A novel fountain code-based mobile IPTV multicast system architecture over WiMAX network

Hyunchul Joo [a], Changwoo Yoon [b], Tai-Won Um [b], Hwangjun Song [a,*]

[a] Computer Science and Engineering, POSTECH (Pohang University of Science and Technology), Pohang 790-784, Republic of Korea
[b] Broadcasting & Telecommunications Convergence Research Laboratory, ETRI, Daejeon 305-700, Republic of Korea

ARTICLE INFO

ABSTRACT

In this paper, we present a novel fountain code-based mobile IPTV multicast system architecture over WiMAX network. In the proposed system, the transmission algorithm at a base station determines the control parameters of a fountain-encoded IPTV multicast stream adaptively to the wireless link states of subscribers in order to provide a stable IPTV service with minimum resource usage on WiMAX network, and the channel grouping algorithm at a server makes near-optimal channel grouping based on channel selection preferences to pursue an effective tradeoff between the channel zapping time and the processing complexity of a subscriber. Finally, experimental results are provided to show the performance of the proposed system.

1. Introduction

In the recent, IPTV (Internet Protocol TeleVision) service is considered one of the key applications in the telecommunication market [1,2]. This service model enables service providers to leverage their existing infrastructure and expand their market opportunities by drawing in new subscribers and increasing average revenue per subscriber [3]. It is expected that IPTV service will be extended to wireless networks in the near future, in development called mobile IPTV service [4]. It is becoming more feasible thanks to the growth in mobile devices, advanced broadband networking and communication technologies, and sophisticated video compression technology. In particular, WiMAX (Worldwide interoperability for Microwave Access) network [5,6] is considered one of the strongest candidates to support mobile IPTV service. It is an emerging wireless access network standard that provides high data rates and differentiated services according to individual QoS (Quality of Service) requirements in a wide coverage area. Although WiMAX network provides high data rates in the wireless environment, it is still poor compared to wired networks including Ethernet, DOCSIS (Data Over Cable Service Interface Specifications), and fiber optic networks. Furthermore, it is expected that much higher bandwidth will be required at a BS (Base Station) sooner or later

to support various multimedia services over wireless networks. Thus, mobile IPTV systems must be designed to effectively manage resource usage of IPTV channel streams on WiMAX network.

Generally speaking, there are three distribution schemes for mobile IPTV service over the wireless environment: unicast, multicast, and broadcast transmission. In the case of unicast transmission, mobile IPTV service will use a significant amount of transmission bandwidth because each subscriber requires a dedicated bandwidth as the number of subscribers increase. On the other hand, multicast and broadcast transmission use fixed transmission bandwidth regardless of the number of subscribers, and thus can improve network utilization. Recently, various wireless network standards, e.g. 3GPP/MBMS (Multimedia Broadcast Multicast Service) [7], 3GPP2/BCMCS (BroadCast and MultiCast Service) [8], WiMAX/MBS (Multicast and Broadcast Service) [9,10], DVB/DVB-H (Digital Video Broadcasting – Handheld) [11], etc., already support multicast and broadcast transmission. However, there are still several problems to be solved to successfully deploy multicast or broadcast service over wireless networks. For example, the transmitter should take into account the dynamically changing link states of all subscribers to achieve reliable multicast or broadcast service. Sometimes, it is inevitable that some subscribers experience packet losses caused by the time-varying wireless link characteristics and physical layer impairments, which may seriously degrade video quality. In the case of mobile IPTV service, it is one of the most important performance measures how to

* Corresponding author. Fax: +82 54 279 2299.
  E-mail address: hwangjun@postech.ac.kr (H. Song).

provide stable IPTV service to as many subscribers as possible. Thus, ARQ (Automatic Repeat reQuest) and FEC (Forward Error Correction) schemes [12] are widely employed at the application layer to effectively handle these packet losses in multicast and broadcast transmission. In the ARQ scheme, the subscriber may experience an increased delay due to the elapsed time from when the subscriber sends the retransmission request to when the requested packet arrives. And, as the number of subscribers becomes larger, the ARQ scheme increases uplink/downlink traffic and server load, and thus the resulting video quality may seriously deteriorate by the longer passing time. Furthermore, the ARQ scheme is not appropriate for mobile IPTV service since it has difficulty in handling retransmission requests from subscribers efficiently. On the other hand, the FEC scheme exploits redundant information to compensate for packet losses over wireless networks. By carefully selecting the amount of redundant information [13,14], the FEC scheme can mitigate packet losses without any feedback channel. Particularly, fountain code [15] is standardized by 3GPP and DVB-H as the AL-FEC (Application Layer Forward Error Correction) scheme for mobile IPTV service. Although the FEC scheme is more suitable than the ARQ scheme for mobile IPTV service, the FEC scheme requires some processing for reconstructing original packets at a subscriber, which results in extra processing time. Actually, these processing complexity and processing time at mobile IPTV devices are greatly associated with the QoE (Quality of Experience) of subscribers.

In this paper, we present a novel fountain code-based mobile IPTV multicast system architecture over WiMAX network. In the proposed system, the adaptive transmission algorithm at a BS and the channel grouping algorithm at a server are proposed to efficiently provide IPTV service to subscribers. The transmission algorithm adjusts to the control parameters of a fountain-encoded IPTV multicast stream adaptively to the wireless link states of subscribers in order to minimize resource usage on WiMAX network while satisfying the decoding failure rate constraint of fountain code. The channel grouping algorithm constructs channel groups based on channel selection preferences in order to reduce the average processing complexity while guaranteeing seamless channel changes at a subscriber. The rest of this paper is organized as follows. Related works are introduced in Section 2, the details of the proposed mobile IPTV multicast system architecture are specified in Section 3, experimental results are provided in Section 4, and concluding remarks are given in Section 5.

2. Related works

We study some related works regarding to IPTV multicast/broadcast system in Section 2.1 and review fountain code briefly in Section 2.2.

2.1. IPTV multicast/broadcast system

Adaptive transmission schemes have attracted a large amount of interest to effectively provide multicast/broadcast services over wireless networks. SARM (SNR-based Auto Rate for Multicast) [16] adapts the transmission rate at an access point to the subscriber with the worst channel condition for reliable multicast transmission. In [17], the transmission rate of each multicast stream is determined dynamically based on the proportional fair policy at a BS for achieving high throughput and good fairness. AL-FEC schemes are deployed to handle packet losses caused by unrecoverable transmission errors even though the sparsest modulation scheme and the lowest coding rate are selected for the robust transmission at the MAC (Media Access Control) layer and physical layer [15,18]. In recent years, joint optimization between the trans-

mission rate at the MAC layer and the FEC coding rate at the application layer has been studied actively to support reliable multicast transmission with high network efficiency [19–21].

So far, some schemes have been proposed to reduce the channel zapping time. Cho et al. [22] proposed sending the adjacent IPTV channel streams of the currently viewed IPTV channel stream in advance to the subscriber. When the subscriber's channel change requests are limited to adjacent channel streams, the subscriber can watch the selected channel stream without network delay. In [23], the video encoder periodically generates additional intra frames encoded by a lower bit rate and transmits them with normal video frames. Video decoding can be performed by using additional intra frames without waiting for the normal intra frame. In [24], the proposed scheme constructs multiple time-shifted multicast streams for an IPTV channel stream. The multicast stream with the shortest intra frame waiting time is selected for new channel change requests of the subscriber. In Microsoft Mediaroom [25], a server caches the latest GOPs (Group of Pictures) for each IPTV channel stream. For new channel change requests, the server transmits not only the multicast stream at a regular rate but also an additional unicast stream at a higher rate.

Actually, many research efforts have been devoted to efficiently providing mobile IPTV service over WiMAX network. In [4], She et al. proposed a two-level superposition coded multicasting scheme using SVC (Scalable Video Coding) [26]. The proposed scheme performs the superposition coding to two-layered IPTV channel streams with a static and different modulation scheme. The base layer stream provides IPTV service of minimum quality to as many subscribers as possible and the enhancement layer stream only supports subscribers with the good link conditions. Wang et al. [10] proposed a system to apply UEP (UnEqual Protection) according to the importance of layer-encoded IPTV channel streams. Compared to enhancement layer streams, more robust FEC coding is applied to the base layer stream against wireless channel errors. CMS (Cooperative Multicast Scheduling) [27] was presented to improve the network utilization for IPTV channel stream transmission. In the first phase, an IPTV channel stream is transmitted by using a dense modulation and high coding rate. In the second phase, subscribers with poor link quality are supported by other peer subscribers that have correctly received the IPTV channel stream during the preceding phase.

2.2. Fountain code

Fountain code such as LT [28] and Raptor [29] codes is an emerging block-based FEC scheme [30] and has some advantages for mobile IPTV service due to its high coding efficiency, low encoding/decoding processing time, and flexibility. Actually, there are active trials and commercial deployments for efficiently supporting IPTV service to subscribers using fountain code [18,31]. The detail encoding process of LT code is as follows. First of all, the data stream is divided into source blocks. A source block is composed of source symbols of a predefined size. As shown in Fig. 1(a), an encoding symbol is mapped to some source symbols. The number of mappings is defined as n_{degree} with a range from one to the number of source symbols in a source block (K). n_{degree} of each encoding symbol is determined based on the degree distribution probability ($\{p_1, p_2, \ldots, p_K\}$) that is allocated to $\{1, 2, \ldots, K\}$ values. And, n_{degree} source symbols are randomly selected, and then an encoding symbol is generated by performing bitwise XOR operations of the selected source symbols. This process is repeated until the last encoding symbol is generated. Since an encoding matrix realization can be characterized by the seed number of the pseudo-random number generator, a subscriber can easily reconstruct source symbols from encoding symbols if the server and the subscriber share the same seed number. The decoding process of LT
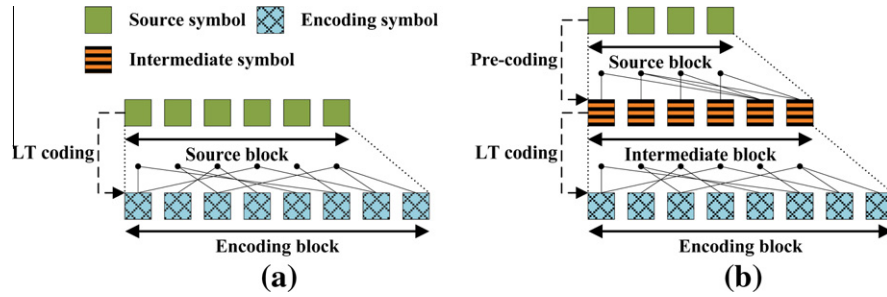
Fig. 1. Fountain encoding process: (a) LT encoding and (b) Raptor encoding.

code is equivalent to solving linear equations. In general, the simple decoding algorithm through message passing [30] is widely used due to its low complexity. At each step, the decoder finds an encoding symbol consisting of only a single source symbol. When this source symbol is recovered by the corresponding encoding symbol, it is added to all the encoding symbols that the source symbol is included. At a subscriber, all source symbols can be recovered if a sufficient number of encoding symbols are available even though some encoding symbols are lost. In general, the number of encoding symbols required for successful fountain decoding is calculated by

$$K' = (1 + \gamma) \cdot K,$$

where γ is the symbol overhead with a very small real number. The above equation means that the number of the received encoding symbols must be slightly larger than K to reconstruct source symbols successfully.

Raptor code is an extension of LT code that achieves linear encoding/decoding processing time. Raptor code is the concatenation of pre-code and LT code as shown in Fig. 1(b). At first, intermediate symbols are generated by pre-coding of source symbols to decrease n_{degree} of encoding symbols in LT code. And then, these intermediate symbols are encoded by LT code with reduced n_{degree} of encoding symbols.

3. Proposed fountain code-based mobile IPTV multicast system

The proposed fountain code-based mobile IPTV multicast system architecture is shown in Fig. 2. It is assumed that an IPTV server receives IPTV channel streams from terrestrial, cable, and satellite broadcasting systems. In the proposed system, fountain code is employed to improve the reliability of IPTV multicast

stream transmission over WiMAX network although it may incur extra processing time. Actually, reliability is the most important factor in the case of mobile IPTV service (it is recommended in [32,33] that the mean time interval between visible video distortions at a subscriber should be greater than one hour). As shown in Fig. 2, source RTP packets of the IPTV channel stream arriving within the protection time window create a source block of fountain codes. It is recommended that the number of source symbols in a source block should be larger than a thousand (i.e. source symbol constraint) [18,34] for fountain coding efficiency. Thus, the protection time window size and the symbol size ought to be set to satisfy the source symbol constraint. And, encoding symbols are generated by the linear combination of source symbols in the protection time window at the application layer, and transmitted to a WiMAX BS in IP multicasting manner. The BS transmits incoming IPTV multicast streams through MBS zones in downlink subframes. At a subscriber, fountain decoding is performed to extract source symbols of the requested IPTV channel stream, and then original RTP packets are obtained. Actually, fountain encoding and decoding processes are performed at the application layers of the server and the subscriber, respectively.

The goal of the proposed system is to support a stable IPTV service with minimum resource usage on WiMAX network and statistically guarantee seamless channel changes with low processing complexity at a subscriber. To achieve this goal, the proposed system includes the adaptive transmission algorithm at a BS and the channel grouping algorithm at the server. The proposed adaptive transmission algorithm broadcasts a fountain-encoded IPTV multicast stream considering the wireless link states of subscribers in order to minimize resource usage on WiMAX network and successfully decode source symbols with a high probability. The proposed channel grouping algorithm is presented to provide seamless channel changes at the cost of a little increased processing
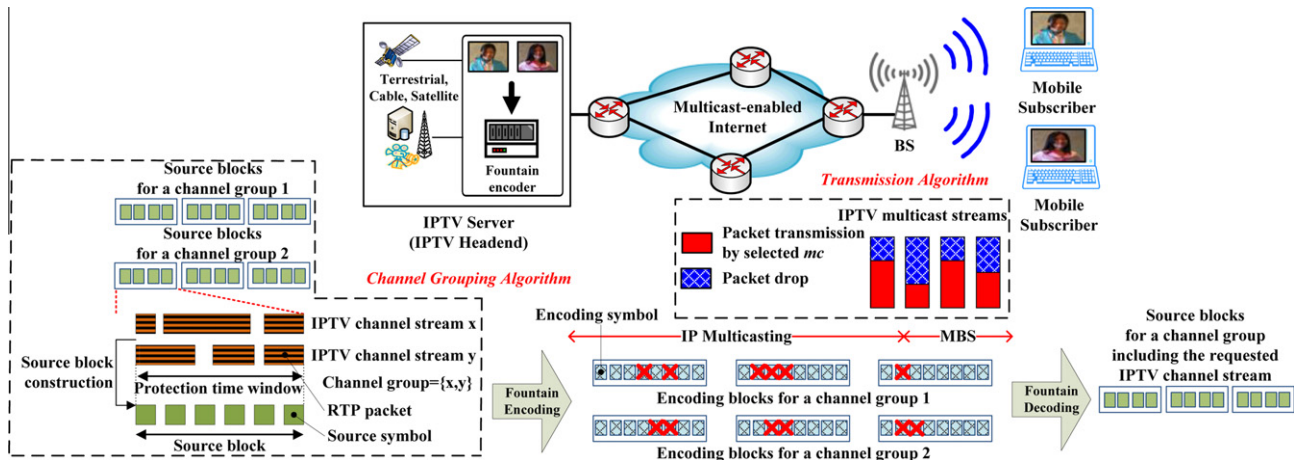

Fig. 2. Proposed fountain code-based mobile IPTV multicast system architecture.

complexity at a subscriber. These algorithms are presented in detail in Sections 3.1 and 3.2, respectively.

3.1. Proposed adaptive transmission algorithm

Mobile IPTV service requires considerable network resources compared to traditional data services, which is a big burden on wireless networks with very limited resources. Therefore, mobile IPTV multicast systems should be designed to efficiently manage resource usage of IPTV multicast streams according to the wireless link states of subscribers. It is feasible since state-of-the-art wireless networks such as WiMAX and 3GPP LTE (Long Term Evolution) networks [35] have advanced techniques to continuously monitor the wireless link states of subscribers and mechanisms to send them back to BS.

The minimum resource allocation unit on WiMAX network is called a slot, which is configured by one sub-channel over one, two, or three OFDM (Orthogonal Frequency Division Multiplex) symbols depending on the subcarrier permutation scheme. A burst is a contiguous series of slots assigned to a subscriber or a group of subscribers. WiMAX network supports some combinations of modulation scheme and coding rate at the physical layer. AMC (Adaptive Modulation and Coding) allows these combinations to be changed per burst based on the wireless link states of subscribers. A tradeoff exists between link stability and data rate. BER (Bit Error Rate) which is increased by fading and shadowing in wireless channels is reduced at the cost of bandwidth efficiency as the modulation scheme becomes sparser and coding rate lower. In contrast, a denser modulation scheme and higher coding rate provide higher data rates in a slot while they increase sensitivity to the wireless channel error and thus may transmit more packets to reconstruct source symbols of fountain code successfully. In this paper, the transmission algorithm at a BS determines the control parameters of a fountain-encoded IPTV multicast stream based on the wireless link states of subscribers in order to minimize resource usage on WiMAX network while keeping the decoding failure rate of fountain code in the tolerable range.

3.1.1. Problem description of adaptive transmission algorithm

Before presenting a detailed description, we make the following assumptions:

(1) The proposed system adopts DL PUSC (Down Link Partial Usage of SubCarrier), which is one of the subcarrier permutation schemes in the downlink sub-frame. DL PUSC is mandatory for all WiMAX network implementations and each slot consists of 24 data subcarriers by two OFDM symbols.
(2) It is assumed that a BS knows the wireless link states of subscribers by using CQI (Channel Quality Indicator) in uplink sub-frames.
(3) Fountain encoding at a server and determining of control parameters of an IPTV multicast stream at a BS are processed based on the protection time window.

The supportable combinations of modulation scheme and coding rate over WiMAX network for IPTV multicast stream transmis-

Table 1
Modulation scheme and coding rate corresponding to mc.

mc	Modulation scheme and coding rate	$R_{slot}(mc)$
1	64QAM, 3/4	216 bits
2	64QAM, 2/3	192 bits
3	16QAM, 3/4	144 bits
4	16QAM, 1/2	96 bits
5	QPSK, 3/4	72 bits
6 (MC_{Max})	QPSK, 1/2	48 bits

sion are denoted by mc, which ranges from one to MC_{Max} in ascending order according to the robustness against the wireless channel error as shown in Table 1. $R_{slot}(mc)$ is the number of transmitted data bits per slot according to mc, which is calculated by

$$R_{slot}(mc) = 24 \cdot 2 \cdot M_{mc} \cdot C_{mc}, \tag{1}$$

where M_{mc} and C_{mc} are the number of bits per symbol and the coding rate according to mc, respectively. M_{mc} depends on the selected modulation scheme, e.g. M_{mc} values of QPSK, 16QAM, and 64QAM are two, four, and six, respectively. When mc is 5, (i.e. case of QPSK modulation scheme and 3/4 coding rate), $R_{slot}(mc)$ is 72 bits (=$24 \cdot 2 \cdot 2 \cdot 3/4$). Now, the decoding failure rate of fountain codes for IPTV multicast stream transmission is calculated by

$$P_{dec\_fail}(n_{pkt}, mc, SNR_{min})$$
$$= 1 - \sum_{i=\left\lceil \frac{(1+\gamma)K}{N_{symbol}} \right\rceil}^{n_{pkt}} \binom{n_{pkt}}{i} (1 - P_{pkt\_err}(mc, SNR_{min}))^i (P_{pkt\_err}(mc, SNR_{min}))^{n_{pkt}-i} \tag{2}$$

where $\lceil x \rceil$ is the smallest integer that is not less than x, N_{symbol} is the number of encoding symbols in a packet, SNR_{min} is the minimum SNR (Signal-to-Noise Ratio) value of all subscribers that participate in an IPTV multicast stream, $P_{pkt\_err}(mc, SNR_{min})$ is the packet error rate when given mc and SNR_{min}, C is the target coding rate of fountain codes (the total number of encoding symbols in an encoding block divided by the total number of source symbols in a source block), and n_{pkt} is the number of transmitted packets for an IPTV multicast stream with a range from the minimum number of packets required for successful fountain decoding ($\lceil (1+\gamma)K/N_{symbol} \rceil$) to the total number of packets in an encoding block ($\lceil K \cdot C/N_{symbol} \rceil$). Actually, n_{pkt} becomes our control variable thanks to the employment of fountain codes, which enhances the controllability of the proposed adaptive transmission algorithm over time-varying wireless link states. Now, we can formulate our problem as follows.

3.1.1.1. Problem formulation of adaptive transmission algorithm. Determine mc and n_{pkt} to minimize

$$\left\lceil \frac{n_{pkt} \cdot (R_{packet} + R_{overhead})}{R_{slot}(mc)} \right\rceil, \tag{3}$$

subject to $P_{dec\_fail}(n_{pkt}, mc, SNR_{min}) \leqslant P_{dec\_fail}^{max}, \tag{4}$

where $P_{dec\_fail}^{max}$ is the tolerable decoding failure rate, R_{packet} is the packet size of the upper layer, and $R_{overhead}$ is the additional overhead bits at MAC layer such as MAC header and CRC (Cyclic Redundancy Check) to transmit a packet. Eq. (3) represents the slot usage on WiMAX network for IPTV multicast stream transmission and Eq. (4) takes into account the decoding failure rate constraint.

3.1.2. Determining process of mc and n_{pkt}

Fig. 3 represents the main operations at a BS. First of all, upper layer traffic is sorted into IPTV and non-IPTV flows by the classifier. MCID (Multicast Connection ID) and independent buffer are allocated to each IPTV multicast stream. Subscribers periodically report their wireless link states to the IPTV resource manager through CQI in uplink sub-frames. SNR_{min} is updated based on this CQI information. Based on the given SNR_{min}, the IPTV resource manager determines mc and n_{pkt} for each IPTV multicast stream to minimize Eq. (3) while satisfying the decoding failure rate constraint. The scheduler reserves MBS zones in downlink sub-frames to transmit n_{pkt} packets by selected mc within the protection time window through the physical module. And, the ($\lceil K \cdot C/N_{symbol} \rceil - n_{pkt}$) remaining packets are removed from the buffer.

The optimal solution $(mc^{opt}, n_{pkt}^{opt})$ of the above problem can be achieved by using the full search-based algorithm. However, the
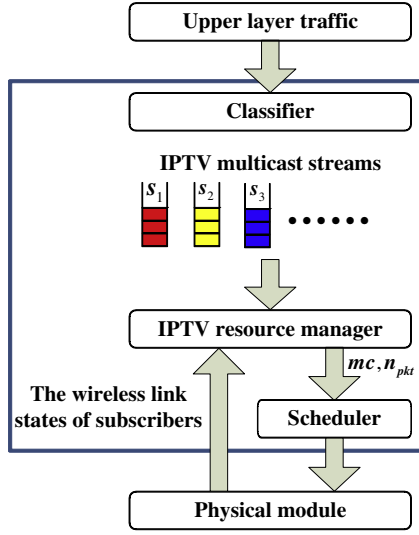
Fig. 3. Main operations at a BS.

full search-based algorithm needs a considerable amount of computation. Theoretically, the computational complexity of the full search-based algorithm is $O(MC_{Max} \cdot (\lceil K \cdot C/N_{symbol} \rceil - \lceil (1 + \gamma) \cdot K/N_{symbol} \rceil + 1))$ by checking all possible combinations of mc and n_{pkt}. To reduce the computational complexity, we consider a fast algorithm that provides the same optimal solution. Compared to the full search-based algorithm, the fast algorithm reduces the searching range by using the following useful properties.

Property 1. *When mc and SNR_{min} are given, $n^*_{pkt}(mc, SNR_{min})$ is set to the minimum value among possible n_{pkt} values that satisfy the decoding failure rate constraint, i.e.*

$$n^*_{pkt}(mc,SNR_{min}) = \arg\min_{n_{pkt}} d(n_{pkt}, mc, SNR_{min}) \text{ for } \left\lceil \frac{(1+\gamma)K}{N_{symbol}} \right\rceil \leqslant n_{pkt} \leqslant \left\lceil \frac{K \cdot C}{N_{symbol}} \right\rceil,$$

$$\text{where } d(n_{pkt}, mc, SNR_{min}) = \begin{cases} P^{max}_{dec\_fail} - P_{dec\_fail}(n_{pkt}, mc, SNR_{min}) \\ \quad : \text{if } P_{dec\_fail}(n_{pkt}, mc, SNR_{min}) \leqslant P^{max}_{dec\_fail}, \\ \infty : \text{otherwise}. \end{cases} \quad (5)$$

Property 2. *A denser modulation scheme and higher coding rate may increase the number of transmitted packets to satisfy the decoding failure rate constraint because they increase sensitivity to the wireless channel error. In other words, if $mc_i > mc_j$, $n^*_{pkt}(mc_i, SNR_{min}) \leqslant n^*_{pkt}(mc_j, SNR_{min})$. That is, Eq. (5) can be represented by*

$$n^*_{pkt}(mc_j, SNR_{min}) = \arg\min_{n_{pkt}} d(n_{pkt}, mc_j, SNR_{min}) \text{ for } n^*_{pkt}(mc_i, SNR_{min})$$

$$\leqslant n_{pkt} \leqslant \left\lceil \frac{K \cdot C}{N_{symbol}} \right\rceil. \quad (6)$$

Based on the above properties, the fast algorithm is implemented to find the optimal solution $\left(mc^{opt}, n^{opt}_{pkt} \right)$ of the problem formulation as follows. In the fast algorithm, the required computational complexity is reduced to $O(\lceil K \cdot C/N_{symbol} \rceil - \lceil (1 + \gamma)K/N_{symbol} \rceil + 1)$.

Step 1: *Initially, $mc = 1$ and $n_{pkt} = \lceil K \cdot C/N_{symbol} \rceil$.*
Step 2: *Examine the decoding failure rate constraint by Eq. (4) with mc and n_{pkt}. If the constraint is satisfied, $n^*_{pkt}(mc, SNR_{min}) = n_{pkt}$ and then repeat Step 2 with $n_{pkt} = n_{pkt} - 1$. Otherwise, go to Step 3.*
Step 3: *If mc is not equal to MC_{Max}, return to Step 2 with $mc = mc + 1$.*

Step 4: *Select a combination to show the best performance through Eq. (7).*

$$mc^{opt} = \arg \min_{1 \leqslant mc \leqslant MC_{Max}} \left(\left\lceil \frac{n^*_{pkt}(mc, SNR_{min}) \cdot (R_{packet} + R_{overhead})}{R_{slot}(mc)} \right\rceil \right),$$

$$n^{opt}_{pkt} = n^*_{pkt}(mc_{opt}, SNR_{min}). \quad (7)$$

3.2. Proposed channel grouping algorithm

The channel zapping time is adopted in this paper as a performance measure of seamless channel change, which is generally defined as the time duration until a subscriber watches the chosen channel after the channel selection. Compared to traditional broadcasting services, the relatively long channel zapping time of mobile IPTV multicast systems is a big obstacle and should be addressed for successful deployment [36–38]. In the proposed mobile IPTV multicast system, the channel zapping time consists of buffering delay, fountain decoding processing delay, and video decoding delay. Fig. 4 represents an example of a channel change scenario. It is assumed that a subscriber sends the channel change request by clicking buttons on a mobile device. If the requested IPTV channel stream is already broadcasted, the mobile device can directly receive the fountain encoding symbols. Otherwise, it sends the channel request message to a BS, and then the BS will broadcast the IPTV multicast stream including the requested IPTV channel stream. In this case, a processing delay is incurred, but it is negligible compared to the other delay factors. After the start of receiving the encoding symbols, the mobile device should wait for a while until a sufficient number of encoding symbols are available for successful fountain decoding (this is called a buffering delay). If the number of encoding symbols arriving between the channel change request time and the starting time of the next protection time window is not big enough for successful fountain decoding, the buffer must be cleared for the encoding symbols in the next protection time window. Otherwise, the mobile device can get source symbols from the encoding symbols accumulated at the buffer with some fountain decoding processing delay. It is another benefit of the employment of fountain codes. Video decoding delay consists of decoding processing delay and GOP structure delay. The GOP structure delay is inevitable for continuous video playback because compressed video stream can be decoded and displayed on a mobile device starting with a key frame (I (Intra) frame is a key frame in the MPEG-2 codec or IDR (Instantaneous Decoder Refresh) frame in the H.264/AVC codec).

In this paper, the channel grouping algorithm is designed to reduce the channel zapping time at the price of the increased processing complexity at a subscriber. First of all, IPTV channel streams are categorized into several channel groups. The IPTV channel streams in a channel group during the protection time window generate a source block. Then, fountain encoding is performed for the source block, and the resulting encoding symbols are inserted into packets. It has some advantages with respect to
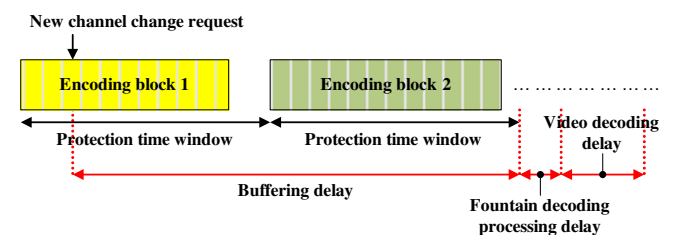

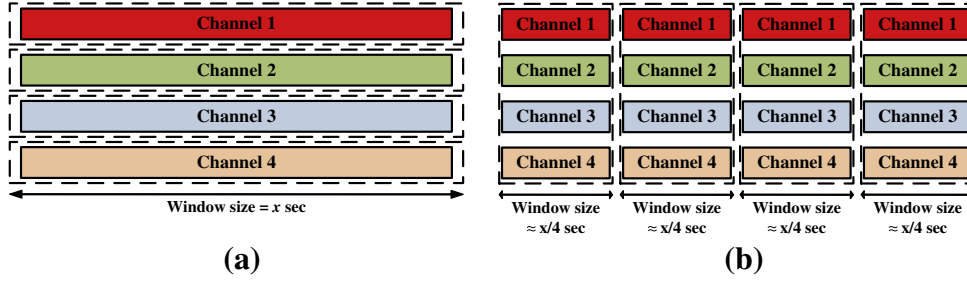**Fig. 4.** An example of a channel change scenario.

Fig. 5. Examples of channel grouping: (a) a single IPTV channel stream and (b) four IPTV channel streams per channel group.

channel zapping time. For example, when a channel group includes only a single IPTV channel stream as shown in Fig. 5(a), its protection time window size is fixed at x seconds to satisfy the source symbol constraint under the assumption that the symbol size is pre-determined, and thus the buffering delay increases with x seconds. On the other hand, when a channel group includes multiple IPTV channel streams as shown in Fig. 5(b), the source symbol constraint can be met with the protection time window size of the quarter of x seconds. Also, when a subscriber switches to another IPTV channel stream in the same channel group, instant channel switching is possible except for the video decoding delay because the requested IPTV channel stream is available at the buffer. However, the additional processing at the subscriber is inevitable since additional non-watching IPTV channel streams are included in the channel group.

Now, we consider an effective tradeoff between the channel zapping time and the processing complexity at the subscriber. In an engineering sense, when a channel group includes more IPTV channel streams with high channel selection preferences, the channel zapping time significantly decreases while the processing complexity remarkably increases at the subscriber, and vice versa.

3.2.1. Problem description of channel grouping algorithm

Before presenting a detailed description, we make the following assumptions:

(1) Source symbol size (T_{symbol}) is fixed during session (it is recommended that T_{symbol} is set in the form of 2^n ($n = 3, 4, 5, \ldots$) for fountain decoding complexity advantage [18]).
(2) It is assumed that subscribers choose their channels independently (if more channel selection pattern information is available, we can achieve better channel grouping).

Some symbolic descriptions are given in the following.

$$\vec{r} = (r_1, r_2, \ldots, r_{N_{CH}}),$$
$$\vec{p} = (p_1, p_2, \ldots, p_{N_{CH}}),$$

where r_i is the required bandwidth for the ith channel stream, p_i is the ith channel selection preference, and N_{CH} is the total number of IPTV channel streams. And, channel grouping vector is defined by

$$\vec{s} = (s_1, s_2, \ldots, s_{m_{gr}}),$$

where m_{gr} is the number of channel groups, and s_j is the jth channel group to satisfy $\bigcup_{j=1}^{m_{gr}} s_j = U = \{1, 2, \ldots, N_{CH}\}$ and $\bigcap_{j=1}^{m_{gr}} s_j = \phi$. To reduce the channel zapping time, the protection time window size for each channel group must be the lowest value possible while satisfying the source symbol constraint. Also, it must be larger than the key frame interval to avoid useless video decoding operations. Now, the protection time window size for the jth channel group is calculated by

$$t_{window}(s_j) = \arg\min_t g(s_j, t) \text{ for } T_{window\_min} \leqslant t < \infty, \quad (8)$$

$$g(s_j, t) = \begin{cases} \sum_{i \in s_j} \left\lceil \frac{r_i \cdot t}{T_{symbol}} \right\rceil - K_{\min} : & \text{if } \sum_{i \in s_j} \left\lceil \frac{r_i \cdot t}{T_{symbol}} \right\rceil \geqslant K_{\min}, \\ \infty : & \text{otherwise}, \end{cases}$$

where $K_{\min}$ is lower bound for the source symbol constraint and T_{window_min} is the key frame interval given by the video codec. Now, when the hth channel is selected from the ith channel, the required maximum zapping delay is defined by

$$d_{i,h}(\vec{s}) = \begin{cases} d_{video\_decoding} : & \text{if } i, h \in s_j, \\ 2t_{window}(s_j) + d_{fountain\_decoding} + d_{video\_decoding} : & \text{if } i \notin s_j, h \in s_j, \end{cases}$$

where $d_{fountain\_decoding}$ and $d_{video\_decoding}$ are the fountain decoding processing delay and the video decoding delay, respectively. Now, we can formulate the given channel grouping problem as follows.

3.2.1.1. Problem formulation of channel grouping algorithm. Determine $\vec{s}$ to minimize

$$\sum_{j=1}^{m_{gr}} \sum_{i \in s_j} p_i \cdot \left(\frac{K(s_j) \cdot C}{t_{window}(s_j)} \right), \quad (9)$$

subject to $\sum_{i=1}^{N_{CH}} \sum_{h \in U - \{i\}} p_i \cdot \frac{p_h}{1 - p_i} \cdot d_{i,h}(\vec{s}) \leqslant D_{\max}, \quad (10)$

where $D_{\max}$ is the tolerable upper bound of the maximum average zapping time and $K(s_j)$ is the total number of source symbols in a source block for the jth channel group, which is calculated by

$$K(s_j) = \sum_{i \in s_j} \left\lceil \frac{r_i \cdot t_{window}(s_j)}{T_{symbol}} \right\rceil. \quad (11)$$

The cost function Eq. (9) represents the average number of encoding symbols required for successful fountain decoding per second at a subscriber (it is called the average processing complexity in the following) and Eq. (10) takes into account the channel zapping time constraint.

3.2.2. Determining process of $\vec{s}$

Now, we specify how to determine the total number of channel groups and to select IPTV channel streams to be included in each channel group based on channel selection preferences. Basically, the full search-based algorithm can be employed to obtain optimal channel groups. However, the full search-based algorithm requires a considerable amount of computation complexity. In general, when the total number of IPTV channel streams becomes larger, the number of possible channel grouping combinations increases dramatically, which is calculated by

$$S(N_{CH}) = \sum_{j=1}^{N_{CH}} \frac{1}{j!} \sum_{i=0}^{j-1} (-1)^i \binom{j}{i} (j - i)^{N_{CH}}. \quad (12)$$

To reduce the computational complexity, we propose a fast algorithm that provides a good solution of the above constrained optimization problem. The proposed fast algorithm utilizes the Genetic algorithm [39,40] to obtain a near-optimal solution with low computational complexity. The Genetic algorithm gradually improves all potential solutions through biological evolutionary processes such as crossover and mutation operations. It has been successfully employed in a wide variety of application areas to solve many difficult optimization problems. In addition, it can be effectively implemented on parallel computers to solve large-scale problems since it is an implicitly parallel technique. In the proposed algorithm, the Genetic algorithm is applied to $\vec{g}$ defined by

$$\vec{g} = (g_1, g_2, \ldots, g_{N_{CH}}),$$

where $g_i(1 \leqslant g_i \leqslant m_{gr} \leqslant N_{CH})$ is the channel group number that the ith channel stream is included. It is straightforward to obtain the corresponding $\vec{s}$ from $\vec{g}$. In the following, we present the proposed fast algorithm in detail.

Step 1 (Initialization): To configure the initial population, $N_{population}$ vectors are generated by assigning $[1, N_{CH}]$ to g_i $(1 \leqslant i \leqslant N_{CH})$ randomly.

Step 2 (Crossover): Some parent vectors are selected from the population with a crossover probability. By crossover operation, offspring vectors are generated from the selected parent vectors. There are several types of crossover operation including k-point, uniform, and diagonal. In this paper, we use uniform crossover operation. As shown in Fig. 6(a), uniform crossover operation generates a uniformly random mask and then exchanges relative elements between parent vectors according to the mask.

Step 3 (Mutation): Mutation operation is applied to offspring vectors with a small mutation probability. Mutation operation forces random changes for elements in an offspring vector to avoid the local minimum. There are several types of mutation operation including uniform and Gaussian. In this paper, uniform mutation operation is chosen. As shown in Fig. 6(b), uniform mutation operation selects the target element in the offspring vector and then replaces it with a random channel group number.

Step 4 (Repair): Some offspring vectors may violate the above given constraints. Thus, repair operation is executed to examine whether or not the resulting vectors are feasible solution, and if not, they are fixed. This technique depends on the given problem and is no standard design method. In this paper, repair operation is implemented by simple merge and spilt operations for channel groups. That is, when the vector violates the channel zapping time constraint, channel groups are randomly chosen and combined until the constraint is satisfied. Also, if the protection time window size of a channel group is smaller than T_{window_min}, the corresponding channel group is divided into two new channel groups.

Step 5 (Selection): Evaluate the selected parent vectors and offspring vectors by Eq. (9). Then, select vectors to show the best performance, which are used as the population in the next generation.

Step 6 (Termination): If the iteration is conducted by a predetermined number or the change rate of population is less than a predetermined thresholder, stop. Otherwise, go to Step 2.

4. Experimental results

Experimental results are provided to demonstrate the performance of the proposed fountain code-based mobile IPTV multicast system. OPNET [41] and H.264/AVC JM [42] are employed during the experiment. The experimental environment is set up as follows:

(1) The system architecture for the experiment is given in Fig. 2. It is assumed that throughput from an IPTV server to a WiMAX BS is 1 Gbps and IP multicasting is supported over the wired network. The number of subscribers is set to 20 and they are moving by a random-way point model in the cell.

(2) 10 IPTV channel streams are serviced to subscribers. Each IPTV channel stream is encoded at 30 frames per second and its target bandwidth is set to 204,800 bps. A GOP consists of 15 frames (IPPPPPPPPPPPPPP) and the key frame interval is 1.0 s. For error concealment, the video decoder uses the frame copy from the previous frame to improve the subjective human visual perceptual quality. It is assumed that channel change requests of each subscriber arrive according to Poisson distribution with an expected inter-arrival time of $1/\lambda$, where λ is the request rate and set to 1/60. Channel selection preferences are assumed to be Uniform or Zipf distribution. When the channel selection preferences are Zipf-distributed, the preference of each channel is calculated by

$$p_i = f_i / \sum_{k=1}^{N_{CH}} f_k \quad \text{for} \quad 1 \leqslant i \leqslant N_{CH},$$

where $f_i = 1/i^{\theta}$ and the skew factor (θ) is set to 0.729.

(3) Raptor code [29], which is a type of fountain code, is employed during the experiment. In the encoding process, the source symbol size and the packet payload size are set to 128 and 512 bytes, respectively, as recommended by the 3GPP MBMS specifications [34] (i.e., one packet consists of four encoding symbols). The decoding overhead is two symbols, and K_{min} and T_{window_min} are set to 1024 and 1.0 s, respectively [43].

(4) The parameters including the crossover probability (p_{cross}), the mutation probability ($p_{mutation}$), and the population size ($N_{population}$) can affect to the convergence time and the solution quality in the proposed channel grouping algorithm as shown in Fig. 7. Actually, the parameters analysis of the Genetic algorithm is beyond the scope of this paper. Consid-
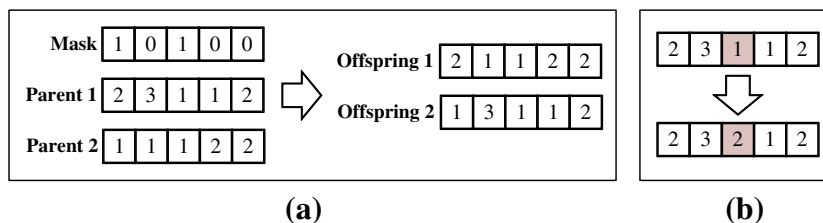

Fig. 6. An example of crossover and mutation operations: (a) uniform crossover operation and (b) uniform mutation operation.
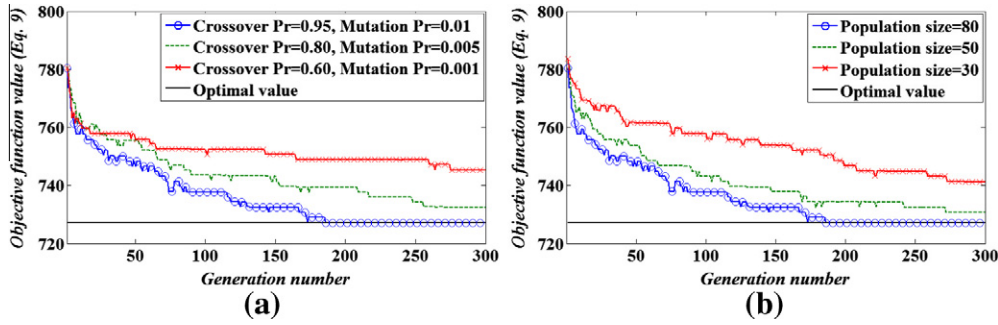
Fig. 7. Performance comparison according to genetic parameters in the proposed channel grouping algorithm when $D_{\max}$ is 3 s: (a) p_{cross} and $p_{mutation}$ with $N_{population}$ = 80, and (b) $N_{population}$ with p_{cross} = 0.95 and $p_{mutation}$ = 0.01.

ering the convergence time and the solution quality in the proposed channel grouping algorithm, p_{cross}, $p_{mutation}$, and $N_{population}$ are empirically set to 0.95, 0.01, and 80, respec-

Table 2
OFDMA parameters of WiMAX network.

Parameters	Value
Channel bandwidth	20 MHz
FFT size	2048-FFT PUSC
Number of data subcarriers	1440
Number of pilot subcarriers	240
Number of null and guardband subcarriers	368
Cyclic prefix	1/8
Oversampling rate	28/25
Subcarrier frequency spacing	10.94 kHz
Useful symbol time	91.4 μs
Guard time	11.4 μs
OFDM symbol duration	102.86 μs
Number of OFDM symbols in 5 ms frame	48
Uplink/downlink boundary	1:3

Table 3
Channel groups according to $D_{\max}$.

Channel selection preferences	$D_{\max}$	$\vec{s}$	m_{gr}
Uniform distribution	12.0	{1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10}	10
	9.0	{1,2}, {3,4}, {5,6}, {7}, {8}, {9}, {10}	7
	7.0	{1,2}, {3,4}, {5,6}, {7,8}, {9}, {10}	6
	5.0	{1,2}, {3,4}, {5,6,7}, {8,9,10}	4
	3.0	{1,2,3,4,5}, {6,7,8,9,10}	2
Zipf distribution	12.0	{1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10}	10
	9.0	{1}, {2,3}, {4}, {5}, {6,8}, {7}, {9}, {10}	8
	7.0	{1,2}, {3,5}, {4}, {6,10}, {7}, {8}, {9}	7
	5.0	{1,2}, {3,4}, {5,6,7}, {8,9,10}	4
	3.0	{1,2,3}, {4,5,6,10}, {7,8,9}	3

tively. The supportable combinations of modulation scheme and coding rate over WiMAX network are set as shown in Table 1. $R_{overhead}$ is 14 bytes, $P_{dec\_fail}^{max}$ is 0.01, and $d_{fountain\_decoding}$ and $d_{video\_decoding}$ are set to 0.1 and 1.1 s, respectively.

(5) As a path loss model, a fixed suburban (Erceg) model [44] is employed with a conservative terrain model which accounted for hilly terrain with moderate-to-heavy tree densities (Terrain A model). As a multipath model, ITU-R Pedestrian B channel model with 3 km/h [45] is employed during the experiment. OFDMA parameters of WiMAX network are set as shown in Table 2.

4.1. Performance of the proposed channel grouping algorithm

First of all, we examine the performance of the proposed channel grouping algorithm according to $D_{\max}$. The target coding rate of fountain codes is fixed to 1.1, and the modulation scheme and coding rate for IPTV multicast stream transmission is set to (QPSK,1/2) to guarantee a stable wireless link to subscribers far away from a BS. The experimental results are summarized in Table 3 and Fig. 8. As $D_{\max}$ becomes smaller, it is apparently observed that the number of channel groups decreases because the number of IPTV channel streams in each channel group increases to satisfy the channel zapping time constraint. As a result, the maximum average channel zapping time decreases while the average processing complexity increases as shown in Fig. 8. For example, when $D_{\max}$ is set to 9 s and the channel selection preferences are Uniform-distributed, it is observed that the resulting number of channel groups is 7, the maximum average channel zapping time is 7.95 s, and the average processing complexity is 352.00 symbols/s. On the other hand, when $D_{\max}$ is changed to 5 s, the proposed channel grouping algorithm constructs four channel groups to satisfy the channel zapping time constraint, and the average processing complexity increases to 573.16 symbols/s. When the channel
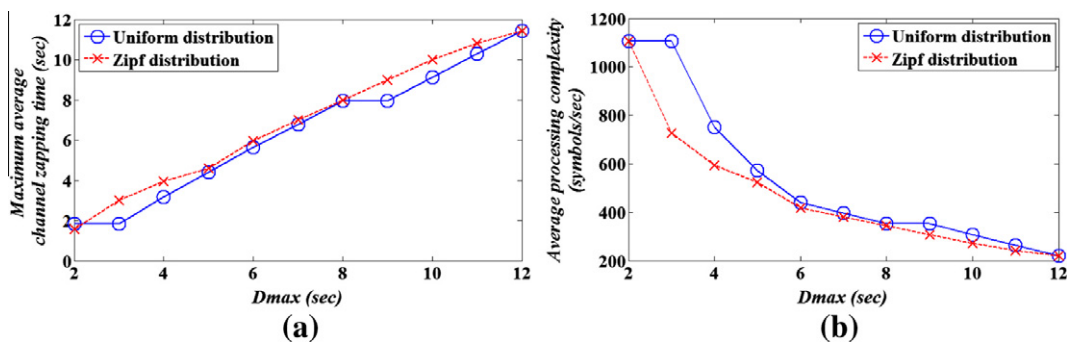

Fig. 8. Performance of the proposed channel grouping algorithm: (a) the maximum average channel zapping time and (b) the average processing complexity.

selection preferences are the Zipf distribution, the same phenomenon is exactly observed. Furthermore, the average processing complexity plot of Zipf distribution always lies below that of Uniform distribution as shown in Fig. 8.

In the following, the proposed channel grouping algorithm is compared with the SCCG (a Single IPTV Channel stream per Channel Group) algorithm [15] and the two Greedy algorithms [46]: (1) Greedy algorithm I (IPTV channel streams with higher channel selection preferences are allocated to a larger channel group) and (2) Greedy algorithm II (IPTV channel streams with lower channel selection preferences are allocated to a larger channel group). For a fair comparison, the number of channel groups and the number of IPTV channel streams in each channel group of Greedy algorithms are same as those of the proposed channel grouping algorithm. Actually, when the number of channel groups and the size of each channel group are given, Greedy algorithm I and Greedy algorithm II provide the lower bound and the upper bound of the channel zapping time, respectively. In contrast, Greedy algorithm I and Greedy algorithm II give the upper bound and the lower bound

of the processing complexity, respectively. When D_{max} is set to 7 s, the experimental results are summarized in Table 4. In the SCCG algorithm, the average processing complexity is the lowest, but the average channel zapping time of some subscribers is much larger than D_{max}. Greedy algorithm I shows lower channel zapping time, but relatively higher average processing complexity than the proposed channel grouping algorithm. Greedy algorithm II violates the channel zapping time constraint although its average processing complexity is less than that of the proposed channel grouping algorithm. On the other hand, the proposed channel grouping algorithm makes channel groups on the basis of channel selection preferences in order to minimize the average processing complexity while satisfying the channel zapping time constraint. Consequently, the proposed channel grouping algorithm keeps the average channel zapping time of all subscribers below D_{max} at the cost of a slightly increased average processing complexity.

Finally, we consider the computational complexity of the proposed channel grouping algorithm. The experiment is performed with a laptop computer running on Windows XP SP3 with a

Table 4
Performance comparison with various channel grouping algorithms.

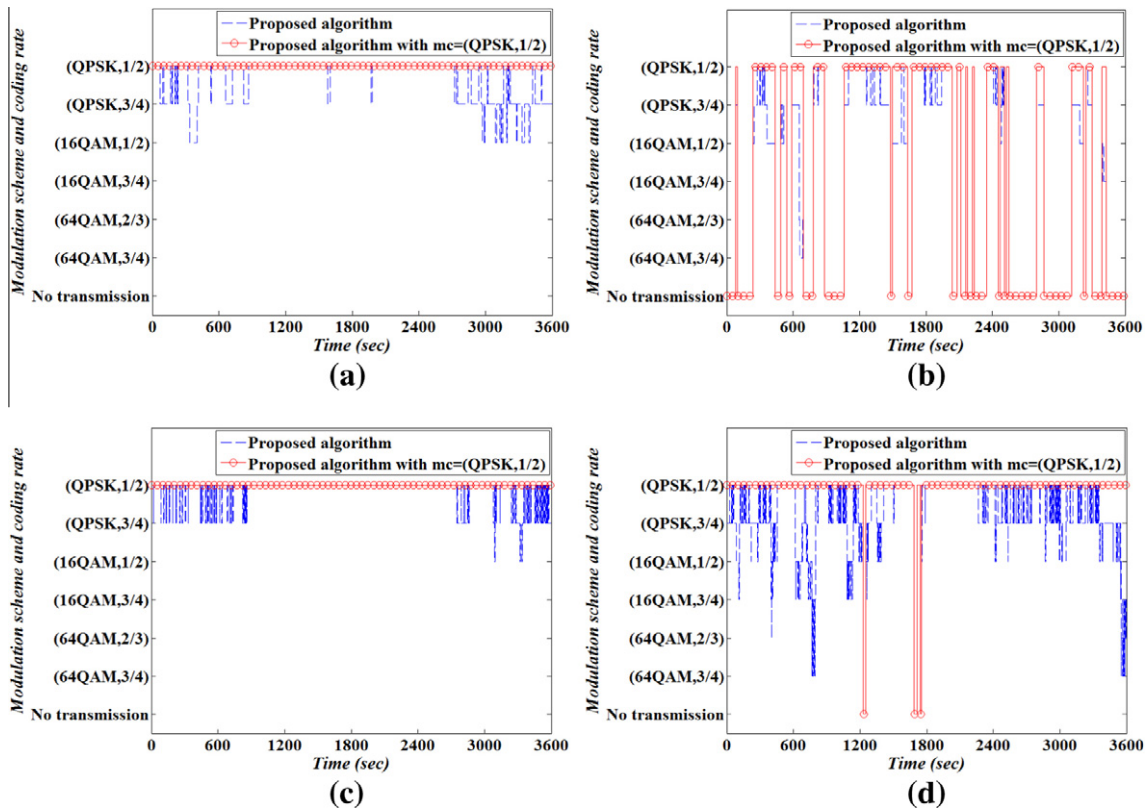| Channel selection preferences	Channel grouping algorithms	Average processing complexity of all subscribers (symbols/s)	Maximum average channel zapping time (s)	Average channel zapping time of all subscribers (s)
Uniform distribution	SCCG algorithm	219.50	11.30	8.85
	Proposed channel grouping algorithm	394.38	6.65	5.36
Zipf distribution	SCCG algorithm	219.51	11.30	8.86
	Greedy algorithm I	397.60	6.19	5.01
	Greedy algorithm II	294.32	9.33	7.29
	Proposed channel grouping algorithm	384.31	6.90	5.35

Fig. 9. Modulation scheme and coding rate: (a) the channel group with the highest group selection preference and (b) the channel group with the lowest group selection preference when D_{max} is 12 s, and (c) the channel group with the highest group selection preference and (d) the channel group with the lowest group selection preference when D_{max} is 3 s.

2.0 GHz Intel Pentium processor and 2 GB RAM. It is observed during the experiment that the proposed channel grouping algorithm can provide the same (optimal) solution as the full search-based algorithm. However, the proposed channel grouping algorithm needs a much smaller computational complexity than the full search-based algorithm. The average execution time is 436.833 s in the case of the full search-based algorithm and 7.740 s in the case of the proposed channel grouping algorithm.

4.2. Performance of the proposed adaptive transmission algorithm

In this section, we examine the performance of the proposed adaptive transmission algorithm. During the experiment, the target coding rate of fountain codes is fixed at 2.0 to effectively handle dynamically time-varying wireless link states by generating a sufficient number of encoding symbols, and channel groups are made by the proposed channel grouping algorithm in Section 4.1 when D_{max} is set to 3, 7, and 12 s under the Zipf distribution. Figs. 9 and 10 represent the control parameters adaptation process of the proposed adaptive transmission algorithm when 20 subscribers are moving in the cell. It is observed that variations of SNR_{min} value increase and the resulting curves of control parameters significantly fluctuate as the group selection preference (the sum of channel selection preferences in a channel group) becomes lower. The experimental results are summarized in Table 5. The average slot usage on WiMAX network and the decoding failure rate of fountain codes at the participating subscriber with the minimum SNR value are measured. To show the superior performance of the proposed adaptive transmission algorithm, two cases are considered: (1) Case I that mc is (QPSK,1/2) and n_{pkt} is the maximum n_{pkt} value of the proposed adaptive transmission algorithm with $mc = $ (QPSK,1/2), and (2) Case II that mc is (QPSK,1/2) and n_{pkt} is
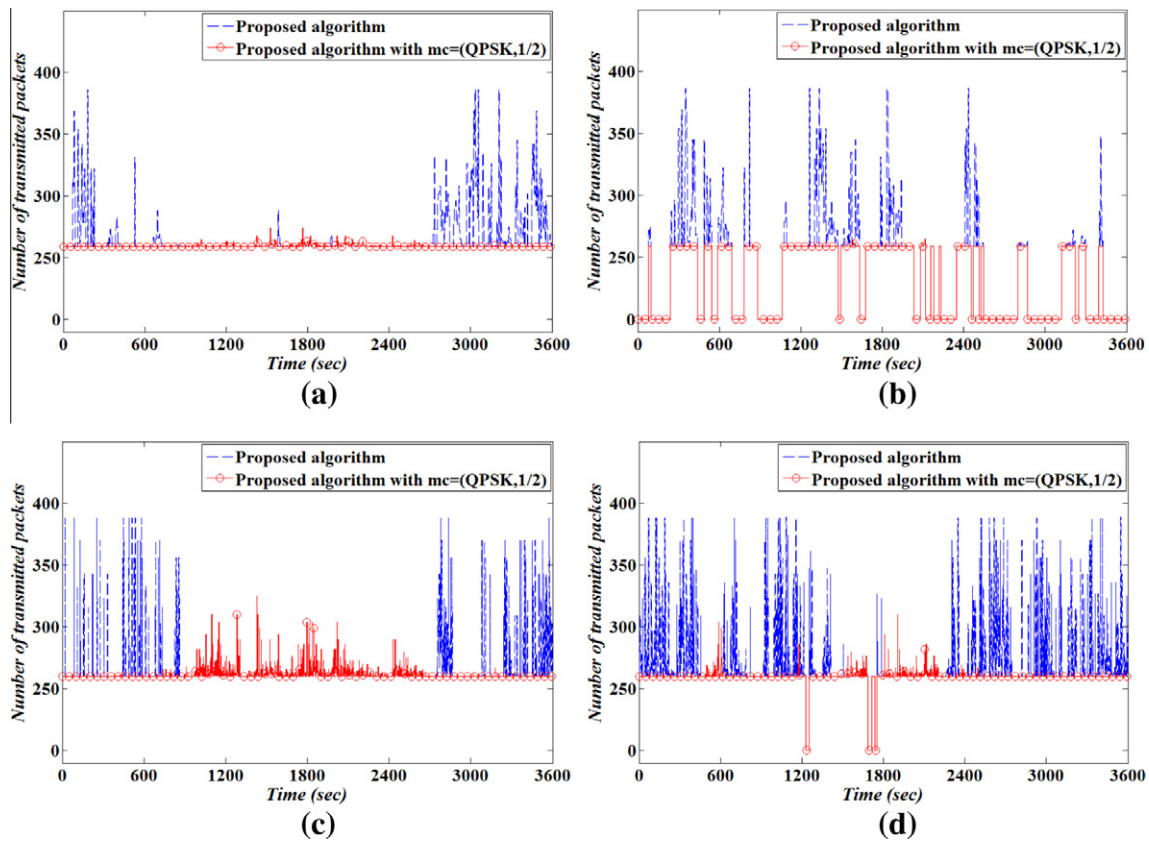

Fig. 10. The number of transmitted packets: (a) the channel group with the highest group selection preference and (b) the channel group with the lowest group selection preference when D_{max} is 12 s, and (c) the channel group with the highest group selection preference and (d) the channel group with the lowest group selection preference when D_{max} is 3 s.

Table 5
DFR (decoding failure rate) and ASU (average slot usage) comparison with various transmission algorithms.

| Channel groups	Transmission algorithms	D_{max} = 12 s		D_{max} = 7 s		D_{max} = 3 s	
		DFR (%)	ASU	DFR (%)	ASU	DFR (%)	ASU
Channel group with the highest group selection preference	Case I	0.00	25,117	0.00	26,859	0.00	29,792
	Case II	3.55	23,742	4.92	23,834	6.14	24,017
	Proposed algorithm with mc = (QPSK,1/2)	0.67	23,781	0.69	23,837	0.69	23,989
	Proposed algorithm	0.68	21,669	0.68	22,626	0.69	23,409
Channel group with the lowest group selection preference	Case I	0.00	12,854	0.00	14,613	0.00	27,918
	Case II	1.03	12,563	1.74	13,914	2.99	23,504
	Proposed algorithm with mc = (QPSK,1/2)	0.69	12,566	0.66	13,923	0.68	23,469
	Proposed algorithm	0.69	10,183	0.67	10,849	0.66	19,853
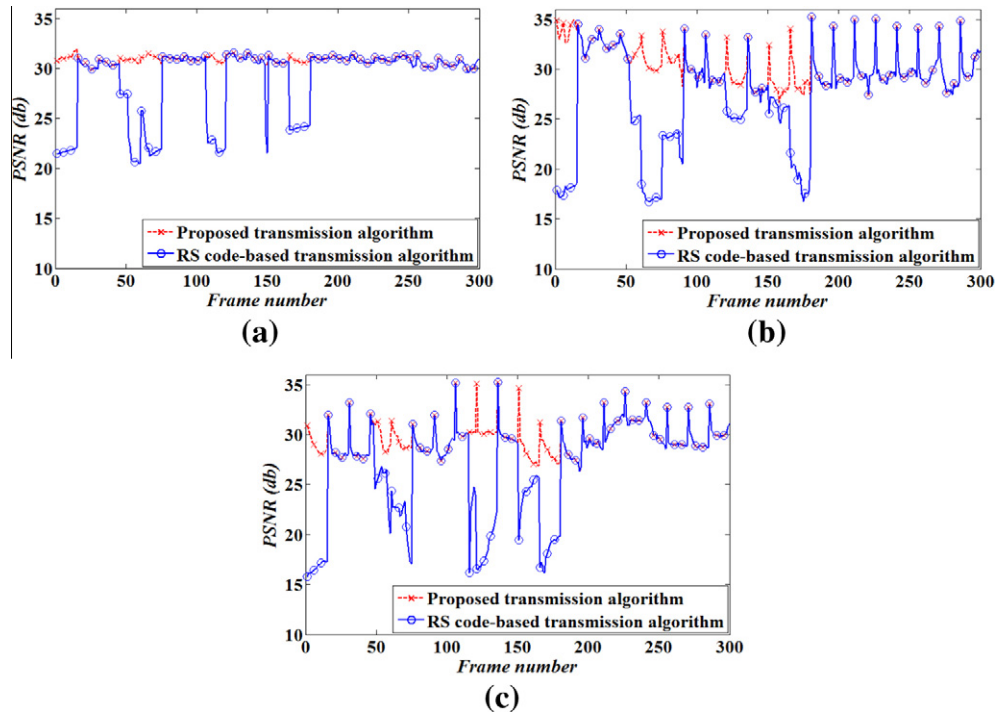
Fig. 11. PSNR plots according to frame number: (a) City, (b) Crew, and (c) Soccer test videos.

the average n_{pkt} value of the proposed adaptive transmission algorithm with mc = (QPSK,1/2). In Case I, the decoding failure rate is the lowest at the cost of slot usage on WiMAX network. Case II violates the decoding failure rate constraint and consumes relatively high WiMAX network resources compared to the proposed adaptive transmission algorithm. On the other hand, the proposed adaptive transmission algorithm minimizes the slot usage on Wi-MAX network while satisfying the decoding failure rate constraint by adjusting the control parameters adaptively to the wireless link states of subscribers.

Finally, the proposed adaptive transmission algorithm is compared with the RS (Reed-Solomon) code-based transmission algorithm. When D_{max} is 12 s, the visual quality is measured at a subscriber with the minimum SNR value that is participating in a channel group with the highest group selection preference. The test video sequences are City, Crew, and Soccer of CIF-size (352×288). For a fair comparison, the symbol size and mc are same for two FEC codes, and the target coding rate of RS code is fixed to a constant between the average number of transmitted

encoding symbols and the number of source symbols of the proposed adaptive transmission algorithm. And, the source block size of RS code is relatively smaller than that of fountain code due to the quadratic decoding complexity of RS code. In fact, (255, 245, 128) RS code is used during the experiment, and thus the source block size is 31,360 bytes. As shown in Fig. 11, the proposed adaptive transmission algorithm is more robust against the wireless channel error, and avoids sudden quality degradation compared to the RS code-based transmission algorithm. The proposed adaptive transmission algorithm supports a stable IPTV service by dynamic adaptation of the control parameters considering the wireless link states of subscribers. In addition, the proposed adaptive transmission algorithm can sustain a larger source block size by using fountain code than the RS code-based transmission algorithm, which can reduce the sensitivity to burst errors over the wireless channel. The subjective video quality comparison is presented in Fig. 12. The subjective video quality of the proposed adaptive transmission algorithm is obviously better than the RS code-based transmission algorithm.
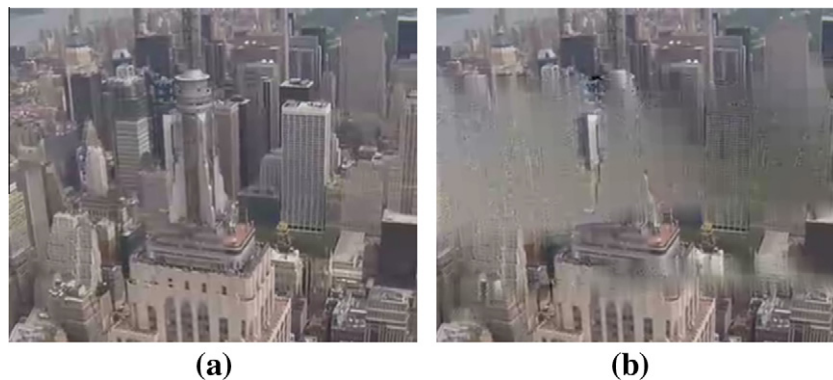

Fig. 12. Subjective video quality comparison when the test video sequence is City: (a) the proposed adaptive transmission algorithm and (b) RS code-based transmission algorithm.

5. Conclusions

In this paper, we have proposed a novel fountain code-based mobile IPTV multicast system architecture over WiMAX network. In the proposed system, the adaptive transmission algorithm at a BS is designed to provide stable IPTV service to all subscribers with minimum resource usage on WiMAX network, and the channel grouping algorithm at a server pursues an effective tradeoff between channel zapping time and processing complexity at a subscriber. Experimental results have shown that the proposed adaptive transmission algorithm has achieved very good network efficiency while satisfying the decoding failure rate constraint. It has also been observed during the experiment that the proposed channel grouping algorithm has achieved low processing complexity on the average while satisfying the channel zapping time constraint at a subscriber.

Although WiMAX network is considered as the target wireless access network in this paper, with the some minor modifications, the proposed system can be easily extended to any advanced wireless networks supporting broadband wireless access including 3GPP LTE, 3GPP LTE-Advanced, and IEEE 802.16m. This will be examined in our future work.

References

[1] J.Y. Kim, J.H. Hahm, Y.S. Kim, J.K. Choi, NGN architecture for IPTV service without effect on conversational services, in: International Conference on Advanced Communication Technology, vol. 1, 2006, pp. 465–469.
[2] A. Harris, G. Ireland, Enabling IPTV: What Carriers Need to Know to Succeed, International Data Corporation, May 2005, White Paper.
[3] H.C. Joo, H. Song, Wireless link state-aware H.264 MGS coding-based mobile IPTV system over WiMAX network, Journal of Visual Communication and Image Representation 21 (2010) 89–97.
[4] J. She, F. Hou, P.H. Ho, L.L. Xie, IPTV over WiMAX: key success factors, challenges, and solutions, IEEE Communications Magazine 45 (8) (2007) 87–93.
[5] IEEE 802.16 TGe Working Document, (Draft Standard) – Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, 802.16e/D4, August 2004.
[6] IEEE Standard 802.16-2004, IEEE Standard for Local and Metropolitan Area Networks-Part 16: Air Interface for Fixed Broadband Wireless Access Systems, 2004.
[7] 3GPP, Multimedia Broadcast/Multicast Service; Architecture and functional description (Release 6), 3GPP Tech. Specification TS 23.246 v6.9.0, December 2005.
[8] 3GPP2, Interoperability Specification (IOS) for Broadcast Multicast Service (BCMCS), 3GPP2 TSG-A Specification A.S0019-0 v1.0, November 2004.
[9] T. Jiang, W. Xiang, H. Chen, Q. Ni, Multicast broadcasting services support in OFDMA-based WiMAX systems, IEEE Communications Magazine 45 (8) (2007) 78–86.
[10] J. Wang, M. Venkatachalam, Y. Fang, System architecture and cross-layer optimization of video broadcast over WiMAX, IEEE Journal on Selected Areas in Communications 25 (4) (2007) 712–721.
[11] ETSI, Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H), ETSI EN 302304 v1.1.1, November 2004.
[12] W. Stallings, Data and Computer Communications, seventh ed., Prentice Hall, India, 2004.
[13] N. Nikaein, H. Labiod, C. Bonnet, MA-FEC: A QoS-based adaptive FEC for multicast communication in wireless networks, in: IEEE International Conference on Communications, 2000, pp. 954–958.
[14] A. Nafaa, T. Taleb, L. Murphy, Forward error correction strategies for media streaming over wireless networks, IEEE Communications Magazine 46 (1) (2008) 72–79.
[15] M. Luby, T. Stockhammer, M. Watson, IPTV systems, standards and architectures: Part II – Application layer FEC in IPTV services, IEEE Communications Magazine 46 (2008) 94–101.
[16] Y. Park, Y. Seok, N. Choi, Y. Choi, J.M. Bonnin, Rate-adaptive multimedia multicasting over IEEE 802.11 wireless LANs, in: IEEE Consumer Communications and Networking Conference, vol. 1, pp. 178–182, January 2006.
[17] H. Won, H. Cai, D.Y. Eun, K. Guo, A. Netravali, I. Rhee, K. Sabnani, Multicast scheduling in cellular data networks, IEEE Transactions on Wireless Communications 8 (9) (2009) 4540–4549.
[18] J. Zfzal, T. Stockhammer, T. Gasiba, W. Xu, Video streaming over MBMS: a system design approach, Journal of Multimedia 1 (5) (2006) 25–35.
[19] M. Choi, M. Samokhina, K. Moklyuk, S. Choi, J. Heo, S.J. Oh, VPAL: video packet adaptation layer for reliable video multicast over IEEE 802.11n WLAN, Computer Communications 33 (18) (2010) 2271–2281.
[20] ö. Alay, T. Korakis, Y. Wang, S. Panwar, Dynamic rate and FEC adaptation for video multicast in multi-rate wireless networks, ACM/Springer Mobile Networks and Applications 15 (3) (2010) 425–434.
[21] T.A. Courtade, R.D. Wesel, A cross-layer perspective on rateless coding for wireless channels, in: IEEE International Conference on Communications, pp. 1–6, June 2009.
[22] C. Cho, I. Han, Y. Jun, H. Lee, Improvement of channel zapping time in IPTV services using the adjacent groups join-leave method, in: International Conference on Advanced Communication Technology, vol. 2, 2004, pp. 971–975.
[23] J.M. Boyce, A.M. Tourapis, Fast efficient channel change [set-top box applications], in: IEEE International Conference on Consumer Electronics, January 2005, pp. 1–2.
[24] P. Siebert, T.N.M. Van Caenegem, M. Wagner, Analysis and improvements of zapping times in IPTV systems, IEEE Transactions on Broadcasting 55 (2) (2009) 407–418.
[25] Architecture of Microsoft Mediaroom, Microsoft Mediaroom Server 1.1 SP 3.2 Product Documentation.
[26] H. Schwarz, D. Marpe, T. Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard, IEEE Transactions on Circuits and Systems for Video Technology 17 (9) (2007) 1103–1120.
[27] F. Hou, L.X. Cai, P.H. Ho, A cooperative multicast scheduling scheme for multimedia services in IEEE 802.16 Networks, IEEE Transactions on Wireless Communications 8 (3) (2009) 1508–1519.
[28] M. Luby, LT Codes, in: Annual Symposium on Foundations of Computer Science, 2002, pp. 271–280.
[29] A. Shokrollahi, Raptor codes, IEEE Transactions on Information Theory 52 (6) (2006) 2551–2567.
[30] D.J.C. MacKay, Fountain codes, IEE Communications 152 (6) (2005) 1062–1068.
[31] G. Marfia, M. Roccetti, A. Cattaneo, A. Sentinelli, A. Vitali, L. Celetto, M. Gerla, Digital fountains + P2P for future IPTV platforms: a test-bed evaluation, in: IFIP International Conference on New Technologies, Mobility, and Security (NTMS), February 2011.
[32] ETSI, Digital Video Broadcasting (DVB); Guidelines for DVB-IP Phase 1 Handbook, Tech. Rep. ETSITR102542V1.3.1, November 2007.
[33] T. Mladenov, S. Nooshabadi, K. Kim, MBMS raptor codes design trade-offs for IPTV, IEEE Transactions on Consumer Electronics 56 (3) (2010) 1264–1269.
[34] 3GPP, Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs, 3GPP Tech. Specification TS 26.346 v6.6.0, October 2006.
[35] D. Astely, E. Dahlman, A. Furuskär, Y. Jading, M. Lindström, S. Parkvall, LTE: the evolution of mobile broadband, IEEE Communications Magazine 47 (4) (2009) 44–51.
[36] H.C. Joo, H. Song, D.B. Lee, I.K. Lee, An effective IPTV channel control algorithm considering channel zapping time and network utilization, IEEE Transactions on Broadcasting 54 (2) (2008) 208–216.
[37] D.B. Lee, H.C. Joo, H. Song, An effective channel control algorithm for integrated IPTV services over DOCSIS CATV networks, IEEE Transactions on Broadcasting 53 (4) (2007) 789–796.
[38] D.B. Lee, H. Song, QoE-aware mobile IPTV channel control algorithm over WiMAX network, Journal of Visual Communication and Image Representation 21 (2010) 245–255.
[39] R.L. Haupt, S.E. Haupt, Practical Genetic Algorithms, second ed., John Wiley & Sons, 2004.
[40] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer, Heidelberg, Germany, 2003.
[41] OPNET Modeler [Online]. Available: <www.opnet.com>.
[42] Joint Video Team (JVT) Reference Software, version 14.2. [Online]. Available: <http://iphome.hhi.de/suehring/tml/download/jm14.2.zip>.
[43] A. Shokrollahi, Raptor Codes, Tech. Rep. DR2003-06-001, Digital Fountain, June 2003.
[44] V. Erceg, L.J. Greenstein, S.Y. Tjandra, S.R. Parkoff, A. Gupta, B. Kulic, A.A. Julius, R. Bianchi, An empirically based path loss model for wireless channels in suburban environments, IEEE Journal on Selected Areas in Communications 17 (1999) 1205–1211.
[45] Recommendation ITU-R M.1225, Guidelines for Evaluation of Radio Transmission Technologies for IMT-2000, Annex 2, 1997.
[46] E. Horowitz, S. Sahni, S. Rajasekaran, Fundamentals of Computer Algorithms, second ed., Galgotia Publications, New Delhi, 2003.