Popularity–Based Adaptive Content Delivery Scheme with In-Network Caching

Jeong Yun Kim, Gyu Myoung Lee, and Jun Kyun Choi

To solve the increasing popularity of video streaming services over the Internet, recent research activities have addressed the locality of content delivery from a network edge by introducing a storage module into a router. To employ in-network caching and persistent request routing, this paper introduces a hybrid content delivery network (CDN) system combining novel content routers in an underlay together with a traditional CDN server in an overlay. This system first selects the most suitable delivery scheme (that is, multicast or broadcast) for the content in question and then allocates an appropriate number of channels based on a consideration of the content's popularity. The proposed scheme aims to minimize traffic volume and achieve optimal delivery cost, since the most popular content is delivered through broadcast channels and the least popular through multicast channels. The performance of the adaptive scheme is clearly evaluated and compared against both the multicast and broadcast schemes in terms of the optimal in-network caching size and number of unicast channels in a content router to observe the significant impact of our proposed scheme.

Keywords: Content delivery network, in-network caching, request routing, content popularity.

I. Introduction

In a content delivery network (CDN), a CDN server is traditionally used to reduce traffic on the Internet backbone by offloading traffic requests from the origin server. However, sitting outside networks provided by Internet service providers (ISPs), a CDN server cannot reduce traffic on the transit or peering links that connect the ISP network with the Internet backbone and other ISP networks [1]. As demand for content access and delivery over the Internet increases, innovative CDN architectures and technologies are becoming increasingly important to efficiently cache and distribute the surging amount of video content.

To minimize delivery latency and inter-ISP traffic, a lot of recent researches address localized delivery of large content volumes from a network edge by introducing a storage module into network entities (for example, a content router) [2]–[3]. In other words, a content router can be allowed to provide in-network caching and localized delivery while continuing to support its basic features such as packet forwarding and routing. Therefore, from the viewpoint of the design of a content router, the optimal in-network caching size should be carefully determined to minimize the performance degradation that results from the introduction of such a storage module.

In general, content delivery schemes can be classified into three major types. First, a unicast scheme does not appropriate well at a large scale and is, therefore, not discussed further in this paper. Second, a multicast scheme allows a number of requests for the same content to be grouped together and served by a single multicast stream. In a batching-based multicast scheme [4] for example, several content requests are delayed for a period of time before finally serving the resulting batch via a multicast stream. In a patching-based multicast

Manuscript received Sept. 23, 2013; revised Feb. 17, 2014; accepted June 27, 2014.

This research was supported by the ICT Standardization program of MISP (The Ministry of Science, ICT & Future Planning).

Jeong Yun Kim (corresponding author, jykim@etri.re.kr) is with the Communications & Internet Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Gyu Myoung Lee (G.M.Lee@ljmu.ac.uk) is with the Liverpool John Moores University, Liverpool, UK.

Jun Kyun Choi (jkchoi59@kaist.edu) is with the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea.

scheme [5]–[6], a content request is first served by a unicast stream and then joined back to a multicast stream. Third, a broadcast scheme [7] can broadcast content on dedicated channels at a pre-defined schedule.

Owing to the limitations of content caching and content delivery capabilities, content routers seem very unlikely to cache all content. However, it would be better to cache and deliver a prefix (that is, the beginning portion of the content), if its length is sufficiently short. In addition, prefix caching has a number of advantages, such as a reduction of both delivery latency to clients and traffic volume over networks [5], [8]–[9], particularly compared to the threshold-based multicast scheme running on a centralized server [6], [10]. Therefore, the CDN server can only deliver the suffix — that is the remaining portion other than the prefix — to multiple clients through a single multicast stream.

Our previous work in [11] showed that the performance of a patching-based multicast scheme is much better than that of batching-based multicast schemes. However, the former requires that content routers perform relatively complex processing operations. This is caused by the occurrence of changes in suffix lengths, which is due to the variation in the arrival times of suffix requests. Thus, compared to the latter scheme, which has a fixed suffix length, patching-based multicast schemes need more complex operations. Based on this context, this paper mainly focuses on patching-based multicast and broadcast schemes.

Proxy-assisted multicast schemes [5], which combine proxy prefix caching with multicast schemes, such as batching and patching, are generally known as their system control is simpler than that of broadcast schemes. Such schemes can collect more requests for the same content because they are served by a single multicast stream. On the other hand, proxy-assisted broadcast schemes [7] can significantly reduce the network resource requirements as well as service latency by broadcasting content to dedicated multicast channels. However, most research has focused on developing multicast schemes for generally minimizing the aggregate network bandwidth rather than the network bandwidth consumed by only proxy servers. The request-routing system (RRS) used in a traditional CDN system is used to redirect client requests to the closest surrogate by considering network proximity to provide fast delivery [2]–[3], [12]–[13]. This paper first presents detailed operations of a persistent RRS that can redirect all client requests for the same content to a particular content router once the router is chosen from the first request. Therefore, such requests can consume only a single multicast stream during their prefix lengths, thereby reducing the amount of network resources used. In addition, the persistent RRS can provide a finer granularity (for example, content chunk level) than that of the original RRS (for example, content file level).

With the persistent RRS and in-network caching, this paper introduces a hybrid CDN system that combines novel content routers in the underlay with the CDN server in the overlay. In addition to this, the hybrid CDN system is capable of providing adaptive content delivery. As an efficient delivery scheme is adaptively selected according to content popularity for the overall performance gain, the proposed popularity-based content delivery scheme can significantly reduce delivery latency and traffic volume over the network. Given the number of multicast channels in the CDN server, we address the problem of both minimizing the average number of channels (the required capacity) at the content routers and determining the optimal prefix length (that is, in-network caching size). We also evaluate and compare the performance of the proposed popularity-based adaptive scheme with other content delivery schemes to highlight the fact that the proposed one clearly has performance improvement against both the multicast and broadcast schemes coupled with in-network caching.

The remainder of this paper is organized as follows. The CDN system model is briefly presented in Section II. Section III describes a popularity-based adaptive content delivery technique with in-network caching. In Section IV, we evaluate the performance of content delivery schemes under varying conditions. The paper is concluded in Section V.

II. System Model

We illustrate the hybrid CDN system, which consists of an origin server, a CDN server, a persistent RRS, and content routers, in Fig. 1. A group of clients receiving content delivered across networks from the CDN server through the content routers are considered. The Hypertext Transfer Protocol (HTTP) is used for describing the requested content by its uniform resource identifier (URI) [14]. In general, the origin server is managed by the content provider and located in the data center. It also stores content that is distributed to both the CDN server and content routers before such a request is made.

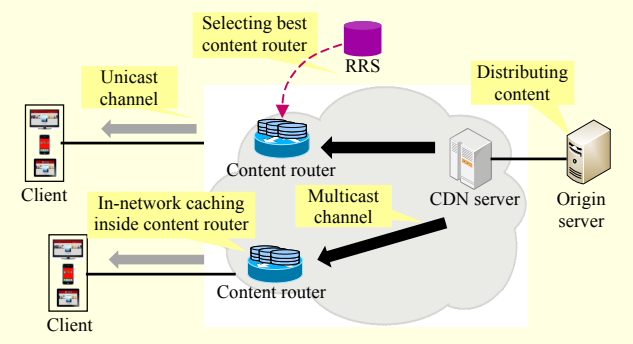

Fig. 1. Hybrid CDN system architecture.

ETRI Journal, Volume 36, Number 5, October 2014
http://dx.doi.org/10.4218/etrij.14.0113.0090

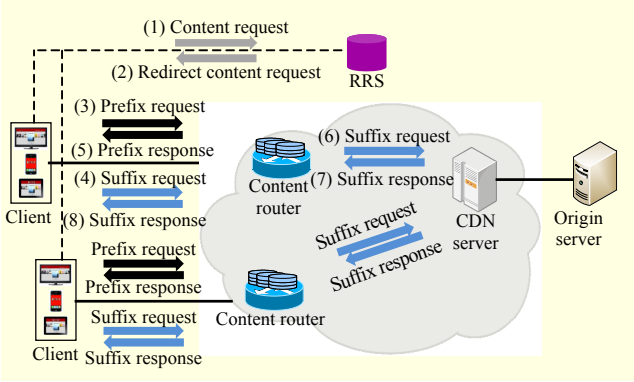

Fig. 2. Multicast delivery scheme with in-network caching and operation.

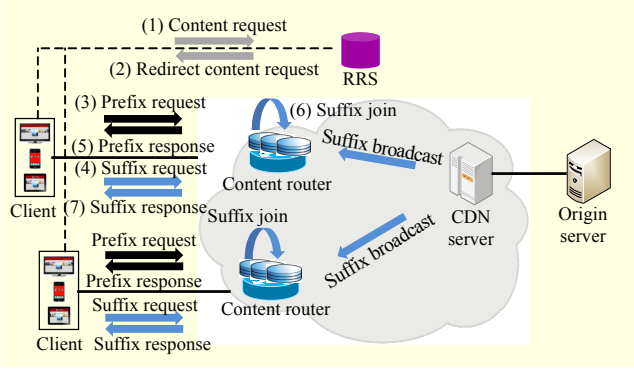

Fig. 3. Broadcast delivery scheme with in-network caching and operation.

Thus, the ISP is aware of what both the CDN server and content routers have cached [1], [15]. The CDN server can deliver the requested suffixes to the clients through multicast channels. In addition, the content router is a network element that acts as a regular router. It can also cache and deliver the in-network caching prefix to the client, though with a buffer of limited size, through unicast channels.

In Figs. 2 and 3, the persistent RRS is used to locate the best content router, for a particular client, while providing the granularity of the content chunk level in step 1. If the request is satisfied, then the RRS can return (in step 2) a status code, such as *HTTP 300 Multiple Choices*, in its response to inform the client of the new URIs of both the content routers and the CDN server. Such URIs also indicate the content name and its range — namely, the content chunk. This paper fundamentally assumes that content can be divided into two parts: a prefix and a suffix. The client should then simultaneously reissue its prefix and suffix requests with two or more *HTTP GETs* to the content routers and CDN server, respectively. If both can return the requested chunk (that is, prefix and suffix), then they do so in their response. They can indicate its success with the appropriate status code: *HTTP 206 Partial Content* [14]. Along with the status code, they include the chunk itself in their responses.

For simplicity, we assume that the clients always request playback from the beginning of the content and that prefixes are always available in the content routers. The content router can intercept client requests and deliver the prefix directly to clients. It then contacts the CDN server to issue a request for the suffix, and clients can, therefore, receive the remaining part of that content by joining the suffix streams at the content router. The content router will calculate the transmission and reception schedules so that the time and channel for transmitting and receiving the content are determined using the schedules [5], [8].

For efficient usage of the bandwidth, it is important to know of a video's popularity. There have been various studies related to video popularity. In [16], video popularity was reported to follow a Zipf distribution with skew factor 0.271; that is, 80% of the user's demand is for about 20% of the most popular videos and 20% of the user's demand is for the remaining 80% of the most popular videos. This fact helps with the design of the efficient delivery schemes, whereby we use a broadcast scheme for popular content and a multicast scheme for less popular content. In this sense, we assume that content popularity follows the Zipf distribution. Furthermore, we assume that information about the popularity of content is available by means of statistics and expectation.

In addition, previous studies exploring the distribution of multimedia files in CDNs used Zipf distributions to characterize the popularity of the different contents [5], [16]. Although the popularity of content does not exactly fit the Zipf distribution, many researchers still adopt the Zipf approach to model popularity in CDNs. With the aforementioned assumption, the costs in Fig. 4 are deduced by using (9) and (15).

We assume that costs associated with content routers are mainly linked to the delivery, rather than the caching, of content — a fact reflected by the trend in ever-decreasing storage costs. We also consider that there are enough channels in the CDN system so that the probability of running out of

Video content	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)	F(7)	F(8)	F(9)	F(10)	F(11)
	◄ L ►										
Server channel 1	F(1)	F(1)	F(1)	F(1)	F(1)	F(1)	F(1)	F(1)	F(1)	F(1)	F(1)
Server channel 2	F(2)	F(2)	F(2)	F(2)	F(2)	F(2)	F(2)	F(2)	F(2)	F(2)	F(2)
Server channel 3	F(3)	F(3)	F(3)	F(3)	F(3)	F(3)	F(3)	F(3)	F(3)	F(3)	F(3)
Server channel 4	F(4)	F(5)	F(4)	F(5)	F(4)	F(5)	F(4)	F(5)	F(4)	F(5)	F(4)
Server channel 5	F(6)	F(7)	F(6)	F(7)	F(6)	F(7)	F(6)	F(7)	F(6)	F(7)	F(6)
Server channel 6	F(8)	F(9)	F(10)	F(11)	F(8)	F(9)	F(8)	F(9)	F(10)	F(11)	F(8)

Fig. 4. Example of a fast broadcast (FB) scheme when partition function $f(n_i)$ and number of server channels ($K_i = 6$) are given.

such channels can be neglected. Some system parameters are identified from [17]–[19] as follows. We use N_v to denote the number of content types in the system and S as the total number of content routers. The available number of multicast channels in the CDN server is denoted by N_c, and L_i is the length (in minutes) of the ith content, where $1 \le i \le N_v$. Each request for content i arrives at content router s ($1 \le s \le S$) according to a Poisson process with a rate of $\lambda_{i,s}$ requests/min. The aggregate requests for content i and the overall external request rate are given, respectively, by

$$\lambda_i = \sum_{s=1}^{S} \lambda_{i,s} \tag{1}$$

and

$$\Lambda = \sum_{i=1}^{N_v} \lambda_i . \tag{2}$$

III. Efficient Content Delivery Scheme with In-Network Caching

1. Multicast Scheme with In-Network Caching

In the multicast scheme coupled with in-network caching, as shown in Fig. 2, let W_i be the prefix length for content i, which also corresponds to the patching window for in-network caching in content routers [5]. Suffixes (of length L_i) of content are stored and delivered from the CDN server by means of multicast channels, while prefixes (of length W_i) stored in content servers are delivered to clients through unicast streams.

When the first request arrives in the content router in steps 3 and 4, a patching window will be started for time interval W_i. The requests for the same content that arrive within the window will form a group, and then a single multicast from the CDN server is initiated by the first request and carried out to all clients in the group. Furthermore, since the range of the suffix always includes that of the prefix, the content router relays the suffix request to the CDN server in step 6, whereas in response to step 3, it issues the prefix response to the client with an *HTTP 204 No Content*. With an *HTTP 200 OK*, the CDN server immediately begins transmitting the suffix to the content router in step 7, where a copy of the suffix is transmitted to clients with an *HTTP 200* in step 8.

For the following requests that arrive later than the first request, the clients can obtain the missing initial portion through a patching stream with an *HTTP 206* in step 5. At the same time, they will obtain the rest of the content by tuning to an ongoing multicast stream with an *HTTP 206* in step 8. Once clients start to receive the content from a multicast channel, a patching stream will be released after receiving the missing part that the CDN server cannot transmit to the client. The patching stream is, therefore, "transient" in nature and of a short duration. For requests for the same content within the window, the content router repeatedly copies the suffix in proportion to the number of requests and then transmits it to the clients [14].

For the first request that arrives after the end of the patching window, it initiates a new window whereby the same operations should be repeated. Therefore, the average interval between successive multicast streams is given by $W_i + 1/\lambda_i$. The required number of multicast channels for the ith content is given by

$$M_i = \frac{L_i}{W_i + 1/\lambda_i}, \qquad 1 \le i \le N_v . \tag{3}$$

Since the expected prefix length of the patching stream is $W_i/2$, the total average number of channels allocated to the content routers is given by

$$U_M = \sum_{i=1}^{N_v} \frac{1}{2} \cdot \lambda_i \cdot W_i . \tag{4}$$

The problem of minimizing the total average number of channels allocated to the content routers is solved by determining the optimal value of W_i, subject to the constraint $\sum_{i=1}^{N_v} M_i = N_c$. Since L_i is the length of content i, we always have $L_i \ge W_i \ge 0$, and then $M_i \ge 0$ from (3). Given positive constants, the following optimization problem is formulated:

$$\begin{aligned} &(P1)\ \min \sum_{i=1}^{N_v} \frac{1}{2} \cdot \lambda_i \cdot W_i , \\ &\text{subject to } \sum_{i=1}^{N_v} M_i = N_c ,\ M_i \ge 0,\ 1 \le i \le N_v . \end{aligned} \tag{5}$$

The optimization problem ($P1$) has a unique optimal solution that can be obtained analytically. It follows from (3) that

$$W_i = \frac{L_i}{M_i} - \frac{1}{\lambda_i}, \qquad 1 \le i \le N_v . \tag{6}$$

By substituting (5) for (6), the problem ($P1$) can be rewritten as

$$\begin{aligned} &(P2)\ \min \sum_{i=1}^{N_v} \frac{\lambda_i}{2} \cdot \left(\frac{L_i}{M_i} - \frac{1}{\lambda_i}\right), \\ &\text{subject to } \sum_{i=1}^{N_v} M_i = N_c ,\ M_i \ge 0,\ 1 \le i \le N_v . \end{aligned} \tag{7}$$

When the Karush–Kuhn–Tucker (KKT) condition of ($P2$) is given, we can solve the optimal prefix length by setting $\partial(P2)/\partial M_i = 0$ and using the Lagrangian multipliers with respect to the equality constraint and inequality constraints. In our system model, we derived the optimal prefix length in (8) that minimizes the average number of channels allocated to the content routers for each content i from ($P2$). The optimal prefix length, which indicates the in-network caching size in the content routers, is given by

$$W_i = \frac{\sqrt{\lambda_i \cdot L_i} \cdot \sum_{k=1}^{N_v} \sqrt{\lambda_k \cdot L_k}}{\lambda_i \cdot N_c} - \frac{1}{\lambda_i} . \tag{8}$$

ETRI Journal, Volume 36, Number 5, October 2014
http://dx.doi.org/10.4218/etrij.14.0113.0090

From (3), we find that there is a trade-off between the prefix length and the number of multicast channels because having longer prefixes reduces the necessary number of multicast channels of the CDN server but increases the number of unicast channels of the content router. By combining (4) and (8), when in-network caching size W_i is cached in the content routers, the total average number of channels allocated to the content routers for content i is given by

$$U_M = \sum_{i=1}^{N_v} \left[\frac{\sqrt{\lambda_i \cdot L_i} \cdot \sum_{k=1}^{N_v} \sqrt{\lambda_k \cdot L_k}}{2 \cdot N_c} - \frac{1}{2} \right]. \quad (9)$$

2. Broadcast Scheme with In-Network Caching

Broadcast schemes, in general, are wasteful when the arrival rate is not high enough, since a broadcast channel is scheduled independent of any user request and dedicated to a video content [7], [20]–[21]. On the other hand, a broadcast scheme coupled with in-network caching, as shown in Fig. 3, not only significantly reduces the CDN server and network resource requirements but is also capable of immediately providing service to a large number of clients by taking advantage of in-network caching available at the content routers.

Before initiating the requests to the content routers, the CDN server periodically broadcasts video content to the content routers through a number of dedicated broadcast channels, as shown in Fig. 3. When the first request arrives in the content router in steps 3 and 4, it immediately joins an appropriate broadcast channel without waiting for the beginning of the next broadcast period in step 6. With an *HTTP 206 OK*, the content router immediately begins transmitting a copy of the suffix to the client (step 7). At the same time, the content router sends a response including the missing prefix of the video content with an *HTTP 206* to the client (step 5).

For the subsequent requests, the same operations should be repeated as such. Once clients start to consume the content from a broadcast channel, a patching stream will be released and the client keeps playing the remaining part from the broadcast channel.

FB is chosen to broadcast the video content in the system model because of the simplicity of the control system among broadcast schemes. The FB model [7], [21] has been introduced to address the scalability issue of video content delivery. The scheme makes the server I/O bandwidth usage independent of the number of clients at the expense of a bounded user waiting time.

The partition function $f(n_i)$, used to partition the video content into some segments, represents the relative length of each segment for content i. The FB divides the video content into a geometrical series of $(1, 2, 4, \ldots, 2^{n_i-1})$, where n_i is the number of broadcast channels for content i at the CDN server [7], [20]. We assume that the network bandwidth on the client side is only sufficient to support two channels at the same time. It is the same condition in the case of the multicast scheme. To satisfy this condition, the partition function $f(n_i)$ of an FB is slightly modified by

$$f(n_i) = \begin{cases} 1 & n_i = 1, 2, 3, \\ 2 & n_i = 4, 5, \\ 2f(n_i - 2) & n_i > 5. \end{cases} \quad (10)$$

An example of an FB scheme is shown in Fig. 4, where partition function $f(n_i)$ and number of server channels (K_i= 6) are given. Channel 1 broadcasts the first segment F(1) periodically, Channels 2 and 3 periodically broadcast segments F(2) and F(3), respectively. Channels 4 and 5 periodically broadcast the next two segments; that is, F(4), F(5) and F(6), F(7), respectively. Channel 6 periodically broadcasts the next four segments; that is, F(8), F(9), F(10), and F(11). The length of each segment is F_i for content i.

By adding two initial segments, a client can join only one broadcast channel while receiving the patching stream from the content router. For simplicity of exposition, we define the summation of the partition function $h(n_i)$ when the number of the server channel is K_i for content i.

$$h(n_i) = \sum_{n_i=1}^{K_i} f(n_i) = \begin{cases} 1 & n_i = 1, \\ 2 & n_i = 2, \\ 3 & n_i = 3, \\ (2^{(n_i-4)/2} \cdot 6) - 1 & n_i > 3,\ n_i \bmod 2 = 0, \\ (2^{(n_i-5)/2} \cdot 8) - 1 & n_i > 3,\ n_i \bmod 2 = 1. \end{cases} \quad (11)$$

Consider video content whose length is L_i. Given the partition function $f(n_i)$, suppose the number of broadcast channels at the CDN server K_i is dedicated to broadcast video content i and let F_i denote the length of the first broadcast segment at the content routers. From the definition of the partition function, we then have

$$L_i = F_i \sum_{n_i=1}^{K_i} f(n_i) = F_i \cdot h(n_i). \quad (12)$$

By setting the first segment of the suffix broadcast equal in size to the prefix length, the bandwidth usage on the long-haul path can be substantially reduced [7], [20]. From (12), we can see that there is a trade-off between the number of broadcast channels and the length of the first segment (that is, in-network caching size), since a smaller number of dedicated CDN server channels, K_i, will result in a larger first broadcast segment, F_i.

To minimize the average number of channels allocated to content routers, the length of first segment (that is, in-network caching size) should be minimized. This leads to the following

optimization problem:

$$(P3)\ \min\ U_B = \sum_{i=1}^{N_v} \frac{1}{2} \cdot \lambda_i \cdot F_i, \quad \text{subject to } \sum_{i=1}^{N_v} K_i = N_c,\ K_i \geq 0,\ 1 \leq i \leq N_v. \tag{13}$$

Using the trade-off between the first segment length, F_i, and the number of CDN server channels, K_i, ($P3$) is rewritten by

$$(P4)\ \min\ U_B = \sum_{i=1}^{N_v} \frac{1}{2} \cdot \lambda_i \cdot L_i \cdot \frac{1}{h(K_i)}, \quad \text{subject to } \sum_{i=1}^{N_v} K_i = N_c,\ K_i \geq 0,\ 1 \leq i \leq N_v.$$

When the KKT condition of ($P4$) is given, we can solve the optimal caching size F_i by setting $\partial(P4)/\partial F_i = 0$ and using the Lagrangian multipliers with respect to the equality constraint. One of these channels transmits only the first segment of the video content. The other channels transmit the remaining segments through their dedicated broadcast channels. The number of concurrent accesses to a CDN server is limited by the number of supportable multicast streams, K_i.

From ($P4$), the number of dedicated channels of the CDN server that minimize the length of the first segment is then given by

$$K_i = \left\lceil 2 \cdot \log_2 [\lambda_i \cdot (\frac{2^{N_c/2}}{\prod_{i=1}^{N_v} \lambda_i})^{1/N_v}] \right\rceil. \tag{14}$$

By combining (12) and (13), when in-network caching size F_i is cached on the content routers, the total average number of channels allocated to the content routers for content i is given by (15) and depends on the number of CDN server channels.

$$U_B = \sum_{i=1}^{N_v} \begin{cases} \frac{1}{2} \cdot \lambda_i \cdot L_i & K_i = 1, \\ \frac{1}{2} \cdot \frac{\lambda_i \cdot L_i}{2} & K_i = 2, \\ \frac{1}{2} \cdot \frac{\lambda_i \cdot L_i}{3} & K_i = 3, \\ \frac{1}{2} \cdot \frac{\lambda_i \cdot L_i}{(2^{(K_i-4)/2} \cdot 6) - 1} & K_i > 3,\ K_i \bmod 2 = 0, \\ \frac{1}{2} \cdot \frac{\lambda_i \cdot L_i}{(2^{(K_i-5)/2} \cdot 8) - 1} & K_i > 3,\ K_i \bmod 2 = 1. \end{cases} \tag{15}$$

3. Adaptive Scheme Based on Content Popularity

For efficient content delivery, it is important to know the popularity of the content in question. We assume that content, ranked according to popularity, can be divided into two groups; the content having mean arrival rates $\lambda_1, \lambda_2, \ldots, \lambda_{N_v}$, respectively, where N_v denotes the rank index of popularity.

Since a broadcast scheme is scheduled independent of any user request, the most popular content is likely to be transmitted through periodic broadcasting. On the other hand, the least popular content is, preferably, transmitted through multicasting because a multicast scheme will be scheduled only when the content is requested [21]. Therefore, the broadcasting of each video content demands one or more channels dedicated to it, while the video content delivered through multicasting usually share a pool of channels of the CDN server.

Owing to the skewed popularity, even among the most popular video content, a CDN system needs to be designed for carefully selecting an appropriate content delivery scheme and intelligently allocating resources between the content routers and CDN server. To account for the skewed popularity, we propose an efficient content delivery technique, called a popularity-based adaptive content delivery scheme, that selects either a broadcast scheme or a multicast scheme by considering content's popularity. The proposed adaptive content delivery scheme broadcasts the most popular content using the broadcast scheme, while delivering the least most popular content using the multicast scheme.

Given the total number of available channels (the capacity) of the CDN server, distributing them for individual broadcasting and the multicasting pool so as to achieve the optimal content delivery cost is a nonlinear optimization problem. The popularity-based adaptive scheme aims to minimize the average total number of unicast channels and the average caching size of the content routers, using dynamic programming, for a group of video content with highly skewed popularity. Depending on the relative popularity of the content, the adaptive content delivery scheme selects the most suitable delivery scheme for all content, and then it allocates the appropriate number of channels to each.

By taking advantage of the selection algorithm for determining the most suitable delivery scheme (see Fig. 5), the proposed adaptive scheme classifies the N_v pieces of content

Given number of server channels, N_c and number of content types, N_v
Determine number of channels and types allocated to broadcast, l and k
for all content request i **do**
if cost of broadcast, U_B < cost of multicast, U_M **then**
$k = k + 1$
end if
end for
for all content request $i \leq k$ **do**
$l = l + K_i$
end while
Determine number of channels and types allocated to multicast, $N_c - l$ and $N_v - k$
Content $0 \leq i \leq k$ with number of channels l belong to Broadcast
Content $k+1 \leq i \leq N_v$ with number of channels $N_c - l$ belong to Multicast

Fig. 5. Selection algorithm for determining the most suitable delivery scheme.

ETRI Journal, Volume 36, Number 5, October 2014
http://dx.doi.org/10.4218/etrij.14.0113.0090

into two groups according to their popularity; namely, the most popular content ($0 \le k \le N_v$) and the least popular content ($N_v - k$). The former group is assigned $0 \le l \le N_c$ channels for fast broadcasting, and the latter group is assigned the remaining $N_c - l$ channels for multicasting. Note that one of these groups will not exist if $k = 0, N_v$.

Once the specific values of k and l are calculated using the selection algorithm, the number of broadcast channels and multicast channels are determined by replacing N_v and N_c with k and l in (9) and (14). By applying either a multicast scheme or a broadcast scheme in consideration of content popularity, the minimum average number of channels of the content routers for the proposed adaptive scheme can then be achieved using the following dynamic programming formulation ($P5$):

$$(P5)\ \min_{\substack{0\le k\le N_v\\ 0\le l\le N_c}} \sum_{i=1}^{k} \frac{\lambda_i}{2}\cdot\frac{L_i}{h(K_i)} + \sum_{i=k+1}^{N_v}\frac{1}{2}\cdot\left(\frac{\sqrt{\lambda_i\cdot L_i}\cdot\sum_{j=k+1}^{N_v}\sqrt{\lambda_j\cdot L_j}}{N_c - l} - 1\right),$$

$$\text{subject to } \sum_{i=1}^{k} K_i = l,\ \sum_{i=k+1}^{N_v} M_i = N_c - l,\ K_i \ge 0,\ M_i \ge 0,\ 1 \le i \le N_v. \quad (16)$$

IV. Performance Analysis

In this section, we evaluate the performance of the proposed content delivery scheme, comparing to a multicast and a broadcast scheme with in-network caching. As many researchers [3], [22] have only showed performance gains over the core network for the introduction of content routers with in-network caching and different delivery schemes, we focus on performance from the perspective of in-network caching size, the number of streaming channels of content routers, and the number of streaming channels of the CDN server.

The performance analysis is based on the following system parameters: $s = 10$, $N_v = 200$, $N_c = 800$ to 1,000, $L_i = 90$ min, $\Lambda = 500$ requests/min, and $\lambda_i = \Lambda / (i^{1-\alpha}\sum_{j=1}^{N_v} 1/j^{1-\alpha})$ requests/min for $i = 1, 2, \ldots, N_v$. The relative popularity of the content follows a Zipf distribution with a skew factor of $\alpha = 0.271$. The above system parameters are still effective unless noted otherwise. Without loss of generality, let $\lambda_i > \lambda_j$ for $1 \le i < j \le N_v$; that is, content popularity decreases in accordance with the index. Here, the rank indexes 1 and N_v denote the most- and least-popular, respectively. The ranking index of content popularity $1 \le i \le N_v$ indicating the popularity, is used on the x-axis instead of the arrival rate, λ_i, since it can help to clearly understand the different in-network caching size on the y-axis. The values on the x-axis in the following figures indicate the ranking index of the content popularity, corresponding to arrival rate λ_i in Figs. 6–8.

Figure 6 compares the optimal average in-network caching size of a multicast scheme for different numbers of CDN server channels (N_c = 800, 900, and 1,000), leading to a minimization of the number of unicast patching channels allocated to content routers. The number of CDN server channels is chosen within the range of the aforementioned N_c values to clearly differentiate the performance of multicast and broadcast schemes, since the latter always outperforms the former when N_c is larger than 1,100. As the content popularity decreases, a larger caching size is gradually needed. The caching size changes from 5 (min) to 33 (min) for different numbers of CDN server channels at arrival rate Λ = 500 (requests/min). With delivering the caching portion of the least popular content from content routers, the required capacity of the CDN server for the least popular content is reduced. The gain can, therefore,

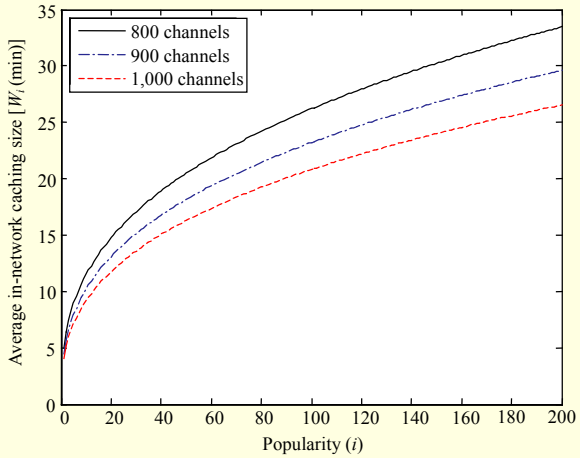

Fig. 6. Average in-network caching size inside content routers via a multicast scheme for different number of CDN server channels.

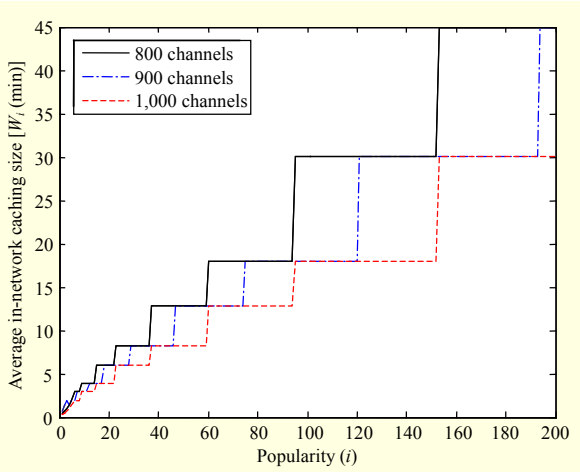

Fig. 7. Average in-network caching size inside content routers via a broadcast scheme for different number of CDN server channels.

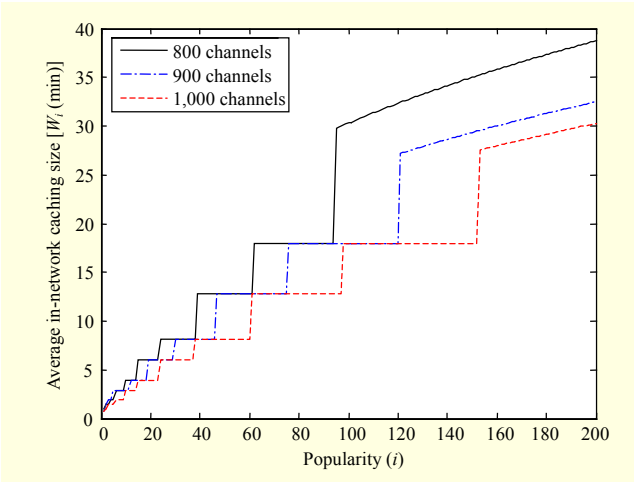

Fig. 8. Average in-network caching size inside content routers via the popularity-based adaptive content delivery scheme for different number of CDN server channels.

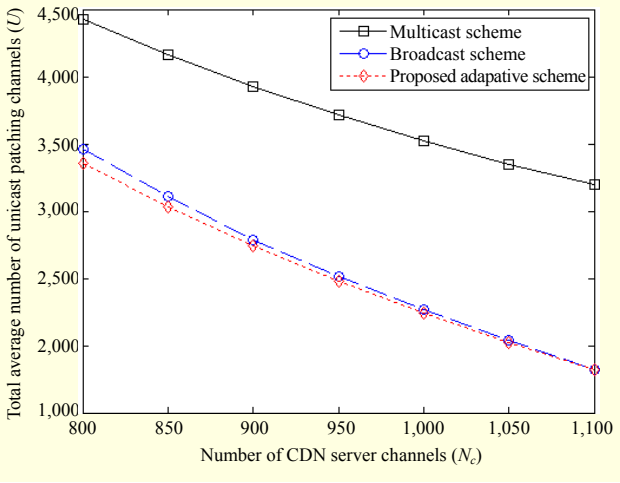

Fig. 9. Comparison of the average number of unicast patching channels for different numbers of CDN server channels.

be used to deliver the more popular content. On the other hand, the caching portion of the most popular content decreases as the number of CDN server channels increases. From the above observation, we identify that a trade-off exists between the capacity of the content router and the CDN server.

Figure 7 shows the optimal average in-network caching size of a broadcast scheme for different numbers of CDN server channels, minimizing the number of unicast patching channels allocated to content routers. Similar to a multicast scheme, the caching size increased in step-up style. The caching size changes from 1 (min) to 45 (min) for different numbers of CDN server channels at arrival rate $\Lambda = 500$ (requests/min). Compared to a multicast scheme, the caching size is smaller for content of high popularity but is larger for content of low popularity. The largest occurring caching size, $F_i = 45$ (min), was equal to half of its content's playback time. The storage capacity of content routers is mainly occupied by the least popular content.

Figure 8 illustrates the optimal average in-network caching size of the popularity-based adaptive content delivery scheme for different numbers of CDN server channels, minimizing the number of unicast patching channels allocated to content routers. We observe that the proposed adaptive scheme requires a total average storage of 3,158 (min), whereas the multicast scheme requires 3,939 (min) and the broadcast scheme requires 3,265 (min) for all content when the number of CDN server channels is 1,000. The proposed adaptive scheme improves the required storage capacity of the content routers compared to the multicast and broadcast schemes by about 19% and 3%, respectively. From Fig. 5 and (16), the proposed adaptive scheme switches over from a broadcast scheme to a multicast scheme when popularity rank index i is between 154

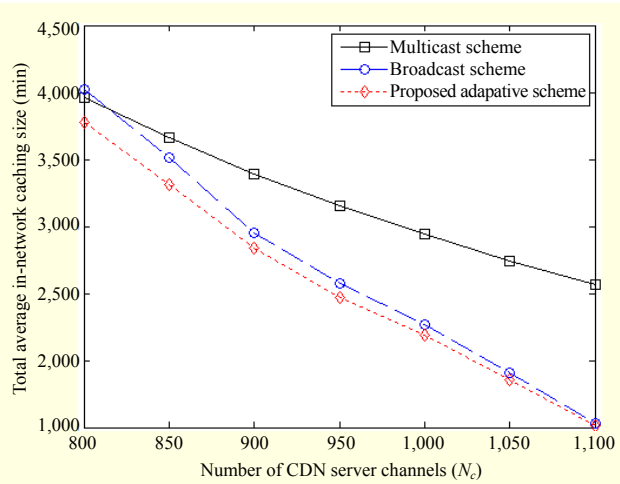

Fig. 10. Average in-network caching size inside content routers via a multicast scheme for different number of CDN server channels.

and 200. We can, therefore, achieve the optimal in-network caching size when applying the proposed adaptive scheme since the caching size of the broadcast scheme suddenly increases from popularity index rank 154, compared to that of the multicast scheme.

The performance of the proposed scheme is compared for all content in terms of the average numbers of channels allocated to the content routers, as shown in Fig. 9. The proposed adaptive scheme requires an average of 2,236 channels at the content routers, whereas the multicast and broadcast schemes require 3,522 and 2,270 channels, respectively. By applying the proposed adaptive scheme, we can reduce the required number of channels compared to the other schemes by up to 36%.

Figure 10 illustrates a comparison of the average total in-network caching size allocated to the content routers for

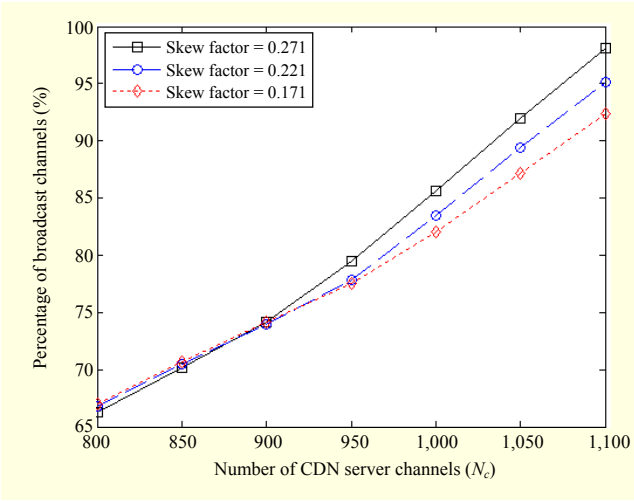

Fig. 11. Fraction of content delivered via broadcast channels for different skew factor in adaptive content delivery scheme.

different numbers of multicast channels of the CDN server. The average total caching size of the adaptive scheme is close to that of the multicast scheme when the number of channels of the CDN server is the smallest; that is, at $N_c = 800$. Otherwise, when it gradually increases, we observe that the average total caching size of the proposed adaptive scheme is almost close to that of the broadcast scheme.

The fraction of content delivered via broadcast channels for different skew factors in the adaptive content delivery scheme is shown in Fig. 11. The fractions are distributed very similar to each other, regardless of the different skew factors, when the number of channels of the CDN server is between 800 and 900. On the other hand, when the number is above 950, the fractions are distributed with more and more diversity as the skew factor increases. In particular, the fractions approach 97% when skew factor α is 0.271.

The results of the performance analysis in this section show that the adaptive scheme considerably outperforms other schemes by considering the content popularity, since the most popular content is delivered through broadcast channels and the least popular through multicast channels.

V. Conclusion

This paper proposed the popularity-based adaptive content delivery scheme in a hybrid CDN system that takes advantage of the traditional CDN server in the overlay and novel content routers in the underlay, while adopting in-network caching in the content routers. By employing the proposed scheme, content routers can adaptively select the most suitable delivery scheme and allocate the appropriate number of channels to efficiently minimize both their streaming and storage capacities for all content, depending on the relative popularity. We showed that the proposed scheme provides a notable performance gain against both the multicast and broadcast schemes coupled with in-network caching in terms of the optimal in-network caching size and number of unicast channels in a content router.

References

[1] D.D. Vleeschauwer and D.C. Robinson, "Optimum Caching Strategies for a Telco CDN," *Bell Labs Tech. J.*, vol. 16, no. 2, Sept. 2011, pp. 115–132.

[2] K. Cho et al., "How Can an ISP Merge with a CDN?," *IEEE Commun. Mag.*, vol. 49, no. 10, Oct. 2011, pp. 156–162.

[3] G. Haßlinger and F. Hartleb, "Content Delivery and Caching from a Network Provider's Perspective," *Comput. Netw.*, vol. 55, no. 8, Dec. 2011, pp. 3991–4006.

[4] W.K.S. Tang et al., "Optimal Video Placement Scheme for Batching VOD Services," *IEEE Trans. Broadcast.*, vol. 50, no. 1, Mar. 2004, pp. 16–25.

[5] 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.

[6] L. Gao and D. Towsley, "Threshold-Based Multicast for Continuous Media Delivery," *IEEE Trans. Multimedia*, vol. 3, no. 4, Dec. 2001, pp. 405–414.

[7] L. Gao, J. Kurose, and D. Towsley, "Efficient Schemes for Broadcasting Popular Videos," *Multimedia Syst.*, vol. 8, no. 4, July 2002, pp. 284–294.

[8] S.H. Gary Chan, "Operation and Cost Optimization of a Distributed Servers Architecture for on-Demand Video Services," *IEEE Commun. Lett.*, vol. 5, no. 9, Sept. 2001, pp. 384–386.

[9] Van Jacobson et al., "Networking Named Content," *Proc. CoNEXT*, Tokyo, Japan, Dec. 2011, pp. 1–12.

[10] D. Eager, M. Vernon, and J. Zahorjan, "Minimizing Bandwidth Requirements for on-Demand Data Delivery," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 5, Sept.–Oct. 2001, pp. 742–757.

[11] J.Y. Kim, G.M. Lee, and J.K. Choi, "Efficient Multicast Schemes Using In-Network Caching for Optimal Content Delivery," *IEEE Commun. Lett.*, vol. 17, no. 5, May 2013, pp. 1048–1052.

[12] A. Barbir et al., "Known Content Network (CN) Request-Routing Mechanisms," RFC3568, July 2003.

[13] M. Masa and E. Parravicini, "Impact of Request Routing Algorithms on the Delivery Performance of Content Delivery Networks," *IEEE Int. Performance, Comput. Commun. Conf.*, Apr. 9–11, 2003, pp. 5–12.

[14] S.A. Thomas, *HTTP Essentials*, Hoboken, NJ: John Wiley & Sons, 2001.

[15] I. Psaras et al., "Modelling and Evaluation of CCN-Caching Trees," *Proc. IFIP Netw.*, Valencia, Spain, 2011, pp. 78–91.

[16] J. Choi, A.S. Reaz, and B. Mukherjee, "A Survey of User Behavior in VoD Service and Bandwidth-Saving Multicast Streaming Schemes," *IEEE Commun. Surveys Tutorials*, vol. 14, no. 1, 2012, pp. 156–169.

[17] G. Xue, "Server Cost Minimization in a Distributed Servers Architecture for on-Demand Video Services," *IEEE Commun. Lett*., vol. 7, no. 9, Feb. 2003, pp. 52–54.

[18] D. Guan and G. Xiong, "Optimal Prefix Cache Allocation among Multiple Cooperative Local Proxies," *Int. Conf. Wireless Commun. Netw. Mobile Comput*., Beijing, China, Sept. 24–26, 2009, pp. 1–4.

[19] L. Dong et al., "Performance Evaluation of Content Based Routing with In-Network Caching," *Wireless Opt. Commun. Conf*., Newark, NJ, USA, Apr. 15–16, 2011, pp. 1–6.

[20] L. Gao, Z.–L. Zhang, and D. Towsley, "Proxy-Assisted Techniques for Delivering Continuous Multimedia Streams," *IEEE/ACM Trans. Netw*., vol. 11, no. 6, Dec. 2003, pp. 884–894.

[21] 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.

[22] Y. Kim and I. Yeom, "Performance Analysis of In-Network Caching for Content-Centric Networking," *Comput. Netw*., vol. 57, no. 3, Sept. 2013, pp. 2465–2482.

Jeong Yun Kim received his BS and MS degrees in electronic engineering from Inha University, Incheon, Rep. of Korea, in 1990 and 1992, respectively and received his PhD degree in information and communication engineering from the Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea, in 2014. Since 1992, he has been with the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea as a special fellow. His main research interests are Future Internet, streaming services, and energy saving technologies including smart grids. He has actively participated in standardization meetings including ITU-T SG 13 (Future Networks & Cloud) as an editor and IETF. He has contributed more than 200 proposals for standards and published more than 50 papers in academic journals and conferences. He is a member of the IEEE.

Gyu Myoung Lee received his BS degree in electronic and electrical engineering from Hong Ik University, Seoul, Rep. of Korea, in 1999 and his MS and PhD degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 2000 and 2007. In 2007, he worked as a guest researcher at the National Institute of Standards and Technology, Gaithersburg, MD, USA. Later that year, he was invited to work on the research staff at the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. In 2008, he was with the Institut Mines-Telecom, Telecom SudParis, Evry, France, as an adjunct associate professor. Then in 2012, he continued his work as an adjunct professor at KAIST, Daejeon, Rep. of Korea. Recently he has been employed as a Senior Lecturer at the Liverpool John Moores University, Liverpool, UK. His research interests include Internet of things, future networks, multimedia services, and energy saving technologies including smart grids. He has actively participated in standardization meetings, including ITU-T SG 13 (Future Networks & Cloud) as a rapporteur, oneM2M, and IETF. He has contributed more than 300 proposals for standards and published more than 100 papers in academic journals and conferences. He is a senior member of IEEE.

Jun Kyun Choi received his BS degree in electronics from Seoul National University, Seoul, Rep. of Korea, in 1982, and his MS and PhD degrees from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 1985 and 1988, respectively. He worked for ETRI from 1986 to 1997 and is currently working as a professor at KAIST.