Hybrid Multicast and Segment-Based Caching for VoD Services in LTE Networks

Kwangjin Choi, Seong Gon Choi, and Jun Kyun Choi

This paper proposes a novel video delivery scheme that reduces the bandwidth consumption cost from a video server to terminals in Long-Term Evolution networks. This proposed scheme combines optimized hybrid multicast with a segment-based caching strategy for use in environments where the maximum number of multicast channels is limited. The optimized hybrid multicast, allocation of multicast channels, and cache allocation are determined on the basis of a video's request rate, the related video's length, and the variable cost per unit size of a segment belonging to the related video. Performance evaluation results show that the proposed scheme reduces a video's delivery costs. This work is applicable to on-demand TV services that feature asynchronous video content requests.

Keywords: LTE, VoD, hybrid multicast, patching, caching.

I. Introduction

The volume of video traffic transferred through data networks has increased exponentially in recent years. This is because people began to consume large amounts of video content through over-the-top services of data networks, rather than from cable and terrestrial broadcast services. Therefore, since network capacity is limited, it has become necessary to reduce video traffic by considering video delivery strategies from the viewpoint of network design and service operation.

To date, there have been many studies aimed at reducing video traffic by better management of broadcasting, multicasting, and caching. The original concepts of *broadcast* and *multicast* were developed taking into account only real-time streams, while on-demand streams, in which the request time is asynchronous, were not a concern. Some researchers proposed combination techniques that used multiple multicast streams to accommodate video-on-demand (VoD) services [1]–[3]. In these cases, however, terminals have to wait to receive the beginning of the first segment of a video; consequently, this creates a problem of waiting time [1]–[3]. In addition, previous studies did not consider environments in which the maximum number of multicast channels is finite, as it is in Long-Term Evolution (LTE) networks.

In addition, traditional caching schemes have been studied under the assumption that only unicast transmissions are used [4]–[6]. These caching schemes considered only popularity distributions of videos and focused on reducing the bandwidth consumption cost generated by videos of high popularity. The bandwidth consumption cost per unit size of a segment was not taken into consideration. It is important to understand that this bandwidth consumption cost changes in accordance with the number of segments and transmission scheme.

Manuscript received Nov. 4, 2014; revised Mar. 12, 2015; accepted May 4, 2015.

This work was partly supported by the ICT R&D program of MSIP/IITP, Rep. of Korea (1391104001, Research on Communication Technology using Bio-inspired Algorithm) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (2015R1A2A2A03004152).

Kwangjin Choi (kwngjn@kaist.ac.kr) is with the Department of Information and Communications Engineering, KAIST, Daejeon, Rep. of Korea.

Seong Gon Choi (corresponding author, sgchoi@cbnu.ac.kr) is with the Department of Information and Communications Engineering, Chungbuk National University, Cheongju, Rep. of Korea.

Jun Kyun Choi (jkchoi59@kaist.edu) is with the Department of Electrical Engineering, KAIST, Daejeon, Rep. of Korea.

To overcome the limitations of previous studies, this paper proposes a novel video delivery scheme called hybrid multicast and segment-based caching (HMSC). The proposed scheme reduces the bandwidth consumption cost of VoD services, in comparison with existing schemes, in LTE networks in which the number of multicast channels is limited. HMSC minimizes unicast streams that cause large bandwidth consumption and reduces generation of unnecessary multicast channels. It achieves this by separating the present multicast stream from the next recurrent multicast stream. In addition, it maximizes the caching effect by storing those video segments that generate higher bandwidth consumption cost per unit size.

The performance of the proposed scheme is evaluated using simulations. It is compared with existing transmission schemes ([1]–[3] and [7]–[8]) that generate multicast channels repeatedly in succession and unconditionally, and with an existing caching scheme [5] that considers only the popularity of the video content. The performance results indicate that HMSC generates a minimal bandwidth consumption cost over the existing transmission schemes. Furthermore, the results also indicate that HMSC significantly reduces video delivery costs over the entire range of the storage capacity of a cache server.

II. Related Work

To deliver substantial video traffic with efficient consumption of available bandwidth, 3GPP has been developing evolved Multimedia Broadcast Multicast Services (eMBMS) [9]. However, the original multicast concept is appropriate only to deliver real-time broadcast content; it is not suitable for application to asynchronous services such as VoD.

There has been a lot of research into the use of broadcast and multicast to reduce the number of serving streams for asynchronous services. Researchers in early studies proposed batching schemes that aggregate arriving requests and provide video delivery using multicast channels [10]–[11]. The major drawback of these solutions is that the terminals that send the requests in advance have to wait for "multicast-channel initiation for batched requests" to arrive.

Other researchers tried to reduce the aforementioned waiting time by transmitting multiple multicast streams for each video. In one study [12], a threshold-based multicast scheme was proposed that staggered a starting time across multiple channels, and in another [2], researchers proposed a fast broadcast (FB) scheme that repeatedly broadcasts 2^{i-1} segments on the ith broadcast channel. In a previously cited study [1], a harmonic broadcast (HB) scheme was proposed, in which each video segment was delivered through a dedicated broadcast channel with harmonically reduced bandwidth.

The broadcast schemes of [1] and [2] did not solve the underlying problem; that is, the requested playing time does not coincide with the transferred time stream. Although the problem was solved in [12] by combining patching with multicasting, this required significantly large bandwidth consumption compared with other broadcast schemes. In our previous study [13], we proposed a hybrid broadcast transmission scheme that required small bandwidth consumption and solved the problem of waiting time. However, this approach assumed an infinite number of broadcast channels and did not consider caching schemes. In addition, it dissipated some bandwidth resource by transmitting broadcast channel streams regardless of whether requests arrived.

There have been many studies on caching schemes to reduce the cost of video delivery from a source server to terminals [4]–[6] and [14]–[15]. Sofman and Krogfoss [5] proposed hierarchical caching for IPTV services, and Vleeschauwer and Laevens [6] studied the effect of caching algorithms for on-demand IPTV services. However, since most studies assumed only unicast transmission, they were focused on caching popular videos that generate large delivery costs as opposed to caching segments. In a few studies [14]–[15], the segment-caching effect was analyzed; however, those researchers proposed caching only prefix segments of a video, not suffix segments. They did not take into account the possibility that the bandwidth consumption cost of a suffix segment of a video might be larger than that for a prefix segment of the same video.

III. System Model

1. Network Architecture

The on-demand video delivery service architecture considered in this paper consists of the public Internet, LTE networks, a source server, cache servers, and terminals that request videos asynchronously. The source server is placed on the public Internet and the terminals are connected to LTE networks. Cache servers are located near Evolved Node B (eNodeB) of LTE networks, as depicted in Fig. 1. The source server stores all video content, and the cache server stores partial segments of each video. If a segment of a video is stored on the cache server, then the segment is delivered from the cache server to terminals. The other segments of the requested video are delivered from the source server.

This paper considers scenarios in which a number of videos are delivered from the source server, or cache servers, to terminals through LTE networks with eMBMS. We assume that eMBMS exploits multicast-broadcast single-frequency network (MBSFN)-operation at base stations, eNodeB. The eNodeB, which is synchronized in an MBSFN area, transfers the same data with the same wireless resource allocated for

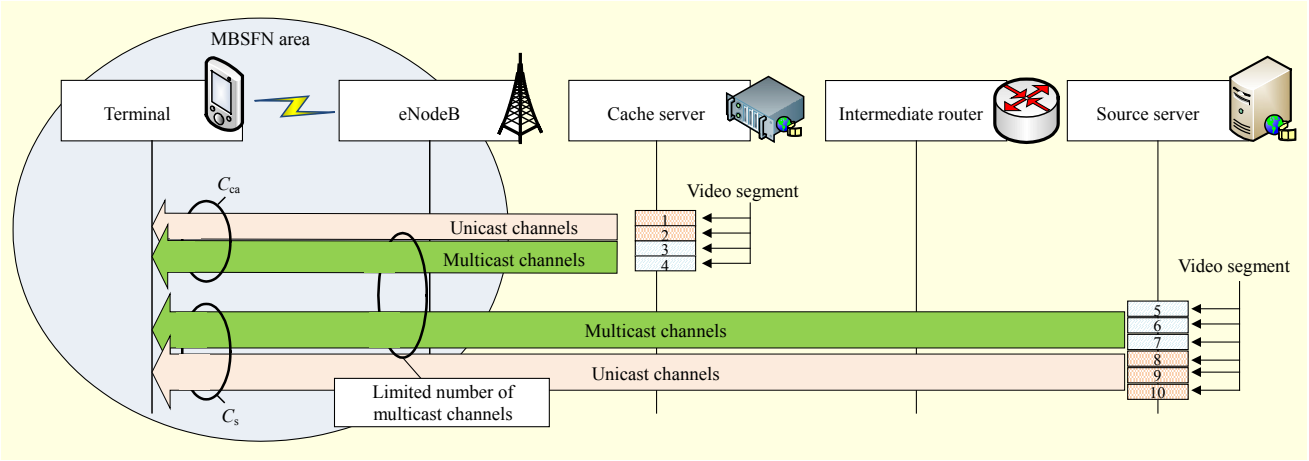

Fig. 1. Network architecture.

broadcast or multicast channel information. In an eNodeB cell, each multicast channel (MCH) is modulated with the same coding scheme in wireless areas, and the number of MCHs is finite [16]–[17].

LTE networks deliver video traffic through unicast or multicast channels. It is assumed that terminals receive multiple multicast channels and unicast channels concurrently. In addition, cache servers that are placed near eNodeB of LTE networks, store video segments to reduce the traffic volume from the source server to terminals.

2. Video Delivery Cost

In this paper, it is assumed that cache server installation cost is insignificantly small, compared with the transmission cost. Hence, the video delivery cost is defined as the average bandwidth consumption cost.

Let $N(v)$ denote the average number of serving channels for a video v per unit of time, and let $C(v)$ denote the average bandwidth consumption of the above serving channels for the video v. Here, $C(v)$ is equal to the average amount of streaming by a server per unit of time. As already mentioned in Section III-1, a cache server is placed between the source server and terminals and takes over the role of streaming to terminals. This means that the greater the number of channels the cache server serves for the requested video, the fewer channels the source server serves. Therefore, the cost related to the cache server, C_{ca}, and the cost related to the source server, C_s, are defined as follows:

$$C_{ca} = \sum_{\forall v_{i,j} \in G} C(v_{i,j}), \tag{1}$$

$$C_s = \sum_{i=1}^{V} \sum_{j=1}^{J_i} C(v_{i,j}) - \sum_{\forall v_{i,j} \in G} C(v_{i,j}), \tag{2}$$

where V is the total number of videos, J_i is the number of segments of the ith video, v_{ij} is the jth segment of the ith video, and G represents a set of segments stored on the cache server.

Table 1. Key parameters used in this paper.

Abbreviations	Explanation and meaning
β_1, β_2	Factors for normalization of the cost
λ_{all}	Sum of the request rates of all videos
λ_i	Request rate of the ith video
b	Streaming rate of a video
C_{all}, C_s, C_{ca}	Total video delivery cost; Video delivery cost by the source server; Video delivery cost caused by the cache server
$C_{(\cdot)}(v_i, m)$	Average bandwidth consumption generated by the video v_i, when allocating m multicast channels to the video v_i
$C_{(\cdot)}(v_{i,j,\delta}, m)$	Average bandwidth consumption generated by the ith video's jth segment of unit size δ when allocating m multicast channels to the video v_i
G	Set of segments stored at the cache server
J_i	Number of segments of the video v_i
L	Length of a video (seconds)
M	Maximum number of multicast channels
$N_{(\cdot)}(v)$	Average number of serving channels for a video v per unit of time
$R(m_i)$	Reduced bandwidth consumption when allocating m_i multicast channels for video i over allocating zero multicast channel for video i
S_{ca}	Size of the cache server storage
V	Total number of videos

In addition, the total video delivery cost, C_{all}, is defined as

$$C_{all} = \beta_1 C_s + \beta_2 C_{ca}, \tag{3}$$

where β_1 and β_2 are scalar factors used to normalize the costs of the source server and the cache server, respectively. It is

assumed that β_2 is very small compared with β_1 since the traffic from the source server is delivered through both core networks and access networks, but the traffic from the cache server is delivered only through access networks. Table 1 presents the key parameters that appear in this paper.

IV. Proposed Video Delivery Scheme

The proposed scheme, HMSC, transmits a video using three transmission schemes that combine multicast and unicast. In this paper, these three hybrid video transmission schemes (based on a combination method) are called unicast (UNI), hybrid fast multicast (HFM), and hybrid harmonic multicast (HHM). UNI disposes a video for a unicast channel; whereas HFM and HHM divide a video into several segments and dispose the segments for multiple dedicated multicast channels.

HMSC determines which transmission scheme is selected (UNI, HFM, or HHM) and how many dedicated multicast channels for a video are allocated. These decisions are made based on the maximum number of multicast channels in the LTE networks, the video length, and the request rate for the video, with the goal of minimizing the video delivery cost. Video segments may be delivered in three ways. In the first way, segments are delivered only through unicast channels. In the second, segments are delivered only through multicast channels, and in the third way, segments are delivered through both unicast and multicast channels. This means that the average bandwidth consumption generated by each video segment may vary, even though the segments belong to the same video. HMSC maximizes the caching effect by storing segments that generate large cost, after analyzing the transmission scheme and average bandwidth consumption.

1. Multicast and Unicast Combination Approach

This subsection describes how UNI, HFM, and HHM work under the proposed multicast and unicast combination approach. In addition, for each of the schemes, we formulate several equations for quantifying their performance. Assuming that the optimal transmission scheme for video v_i is to be determined by the request rate and length of the video, Fig. 2 describes the procedure for unicast and multicast channel initiation when a terminal requests video v_i.

Note that $N_{(\cdot)}(v_i, m)$ and $C_{(\cdot)}(v_i, m)$ denote the average number of serving channels per unit of time and the average bandwidth consumption per unit of time by the servers, respectively, when transmitting video v_i using m multicast channels. Here, $N_{(\cdot)}(v_i, m)$ and $C_{(\cdot)}(v_i, m)$ are dependent on the request rate, λ_i, and video length, L, of video v_i.

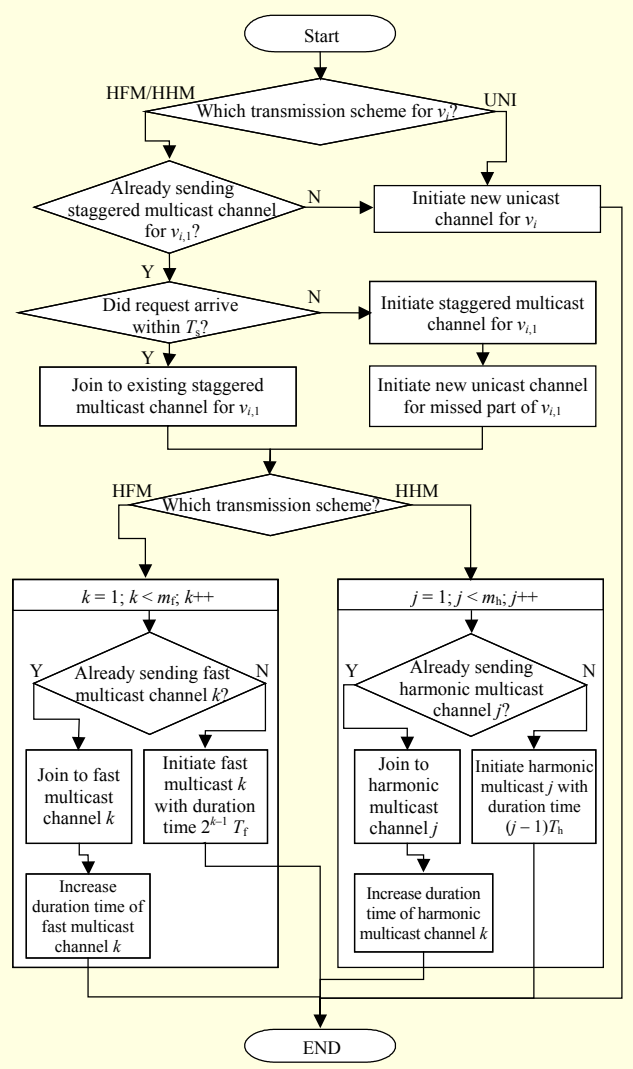

Fig. 2. Procedure of unicast and multicast channel initiation when terminal requests video v_i.

A. UNI

For UNI, all requests are served on unicast channels. Therefore, $N_u(v_i, m)$ and $C_u(v_i, m)$ are calculated as follows:

$$N_u(v_i, m) = N_u(v_i, 0) = \lambda_i L, \tag{4}$$

$$C_u(v_i, m) = C_u(v_i, 0) = b\lambda_i L, \tag{5}$$

where b is the streaming rate of video v_i.

Using (4) and (5), $C_u(v_{i,j,\delta}, m)$, the average bandwidth consumption generated by ith video's jth segment of unit size δ when allocating m multicast channels to video v_i, transmitted in UNI, is calculated as follows:

$$C_u(v_{i,j,\delta}, 0) = C_u(v_{i,j,\delta}) = \frac{C_u(v_{i,j})}{l_{i,j}} = \frac{b\lambda_i L}{bL} = \lambda_i, \tag{6}$$

ETRI Journal, Volume 37, Number 4, August 2015
http://dx.doi.org/10.4218/etrij.15.0114.1278

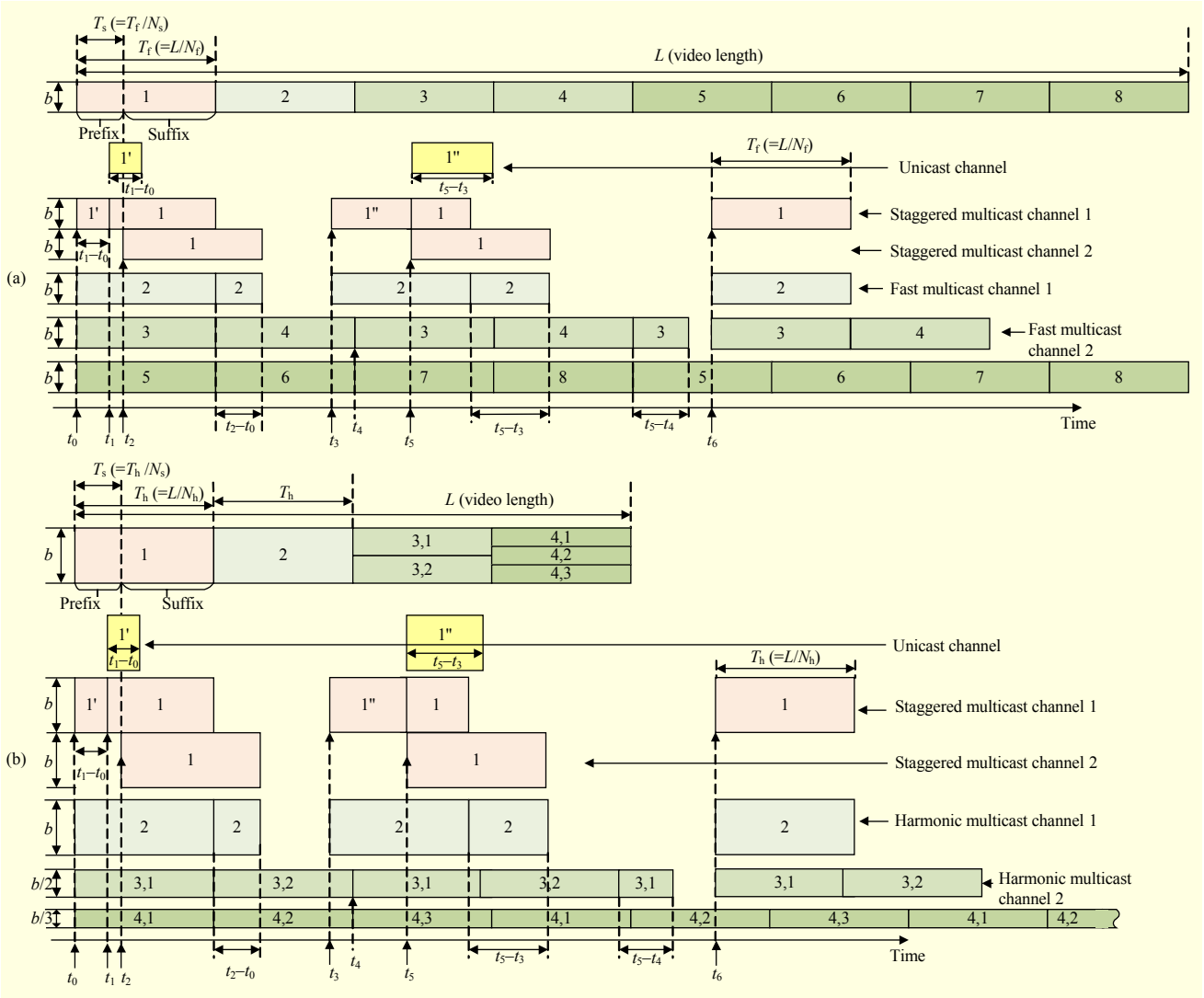

Fig. 3. Proposed transmission schemes: (a) HFM and (b) HHM.

where $l_{i,j}$ is the length of the jth segment of video v_i and $v_{i,j,\delta}$ is a sub-segment of unit size (of the jth segment of video v_i).

B. HFM

For HFM, a video is divided into 2^{m_f} segments of equal size $T_f (= L / 2^{m_f})$ and the video is repeatedly transmitted, exploiting the maximum $m_s + m_f$ multicast channels. In other words, the first segment is delivered through m_s staggered multicast channels with at least T_s (= T_f/m_s) seconds interval. The other $2^{m_f} - 1$ segments are delivered through m_f fast multicast channels. The fast multicast channels of HFM do not consistently occupy the bandwidth, compared with traditional FB schemes [2]–[3]. In HFM, each fast multicast channel is deallocated when any terminal does not request the segments contained in the stream of this fast multicast channel.

As shown in Fig. 3(a), if a request arrives at the server at t_0, then HFM initiates m_f+ 1 new multicast channels for the video since there is no multicast channel already serving the video. The first segment is delivered through "staggered multicast channel 1," and the other segments are delivered through the kth "fast multicast channel," which contains 2^{k-1} segments. The initial duration time of each multicast channel is the same as the size of the segments dedicated to the multicast channel; thus, HFM retains staggered multicast channel 1 during T_f seconds and retains fast multicast channel k during $2^{k-1}T_f$ seconds. After some time, the second request arrives at t_1, within T_s seconds from the start time of staggered multicast channel 1. This delivers the first segment. Then, the terminal that sent the second request joins the existing staggered multicast channel 1, and fast multicast channels 1, 2, and 3. The missed part of the first segment is delivered through a unicast channel.

If the second request arrives T_s seconds after the latest

staggered multicast stream started (for example, at t_2), then the request initiates a new staggered multicast channel for delivering the first segment. Therefore, the duration of the staggered multicast channel does not change. Since the first fast multicast channel is ongoing when the second request arrives, the request joins the first fast multicast channel and $(t_2 - t_0)$ seconds is added to the duration time of the first fast multicast channel to deliver the complete second segment to the terminal that requests it. If the first fast multicast channel is already terminated when the next request arrives (for example, at t_6), then the request reinitiates the first fast multicast channel to receive the second segment. Likewise, the time gap between the start time of the kth fast multicast channel and the arrival time of the latest request may be added to the duration time of the kth fast multicast channel. Alternatively, the kth fast multicast channel may be initiated by the latest request. HFM does not cause a waiting-time problem in the middle of video playtime because the terminal receives the segments of the requested video ahead of time, or in time.

The average number of ongoing unicast channels and staggered multicast channels is derived from [12], and the probability that fast multicast channel k is not ongoing is identical to the probability that any request has not arrived within $2^{k-1}\lambda_i T_f$ seconds. Therefore, the average number of ongoing channels in HFM is calculated as follows:

$$\begin{aligned} N_f(v_i, m) &= N_f(v_i, T_f, T_s, J_i) \\ &= \frac{\lambda_i^2 T_s^2 + 2\lambda_i T_f}{2(\lambda_i T_s + 1)} + \sum_{k=1}^{\log_2 J_i} \left(1 - e^{-2^{k-1}\lambda_i T_f}\right), \end{aligned} \tag{7}$$

where $J_i = 2^{m_f}$, $T_f = L/m_f$, $T_s = T_f/m_s$, and $m = m_s + m_f$.

Since all unicast, and multicast, channel streaming rates in HFM are the same, the average bandwidth-consumption cost per unit of time is calculated as follows:

$$\begin{aligned} C_f(v_i, m) &= bN_f(v_i, m) \\ &= b\left\{\frac{\lambda_i^2 T_s^2 + 2\lambda_i T_f}{2(\lambda_i T_s + 1)} + \sum_{k=1}^{\log_2 J_i} \left(1 - e^{-2^{k-1}\lambda_i T_f}\right)\right\}. \end{aligned} \tag{8}$$

In HFM, only the prefix of the first segment is delivered through both unicast channels and multicast channels, while the suffixes of segments are delivered through only multicast channels. Each segment on multicast channels is delivered using a different time period. For instance, the second segment is transferred at about period T_f, but the third segment is transferred at about period $2T_f$ when the total number of multicast channels is equal to three. The average bandwidth consumption generated by the jth segment of unit size transmitted in HFM is calculated using (8) as follows:

$$C_f(v_{i,j,\delta}, m) = \begin{cases} \dfrac{\lambda_i^2 T_s + 2\lambda_i}{2(\lambda_i T_s + 1)} & \text{for } j = 1, \text{ prefix} \\ \dfrac{\lambda_i}{\lambda_i T_s + 1} & \text{for } j = 1, \text{ suffix} \\ \dfrac{\left(1 - e^{-2^{k-1}\lambda_i T_f}\right)}{2^{k-1} T_f} & \\ \text{for } 2^{k-1} < j \le 2^k,\ 1 \le k \le \log_2 J. & \end{cases} \tag{9}$$

C. HHM

For HHM, a video is divided into m_h segments of equal size, T_h, and the video is repeatedly transmitted using the maximum number of multicast channels, $m_s + m_h$. Since a genuine harmonic multicast scheme does not deliver a video stream to a terminal in time, it requires any terminal that requested video v_i to wait for $(J_i - 1)L / J_i^2$ units of time [18]. Therefore, the first segment is delivered through a unicast channel, and m_s dedicated multicast channels, with at least T_s seconds interval, in effect, multicast to solve the waiting-time problem. The other jth segments are divided into $j - 1$ sub-segments, and the sub-segments are then delivered through harmonic multicast channel $j - 1$ with $b/(j - 1)$ streaming rate. The harmonic-multicast channels of HHM do not consistently occupy their bandwidth, unlike with traditional HB schemes [1], [7]. In HHM, harmonic multicast channels are deallocated when any terminal does not request segments that are contained in the streams of harmonic multicast channels.

As shown in Fig. 3(b), a request arrives at the server at t_0 and HHM initiates $m_h + 1$ new multicast channels for the video if there is no multicast channel serving the video. The basic concept of staggered multicasting, joining the existing multicast channels, and adding the duration time of each of the harmonic multicast channels is similar to HFM. However, each harmonic multicast channel delivers only one segment, and the basic duration of the harmonic multicast channels increases linearly from 1 to $(m_h - 1)$. The average number of ongoing channels in HHM is calculated as follows:

$$\begin{aligned} N_h(v_i, m) &= N_h(v_i, T_h, T_s, J_i) \\ &= \frac{\lambda_i^2 T_s^2 + 2\lambda_i T_h}{2(\lambda_i T_s + 1)} + \sum_{j=1}^{J_i - 1} \left(1 - e^{-j\lambda_i T_h}\right), \end{aligned} \tag{10}$$

where $J_i = m_h + 1$, $T_h = L/m_h$, $T_s = T_h/m_s$, and $m = m_s + m_h$.

Since the streaming rate for the jth segment is $b/(j - 1)$ as depicted in Fig. 3(b), the average bandwidth-consumption per unit of time is calculated as follows:

$$C_h(v_i, m) = b\frac{\lambda_i^2 T_s^2 + 2\lambda_i T_h}{2(\lambda_i T_s + 1)} + b\sum_{j=1}^{J_i - 1} \frac{1}{j}\left(1 - e^{-j\lambda_i T_h}\right). \tag{11}$$

http://dx.doi.org/10.4218/etrij.15.0114.1278

In HHM, only the prefix of the first segment is delivered through both unicast channels and multicast channels, while the suffixes of segments are delivered only through multicast channels. The average bandwidth consumption generated by the jth segment of unit size transmitted in HHM is calculated as follows:

$$C_{\mathrm{h}}(v_{i,j,\delta},m)=\begin{cases}\dfrac{\lambda_i^2T_{\mathrm{s}}+2\lambda_i}{2(\lambda_iT_{\mathrm{s}}+1)} & \text{for } j=1, \text{ prefix},\\ \dfrac{\lambda_i}{\lambda_iT_{\mathrm{s}}+1} & \text{for } j=1, \text{ suffix},\\ \dfrac{1-e^{-(j-1)\lambda_iT_{\mathrm{h}}}}{(j-1)T_{\mathrm{h}}} & \text{for } 2\le j\le J_i.\end{cases} \quad (12)$$

2. Transmission Scheme Selection

Figure 4 describes the relationship between the request rate and the average bandwidth-consumption cost of video v_i for the HFM and HHM transmission schemes. The average bandwidth consumption of HHM is smaller than that of HFM when λ_i is smaller than about 0.025, and the average bandwidth consumption of HFM is smaller than that of HHM when λ_i is larger than about 0.025. Therefore, it is necessary to determine the optimal transmission scheme among UNI, HFM, and HHM to minimize the total video delivery cost when the maximum number of multicast channels is finite.

Let $R(m_i)$ denote the bandwidth consumption cost reduced by allocating m_i multicast channels for video v_i, compared to not allocating any multicast channels (that is, $R(m_i) = C_{\mathrm{u}}(v_i, 0)$ for $m_i = 0$, $R(m_i) = C_{\mathrm{u}}(v_i, 0) - \min[C_{\mathrm{f}}(v_i, m_i), C_{\mathrm{h}}(v_i, m_i)]$ for $m_i > 0$). Therefore, the optimization problem is formulated as

$$\begin{aligned}&\text{maximize}\sum_{i=1}^{V}R(m_i)\\ &\text{s.t.}\quad\sum_{i=1}^{V}m_i\le M,\end{aligned} \quad (13)$$

where V represents the total number of videos and M represents the maximum number of multicast channels.

Note that the optimization problem belongs to the family of the 0-1 knapsack problem [19] and is solved by the following dynamic programming equation:

$$\begin{aligned}D_{0,j}&=0 \quad \text{for} \quad 0\le j\le M,\\ D_{i,j}&=\max\left(D_{i-1,j},D_{i-1,j-m_i}+R(m_i)\right)\\ &=\max\left(D_{i-1,j},D_{i-1,j-m_i}+C_{\mathrm{u}}(v_i,0)\right.\\ &\quad\left.-\min\left(C_{\mathrm{h}}(v_i,m_i),C_{\mathrm{f}}(v_i,m_i)\right)\right)\\ &\quad\text{for}\quad 0<i\le V,0\le j\le M,\end{aligned} \quad (14)$$

where D represents a two-dimensional matrix, and entry D_{ij} denotes the maximum reduction in bandwidth consumption for the first i videos with j multicast channels. This dynamic programming computes the entries from $D_{0,0}$ to $D_{V,M}$, in row order. When computing an entry D_{ij}, we store the selected transmission scheme (from among UNI, HFM, and HHM) into matrix T_{ij} and store the number of allocated multicast channels for v_i into matrix A_{ij}. Therefore, when the total number of videos is i and the maximum number of multicast channels is j, T_{ij} represents the optimal video transmission scheme for the ith video and $A_{i,j}$ represents the optimal number of multicast channels to be allocated to the ith video.

The optimal transmission scheme and number of dedicated multicast channels for each video are determined by tracking A_{ij} from $A_{V,M}$ to $A_{0,0}$. Figure 5 shows an example of the tracking operation of the algorithm involving four videos and four multicast channels. In this figure, $T_{4,4}$ indicates that the optimal transmission scheme for video v_4 is UNI, and $A_{4,4} = 0$ indicates that the optimal number of dedicated multicast channels is "0" for video v_4. Because A_{ij} is the difference value between column index values of entry $D_{i-1,j}$ and entry $D_{i-1,j-m_i}$ for computing D_{ij}, $T_{4-1,4-A_{4,4}}$ = HFM represents the optimal

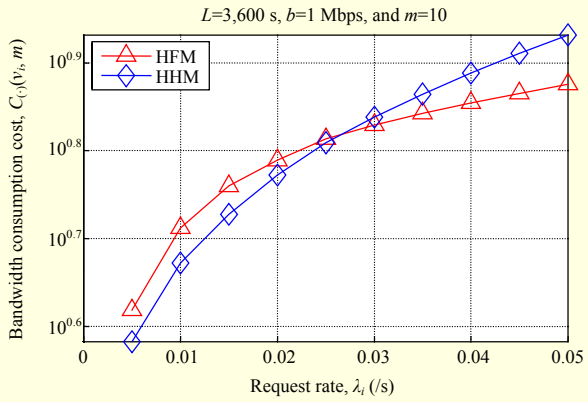

Fig. 4. Video delivery cost of transmission schemes for various video request rates.

	$j=0$	$j=1$	$j=2$	$j=3$	$j=4$
$i=0$	$D_{0,0}$=0 $T_{0,0}$= $A_{0,0}$=0	$D_{0,1}$=0 $T_{0,1}$= $A_{0,1}$=0	$D_{0,2}$=0 $T_{0,2}$= $A_{0,2}$=0	$D_{0,3}$=0 $T_{0,3}$= $A_{0,3}$=0	$D_{0,4}$=0 $T_{0,4}$= $A_{0,4}$=0
$i=1$	$D_{1,0}$=0 $T_{1,0}$=UNI $A_{1,0}$=0	$D_{1,1}$=81.4 $T_{1,1}$=HFB $A_{1,1}$=1	$D_{1,2}$=121.4 $T_{1,2}$=HFM $A_{1,2}$=2	$D_{1,3}$=140.9 $T_{1,3}$=HFB $A_{1,3}$=3	$D_{1,4}$=150.2 $T_{1,4}$=HFB $A_{1,4}$=4
$i=2$	$D_{2,0}$=0 $T_{2,0}$=UNI $A_{2,0}$=0	$D_{2,1}$=81.4 $T_{2,1}$=UNI $A_{2,1}$=0	$D_{2,2}$=130.4 $T_{2,2}$=HFB $A_{2,2}$=1	$D_{2,3}$=170.4 $T_{2,3}$=HFM $A_{2,3}$=1	$D_{2,4}$=194.1 $T_{2,4}$=HFB $A_{2,4}$=2
$i=3$	$D_{3,0}$=0 $T_{3,0}$=UNI $A_{3,0}$=0	$D_{3,1}$=81.4 $T_{3,1}$=UNI $A_{3,1}$=0	$D_{3,2}$=130.4 $T_{3,2}$=UNI $A_{3,2}$=0	$D_{3,3}$=170.4 $T_{3,3}$=UNI $A_{3,3}$=0	$D_{3,4}$=206.7 $T_{3,4}$=HFM $A_{3,4}$=1
$i=4$	$D_{4,0}$=0 $T_{4,0}$=UNI $A_{4,0}$=0	$D_{4,1}$=81.4 $T_{4,1}$=UNI $A_{4,1}$=0	$D_{4,2}$=130.4 $T_{4,2}$=UNI $A_{4,2}$=0	$D_{4,3}$=170.4 $T_{4,3}$=UNI $A_{4,3}$=0	$D_{4,4}$=206.7 $T_{4,4}$=UNI $A_{4,4}$=0

Fig. 5. Example of results of dynamic programming.

transmission scheme for video v_3; that is, HFM. In addition, the optimal number of dedicated multicast channels is "1" for video v_3. Continuing in this way, we found sets of optimal transmission schemes and the optimal number of dedicated multicast channels for each video to be (v_1, HFM, 2), (v_2, HFM, 1), (v_3, HFM, 1), and (v_4, UNI, 0).

3. Cache Allocation

HMSC selects segments to be stored on the cache server, after determination of the optimal transmission scheme and the number of multicast channels for all videos. Equations (1), (2), and (3) lead to the following cache allocation optimization problem to minimize the total video delivery cost, C_{all}:

$$\begin{aligned} &\text{maximize } C_{ca} = \sum_{\forall v_{i,j} \in G} C(v_{i,j}) \\ &\text{s.t. } \sum_{\forall v_{i,j} \in G} S(v_{i,j}) \le S_{ca}, \end{aligned} \tag{15}$$

where G denotes a set of segments stored on the cache server, $S(v_{i,j})$ denotes the size of the jth segment of the ith video, and S_{ca} is the size of the cache server storage. A segment for the last element of G may be divided into smaller segments, and one of the divided segments cached to fill the cache server. Therefore, this formulation is a type of fractional knapsack problem [19], and is solved by the greedy algorithm described in Fig. 6. This greedy algorithm selects the segments to store on the cache server, in order of decreasing average bandwidth consumption, generated by the jth segment of the ith video of unit size, $C(v_{i,j,\delta}, m_i)$. Here, $C(v_{i,j,\delta}, m_i)$ is calculated as in (6), (9), or (12), based on the optimal transmission scheme. The optimal transmission scheme for the ith video, v_i, and the number of allocated multicast channels, m_i, are determined by the transmission scheme selection described in the previous subsection.

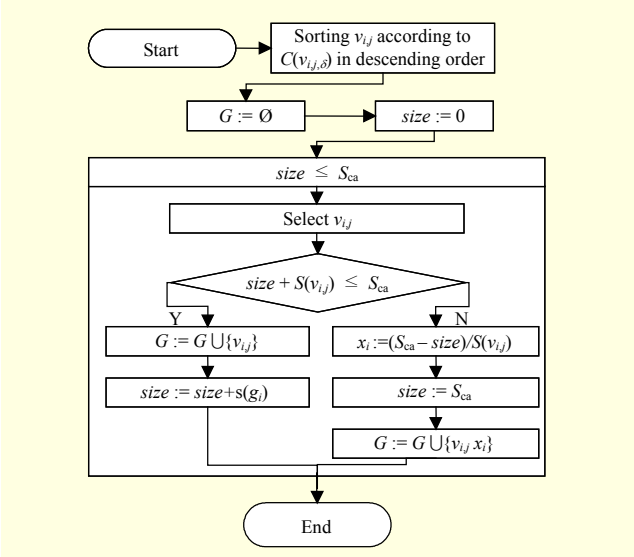

Fig. 6. Greedy algorithm for cache allocation.

V. Performance Evaluation

For the performance evaluation, the cases that serve unicast or one broadcast scheme from among FB [2]–[3], [8] and HB [1], [7] as the transmission scheme, and non-caching (Non) and hot-content-first caching (HOT) [4] as the caching scheme are considered for comparison with the proposed scheme, HMSC. FB transmits a video repeatedly through 2^{m_f} FB channels, and HB transmits a video repeatedly through m_h HB channels. The multicast channel allocation in FB and HB exploits the dynamic programming proposed in this paper, assuming a limited number of multicast channels. The HOT scheme stores videos in decreasing order of popularity as described previously [5].

The request rates of videos vary with their popularity. The request of ith video v_i is generated by a Poisson process, and the request rate of the ith video v_i is represented as λ_i. The request rate of a video may change on an hourly basis in a real environment. However, this paper simplified the video request model based on the Poisson process as the referred previous research [12] to focus on formulations that find the optimal video delivery scheme. The total request rate, λ_{all}, is derived by summing up the request rates of all videos. When all videos are sorted in descending order of popularity so that λ_i is larger than λ_{i+1}, the distribution of selection of the ith video v_i follows the Zipf distribution with power parameter α. The ith video's request rate λ_i is $\lambda H/i^{\alpha}$, where $H = 1/\sum_{i=1}^{V} 1/i^{\alpha}$ describes the normalization constraint.

This paper also assumes that the streaming rate b of all videos is 1 Mbps (a 360p standard quality video's bitrate recommended by YouTube [20]) and the power parameter α of the Zipf distribution is 0.729 [11]. The total request rate λ_{all} is

Table 2. Parameter values.

Parameter	Explanation and meaning	Value
β_1	Scalar factors for normalization of the source	1
β_2	Server's cost and the cache server's cost	0.001
λ_{all}	Sum of the request rates of all videos	0.1–1
b	Streaming rate of a video	1 Mbps
α	Power parameter of the Zipf distribution	0.729
L	Length of a video (s)	3,600 s
M	Maximum number of multicast channels	20–320
V	Total number of videos	20–1,280

ETRI Journal, Volume 37, Number 4, August 2015
http://dx.doi.org/10.4218/etrij.15.0114.1278

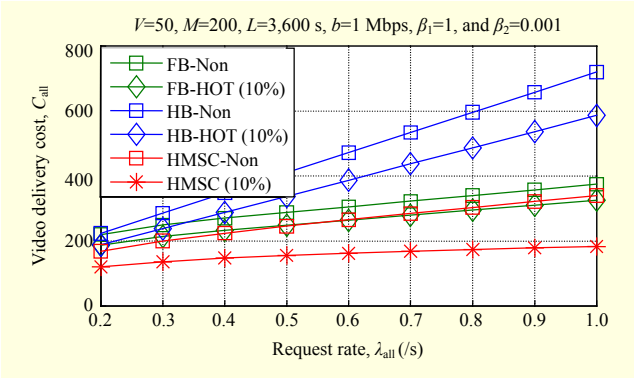

Fig. 7. Performance comparison of HMSC and other schemes as request rate varies.

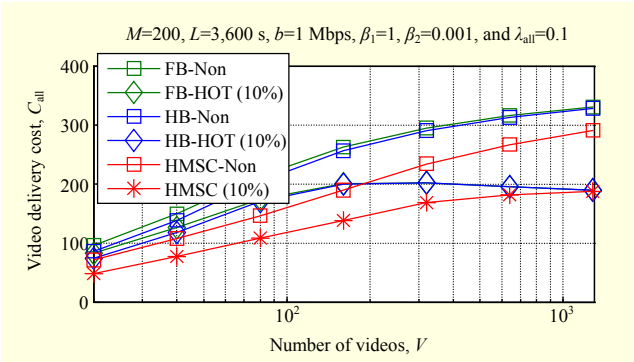

Fig. 8. Performance comparison of HMSC and other schemes as total number of videos varies.

static, and the popularity of videos does not change. In addition, the scalar factors, β_1 and β_2, of video delivery cost in (2) are set at "1" and "1/1,000," respectively. The cache server's storage size is represented as a percentage of the total size of all the videos. In addition, LTE terminals stay in the same MBSFN area while receiving MCH, and the wireless channel status of each terminal is stable, to receive video data. Therefore, it is assumed that LTE terminals are always able to receive video data through unicast and multicast channels at a static data rate configured by eNodeB. Table 2 presents the parameter values used in this section.

Figure 7 shows each caching scheme for different request rates and the video delivery cost of each transmission. It is assumed that the total number of videos, V, is 50, the video length, L, is 3,600 s (running time of most TV dramas), the maximum number of multicast channels, M, is 200, and the cache server storage size is 10% of the size of all the videos. HMSC generates the lowest video delivery cost across the entire range of request rates when each scheme does not use a cache server, and all the traffic is delivered from the source server. When the request rate λ_{all} is 0.6, HMSC-Non reduces the video delivery cost by 14% and 44% for FB-Non and HB-Non, respectively. It was observed that cost reduction increases when the cache server storage is 0.1. The video delivery cost for HMSC (10%) was 39% and 59% less than the corresponding costs for FB-HOT (10%) and HB-HOT (10%), respectively. When the request rate is larger than "3," the costs of FB-Non, HB-Non, FB-HOT, and HB-HOT increase rapidly, although this is not presented here due to the poor resolution of the figure. These results indicate that the number of requests served through a unicast channel in FB and HB increases due to the shortage of multicast channels. Since HMSC exploits multicast channels more efficiently than FB and HB, it accommodates more requests to the multicast channels.

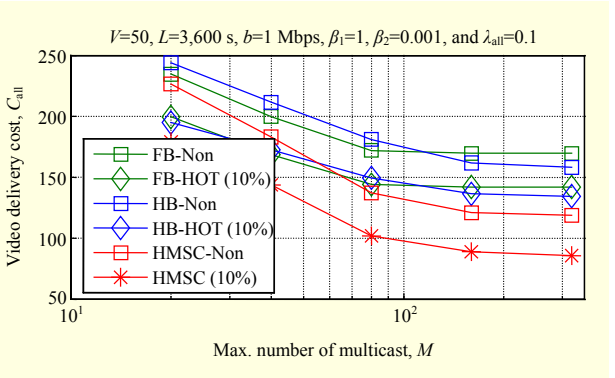

Fig. 9. Performance comparison of HMSC and other schemes as maximum number of multicast channels varies.

Figure 8 shows the video delivery cost as a function of the number of videos, V. The costs for FB; HB under non-caching and HOT caching; and HMSC are plotted on the graph. Note that the video delivery cost of HMSC (10%) is lower than the costs of FB-HOT (10%) and HB-HOT (10%) across the entire range of the number of videos. In addition, the video delivery costs of HMSC-Non, FB-Non, and HB-Non; and the video delivery costs of HMSC, FB-HOT, and HB-HOT converge to a value as the number of videos increases. This is because the effect of allocating multicast channels decreases when the number of videos increases. It is interesting to note from Fig. 8 that the video delivery cost of all schemes tends to decrease when using a caching scheme, and when λ_{all} is larger than a given value. These results suggest that the request rates of the highly popular videos account for a greater portion of the total request rates and that the effect of caching increases as the number of videos increases.

Figure 9 depicts variations in video delivery costs as the maximum number of multicast channels increases in the case where V = 50 and λ_{all} = 0.1. HMSC achieves the best results with respect to that of the second optimal scheme. When the number of multicast channels is larger than about 80, HMSC-Non reduces the video delivery cost, compared with other caching cases (FB-HOT and HB-HOT). Although the video

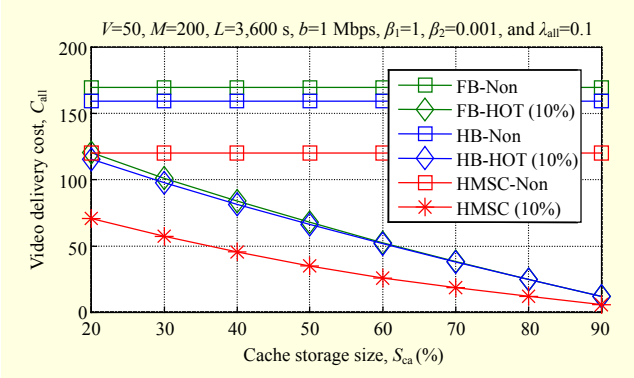

Fig. 10. Performance comparison of HMSC and other schemes as storage size of cache server varies.

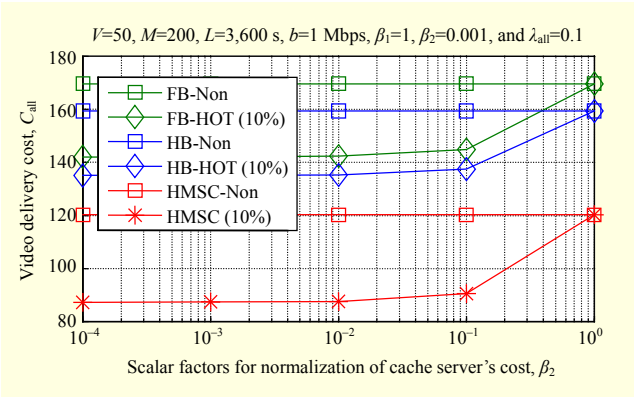

Fig. 11. Performance comparison of HMSC and other schemes as scalar factor for normalization of cache server's cost varies.

delivery cost of each scheme tends to decrease as the maximum number of multicast channels increases, there is also a reduction in the size of the cost-decrease of each scheme. The reason for this result is that the effect of additional allocation of multicast channels decreases as the video delivery cost value approaches the minimal cost of each scheme.

Figure 10 illustrates the results of variation of the video delivery cost for cache servers of different storage size in the case where $V = 50$, $M = 200$, and $\lambda_{all} = 0.1$. This result shows that HMSC is the optimal scheme for any cache server size. HMSC maximizes the effect of caching, and in particular, reduced it to 41% of that needed for HMSC-Non when the storage size of the cache server was 0.2. In addition, FB-HOT reduced the cache server size to 29% of that needed for FB-Non, and HB-HOT reduced it to 28% of that needed for HB-Non.

Figure 11 shows the video delivery cost as a function of the scalar factor β_2 for normalization of the cache server's cost. Note that the video delivery costs of HMSC (10%), FB-HOT (10%), and HB-HOT (10%) sharply approach to the corresponding costs for HMSC-Non, FB-Non, and HB-Non, respectively, when β_2 approaches to 1. This is because approaching to 1 means that the location of the cache server becomes near the source server.

VI. Conclusion

Video transmission schemes and caching schemes have become important considerations due to the rapid increase of VoD services through mobile networks. This paper proposed a novel video delivery scheme to reduce video delivery cost related to bandwidth consumption. In the proposed scheme (HMSC), the most suitable transmission scheme and required number of multicast channels for a video are determined by the formulations developed herein and dynamic programming based on request rate, video length, and maximum number of multicast channels. This new scheme used UNI, HFM, or HHM to effectively combine the unicast and multiple-dedicated multicasts to achieve two objectives: minimizing bandwidth consumption and solving the problem of waiting time. In addition, we determined that the cost generated by a segment of a video is subject to change as the transmission scheme changes. HMSC maximizes the effect of caching using a greedy algorithm by which ordering decreased the average bandwidth consumption of segments of unit size. The results from the performance evaluation show that HMSC significantly reduced the video delivery cost, compared with existing schemes. The proposed scheme could be used to design on-demand video streaming in LTE networks.

References

[1] L.-S. Juhn and L.-M. Tseng, "Harmonic Broadcasting for Video-on-Demand Service," *IEEE Trans. Broadcast.*, vol. 43, no. 3, Sept. 1997, pp. 268–271.

[2] L.-S. Juhn and L.-M. Tseng, "Fast Data Broadcasting and Receiving Scheme for Popular Video Service," *IEEE Trans. Broadcast.*, vol. 44, no. 1, Mar. 1998, pp. 100–105.

[3] S.A. Azad and M. Murshed, "An Efficient Transmission Scheme for Minimizing User Waiting Time in Video-on-Demand Systems," *IEEE Commun. Lett.*, vol. 11, no. 3, Mar. 2007, pp. 285–287.

[4] L. Shen, W. Tu, and E. Steinbach, "A Flexible Starting Point Based Partial Caching Algorithm for Video on Demand," *IEEE Int. Conf. Multimedia Expo*, Beijing, China, July 2–5, 2007, pp. 76–79.

[5] L.B. Sofman and B. Krogfoss, "Analytical Model for Hierarchical Cache Optimization in IPTV Network," *IEEE Trans. Broadcast.*, vol. 55, no. 1, Mar. 2009, pp. 62–70.

[6] D. De Vleeschauwer and K. Laevens, "Performance of Caching Algorithms for IPTV On-Demand Services," *IEEE Trans.*

Broadcast., vol. 55, no. 2, June 2009, pp. 491–501.
[7] H.-I. Kim and S.-K. Park, "A Hybrid Video-on-Demand Data Broadcasting and Receiving Scheme of Harmonic and Staggered Schemes," *IEEE Trans. Broadcast*., vol. 54, no. 4, Dec. 2008, pp. 771–778.
[8] H.-F. Yu, H.-C. Yang, and L.-M. Tseng, "Reverse Fast Broadcasting (RFB) for Video-on-Demand Applications," *IEEE Trans. Broadcast*., vol. 53, no. 1, Mar. 2007, pp. 103–111.
[9] 3GPP TS 25.346, Introduction of the Multimedia Broadcast/ Multicast Service (MBMS) in the Radio Access Network (RAN), Sept. 2012.
[10] A. Dan, D. Sitaram, and P. Shahabuddin, "Scheduling Policies for an On-Demand Video Server with Batching," *Proc. ACM Int. Conf. Multimedia*, New York, USA, Oct. 1994, pp. 15–23.
[11] A. Dan, D. Sitaram, and P. Shahabuddin, "Dynamic Batching Policies for an On-Demand Video Server," *ACM Multimedia Syst*., vol. 4, June 1996, pp. 112–121.
[12] L. Gao and D. Towsley, "Threshold-Based Multicast for Continuous Media Delivery," *IEEE Trans. Multimedia*, vol. 3, no. 4, Dec. 2001, pp. 405–414.
[13] K. Choi, S.G. Choi, and J.K. Choi, "Hybrid Video Transmission Scheme for Minimizing Server Bandwidth Consumption with Zero Start-up Delay in Video-on-Demand Service," *IEEE Commun. Lett*., vol. 16, no. 1, Jan. 2012, pp. 6–8.
[14] B. Wang et al., "Optimal Proxy Cache Allocation for Efficient Streaming Media Distribution," *IEEE Trans. Multimedia*, vol. 6, no. 2, Apr. 2004, pp. 366–374.
[15] C. Jayasundara et al., "Improving Scalability of VoD Systems by Optimal Exploitation of Storage and Multicast," *IEEE Trans. Circuits Syst. Video Technol*., vol. 24, no. 3, Mar. 2014, pp. 489–503.
[16] 3GPP TS 36.300, Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall Description; Stage 2, Mar. 2014.
[17] 3GPP TS 36.443, Evolved Universal Terrestrial Radio Access Network (E-UTRAN); M2 Appl. Protocol (M2AP), Sept. 2014.
[18] J.-F. Paris, S.W. Carter, and D.D.E. Long, "Efficient Broadcasting Protocols for Video on Demand," *Proc. Int. Symp. Modeling, Anal. Simulation Comput. Telecommun. Syst*., Montreal, Canada, July 19–24, 1998, pp. 127–132.
[19] H. Kellerer, U. Pferschy, and D. Pisinger, "*Knapsack Problems*," Berlin, Germany: Springer, 2004, pp. 20–27.
[20] *Recommended Upload Encoding Settings* (*Advanced*), Google, 2014. Accessed Nov. 4, 2014. https://support.google.com/youtube/answer/1722171?hl=en

Kwangjin Choi received his BE degree in electrical and communications engineering from the Korea Advanced Institute of Science Technology (KAIST), Daejeon, Rep. of Korea, in 2005 and his MS and PhD degrees in information and communications engineering from KAIST, in 2007 and 2015, respectively.

Seong Gon Choi received his PhD in information and communications engineering from the Information and Communications University, Daejeon, Rep. of Korea, in 2004. Since 2004, he joined Chungbuk National University, Cheongju, Rep. of Korea, as a professor. Since 2002, he has been working as an editor of ITU-T SG13. His main research interests include broadcast networks; mobility issues; next-generation network and infrastructure deployment; and network management.

Jun Kyun Choi received his BS degree in electronics engineering from Seoul National University, Rep. of Korea, in 1982 and his MS and PhD degrees in electronics engineering from the Korea Advanced Institute of Science Technology (KAIST), Daejeon, Rep. of Korea, in 1985 and 1988, respectively. From June 1986 until December 1997, he was with the Electronics and Telecommunication Research Institute, Daejeon, Rep. of Korea. In January 1998, he joined the Information and Communications University, Daejeon, Rep. of Korea, as a professor. In 2009, he moved to KAIST to work as a professor. He is a senior member of IEEE; an executive member of the Institute of Electronics Engineers of Korea; an editor board member of the Korea Information Processing Society, and life member of the Korea Institute of Communication Science.