure 9 shows the number of bits according to the number of transmitted packets, where the 27 packets of Figure 9 are prioritized by RD attribution, as in [3]. Figure 10 shows the number of the transmitted packets with respect to the channel SNR, where the target BER is set to either 10^{-7} , 10^{-8} , or 10^{-9} . As shown in Figure 10, more packets can be transmitted by considering a higher BER level. The transmission time for the three BER levels of the adaptive modulation strategies are shown in Figure 11a. If a BER level is set for the packets, the transmission time is determined solely by the number of packets. Therefore, the allowed time of 0.2 s is not fully utilized in most cases, as shown in Figure 11a. Consequently, it is unfair to compare the three BER levels of the adaptive modulations with the proposed method, as the proposed method is designed to fully utilize the given transmission time. Therefore, this paper considers a Flexible BER Decision FBeD method, which finds the BER level that minimizes the expected video distortion when applied to all transmitted packets. FBeD is considered to provide a performance upper bound to the adaptive modulation method designed to guarantee a fixed BER level, and FBeD shows better performance than the other three methods in terms of EPSNR, as shown in Figure 11b. Furthermore, as FBeD can fully utilize the



given transmission time, as shown in Figure 11a, it is suitable for comparison with the proposed method. Meanwhile, the proposed method individually determines the BER level for each packet according to its distortion increment (MSE increment) and bit length. Figure 12 shows the allocated BER of each packet and its MSE increment for FBeD and SLO with channel SNRs of 12 and 14 dB. The cumulative transmission time for each transmitted packet using the two methods is shown in Figure 13, where the cumulative time is considered to show the total transmission time reaching 0.2 s. In Figure 13, the transmitted packets are sorted in decreasing order of MSE increment, so that it can be conveniently observed alongside Figure 12. For a 12-dB channel SNR, both FBeD and SLO transmit 17 packets. In the 14 dB case, FBeD and SLO transmit 19 and 22 packets, respectively. As shown in Figure 12a, SLO provides higher protection for 4 of the 17 packets, and more time is allocated for the packets with a high MSE increment, as shown in Figure 13a, where the number of transmitted packets is the same. As shown in Figure 12b, SLO allocates a higher BER for 19 of the 22 packets, so that it allocates less time for packets with a low MSE increment, as shown in Figure 13b. Hence, SLO can transmit three packets more than FBeD while providing a lower BER level for three packets with a high MSE increment. For both values of the channel SNR, SLO shows better performance. This is shown in Figure 14, which shows the number of transmitted packets and EPSNR for various channel SNR cases.

4.2. Transmission path configured with two links

Whereas SLO finds the optimal transmission packet set and BER (modulation level) for a single link, MLO finds these two parameters for multi-links. In the following



simulations, a transmission path configured with multiple links is considered, where the channel SNR of one link on the path changes. In Section 4.2, a transmission path configured with two links (Links 0 and 1) is considered in which the channel SNR of Link 1 is 25 dB and that of Link 0 varies. Figure 15 shows the transmission time used for each link to transmit all of the packets. In Figure 15, it can be seen that less time is allocated to a link with a relatively high SNR, so that the other link can use more time to protect the packets from channel error. For SLO, however, an equal time of 0.1 s is assumed to be allocated to each link, as it is designed for a single link. For multi-links, FBeD is adjusted to allocate the same BER level to all links, so that the transmission time can be flexibly allocated to each link, as in MLO. Figure 16

shows the number of transmitted packets and EPSNR with respect to a varying channel SNR of Link 0. As the same amount of time is allocated to each link when the SNRs of the links are the same, as shown in Figure 15, MLO shows a similar performance to SLO, as shown in Figure 16b. As shown in Figure 16, if the SNR difference between the two links is relatively small (i.e., SNR of Link 0 is 23-33 dB), the EPSNR performance of SLO and MLO is almost identical. In this case, the EPSNRs are higher than that of FBeD. However, as a link with a lower SNR cannot occupy more time in SLO, the EPSNR cannot be improved, even for Link 0 with a channel SNR above 33 dB. When Link 0 has a SNR below 23 dB, Link 0 cannot occupy more time in SLO, so that its EPSNR degrades to below that of FBeD.

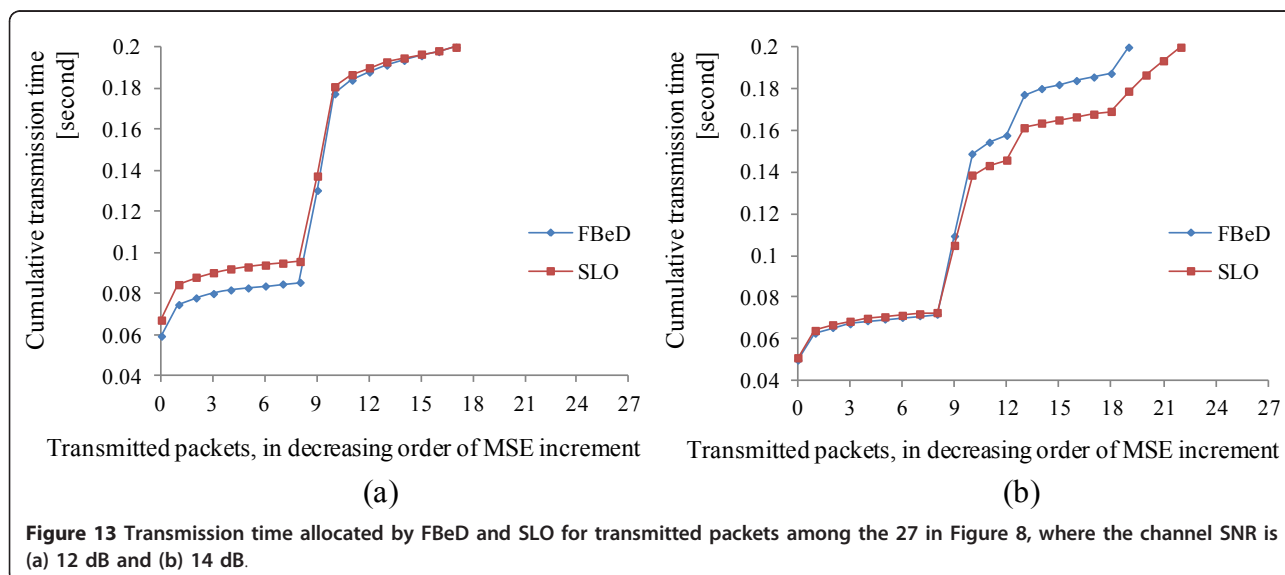
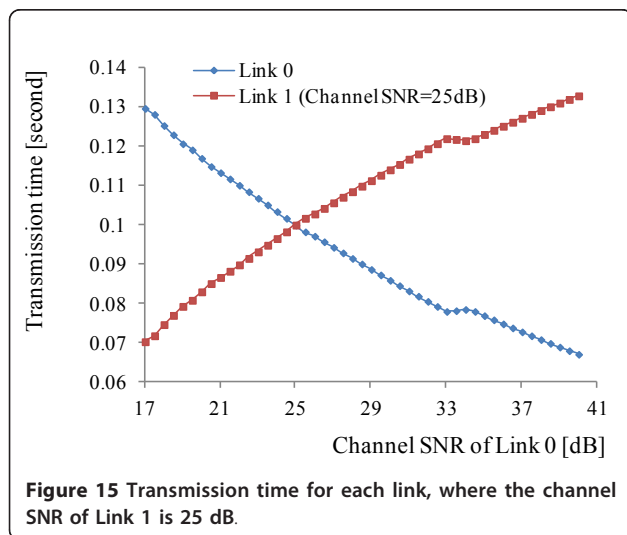
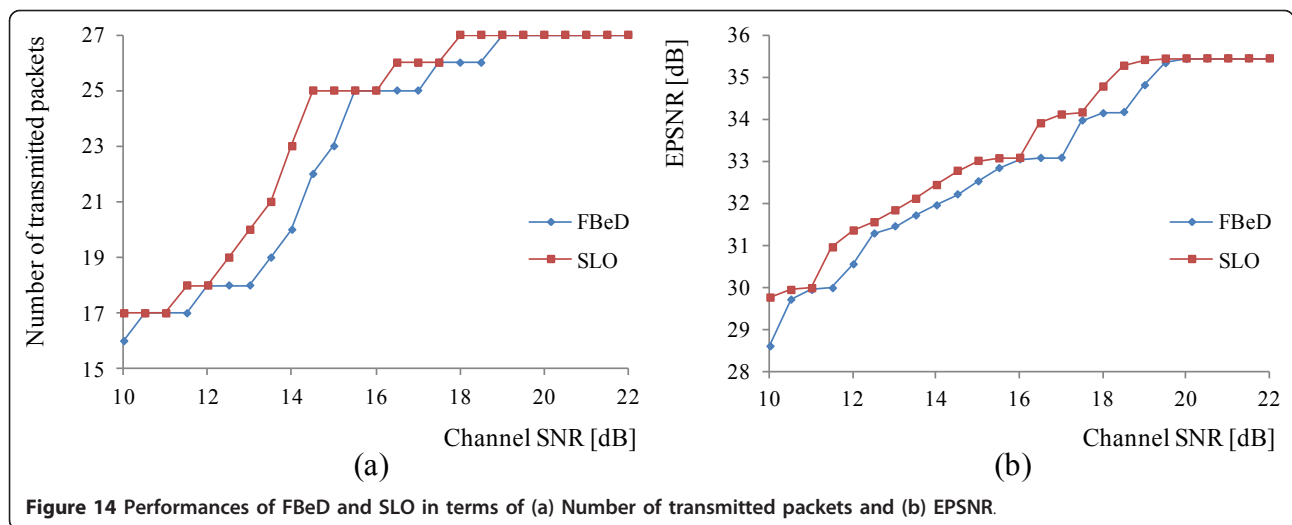
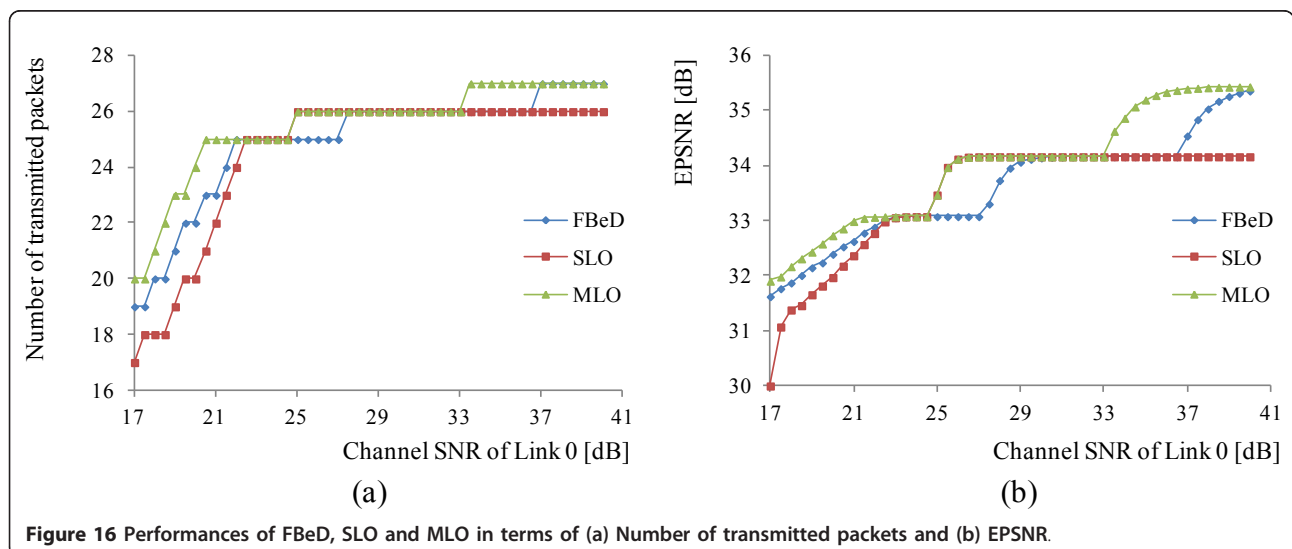


Figure 13 Transmission time allocated by FBeD and SLO for transmitted packets among the 27 in Figure 8, where the channel SNR is (a) 12 dB and (b) 14 dB.



4.3. Transmission path configured with three links

In Figure 17, a path with three links (Links 0, 1, and 2) is considered, where the channel SNRs of Links 1 and 2 are 25 and 30 dB, respectively. Because the EPSNR performance of SLO is sensitive to an imbalance in the channel SNRs of links, SLO becomes more inadequate as the number of links grows. Therefore, the performance of SLO has been omitted from Figure 17. This figure shows the test results from three quarter CIF (QCIF)-15 FPS and three CIF-30 FPS videos, where the bandwidth for each link was set to 50 kbps for QCIF videos and 200 kbps for CIF videos. The GOP size for QCIF videos was set to 4 and that for CIF videos was set to 8. In addition to the two methods with continuous modulation levels (FBeD (CNT), MLO (CNT)), the performance of the methods with discrete modulation levels (as discussed in Section 3.7) is also shown in



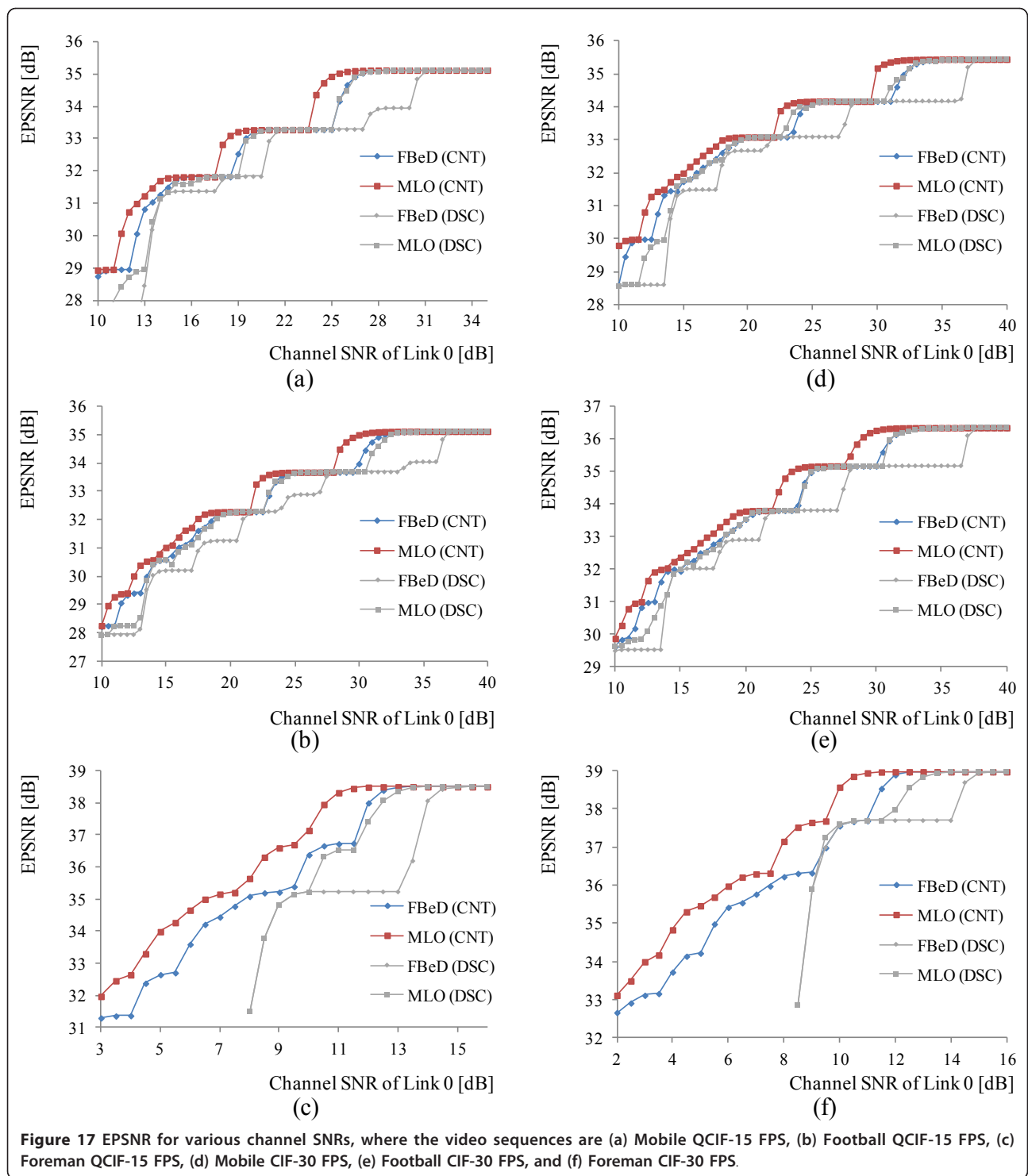


Figure 17 EPSNR for various channel SNRs, where the video sequences are (a) Mobile QCIF-15 FPS, (b) Football QCIF-15 FPS, (c) Foreman QCIF-15 FPS, (d) Mobile CIF-30 FPS, (e) Football CIF-30 FPS, and (f) Foreman CIF-30 FPS.

Figure 17. For the six graphs in Figure 17, MLO (CNT) shows better performance than FBeD (CNT). In both the QCIF and CIF cases, Foreman shows acceptable video quality (EPSNR higher than 30 dB) for relatively low channel SNR compared to Mobile and Football for FBeD (CNT) and MLO (CNT). This is because the

visual and motion characteristics of Foreman are relatively simple compared to Mobile and Football, so it can be compressed more. Therefore, a lower modulation level can be allocated for Foreman while transmitting as many packets as required for acceptable video quality. However, as shown in Figure 17c, f, continuous

modulation levels below that supported by discrete MQAM ($\mu_{i,h}$ less than 4 in (20)) are evaluated by FBeD and MLO for low channel SNR, so that FBeD (DSC) and MLO (DSC) degrade severely as the channel SNR degrades. Nevertheless, it is observed that MLO (DSC) shows improvement over MLO (DSC) for the six video sequences.

5. Conclusions

This article proposed a method to jointly exploit the bit rate and channel adaptation provided by MGS and adaptive modulation over a transmission path consisting of multiple wireless links. The proposed algorithms found the optimal packet transmission scheme by extracting packets according to an RsD attribution quantified as the distortion increment over the delay decrement. In order for the extracted packets to be transmitted, the proposed algorithms also found the optimal modulation allocation. The two factors of packet extraction and modulation allocation were optimized simultaneously by solving an optimization problem to minimize the total distortion.

Abbreviations

BER: bit error rate; CDMA: code division multiple access; CIF: common intermediate format; CLO: cross-layer optimization; CNEP: combinations of number of effective packets; CSI: channel state information; EPSNR: expected peak signal-to-noise ratio; FPS: frames per second; GOP: group of pictures; MGS: medium grain scalability; MLO: multi-link optimization; MQAM: multi-level quadrature amplitude modulation; MSE: mean squared error; NAL: network abstraction layer; NEP: number of effective packets; PDR: packet drop rate; PER: packet error rate; PHY: physical layer; PLR: packet loss rate; RD: rate distortion; RG: reference group; RsD: resource distortion; SLO: single link optimization; SNR: signal-to-noise ratio; SVC: scalable video coding; UEP: unequal protection.

Author details

¹Advanced Wireless Communication Research Center, University of Electro-Communications, Tokyo, Japan ²Department of Radio-Communications, Kyunghee University, Seoul, Korea

Competing interests

The authors declare that they have no competing interests.

Received: 15 July 2011 Accepted: 14 March 2012

Published: 14 March 2012

References

1. H Schwarz, D Marpe, T Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard. *IEEE Trans Circ Syst Video Technol.* **17**(9), 1103–1120 (2007)
2. T Schierl, T Stockhammer, T Wiegand, Mobile video transmission using scalable video coding. *IEEE Trans Circ Syst Video Technol.* **17**(9), 1204–1217 (2007)
3. I Amonou, N Cammas, S Kervadec, S Pateux, Optimized rate-distortion extraction with quality layers in the scalable extension of H.264/AVC. *IEEE Trans Circ Syst Video Technol.* **17**(9), 1186–1193 (2007)
4. CH Foh, Y Zhang, Z Ni, J Cai, KN Ngan, Optimized cross-layer design for scalable video transmission over the IEEE 802.11e networks. *IEEE Trans Circ Syst Video Technol.* **17**(12), 1665–1678 (2007)
5. G Cheung, A Zakhor, Bit allocation for joint source/channel coding of scalable video. *IEEE Trans Image Process.* **9**(3), 340–356 (2000). doi:10.1109/83.826773

6. E Maani, AK Katsaggelos, Unequal error protection for robust streaming of scalable video over packet lossy networks. *IEEE Trans Circ Syst Video Technol.* **20**(3), 407–416 (2010)
7. N Nejati, H Yousefzadeh, H Jafarkhani, Distortion optimal transmission of multi-layered FGS video over wireless channels. *IEEE Trans Sel Areas Commun.* **28**(3), 510–519 (2010)
8. LP Kondi, D Srinivasan, DA Pados, SN Batalama, Layered video transmission over wireless multirate DS-CDMA links. *IEEE Trans Circ Syst Video Technol.* **15**(12), 1629–1637 (2005)
9. G Su, Z Han, M Wu, A scalable multiuser framework for video over OFDM networks: fairness and efficiency. *IEEE Trans Circ Syst Video Technol.* **16**(10), 1217–1231 (2006)
10. H Ha, C Yim, Y Kim, Cross-layer multiuser resource allocation for video communication over OFDM networks. *EURASIP J Comput Commun.* **31**(15), 3553–3563 (2008). doi:10.1016/j.comcom.2008.05.010
11. YP Fallah, H Mansour, S Khan, P Nasiopoulos, HM Alnuweiri, A link adaptation scheme for efficient transmission of H.264 scalable video over multirate WLANs. *IEEE Trans Circ Syst Video Technol.* **18**(7), 875–887 (2008)
12. H Mansour, YP Fallah, P Nasiopoulos, V Krishnamurthy, Dynamic resource allocation for MGS H.264/AVC video transmission over link-adaptive networks. *IEEE Trans Multimedia.* **11**(8), 1478–1491 (2009)
13. N Mastrorade, DS Turaga, M Schaar, Collaborative resource exchanges for peer-to-peer video streaming over wireless mesh networks. *IEEE Trans Sel Areas Commun.* **25**(1), 108–118 (2007)
14. T Xiaolin, Y Andreopoulos, M Schaar, Distortion-driven video streaming over multihop wireless networks with path diversity. *IEEE Trans Mobile Comput.* **6**(12), 1343–1356 (2007)
15. Y Zhang, S Qin, Z He, Fine-granularity transmission distortion modeling for video packet scheduling over mesh networks. *IEEE Trans Multimedia.* **12**(1), 1–12 (2010)
16. Z Zhang, Z Hu, A Hai, D Lu, Performance study of error resilience with multiple reference frames in H.264/AVC in conditions of low bit-rate, in *ICEMI'09*, pp. 3-298–3-301 (2009)
17. JSVM Software Manual, Version. JSVM 9.19.7 (January 2010)
18. ST Chung, AJ Goldsmith, Degrees of freedom in adaptive modulation: a unified view. *IEEE Trans Commun.* **49**(9), 1561–1571 (2002)

doi:10.1186/1687-1499-2012-105

Cite this article as: Kim et al.: Modulation level allocation for MGS streaming over a multihop wireless channel. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:105.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com