Multimed Tools Appl (2017) 76:21519–21546
DOI 10.1007/s11042-016-4008-8

Management of scalable video streaming in information centric networking

Saeed Ullah[1] · Kyi Thar[1] · Choong Seon Hong[1]

Received: 25 March 2016 / Revised: 31 August 2016 / Accepted: 23 September 2016 /
Published online: 29 October 2016

Abstract Ability of caching the contents is one of the most important feature of an Information Centric Networking (ICN) node. By managing the cache space intelligently we can improve network's performance and increase users' Quality of Experience (QoE). Moreover, scalable video streaming in ICN is envisioned to be very beneficial as well as a challenging issue. In this paper, we are proposing a mechanism for cache management and request forwarding policies for scalable video streaming in ICN. Our proposed cache decision policy ensures to cache the base layer of a scalable video near to the users, which is mandatory layer for decoding any SVC encoded video and is needed by all the users with any data-rate budget, and consequently cache the higher layers in the upper nodes in the CCN/ICN within a specific RTT range. Furthermore, our intelligent cache decision cover fairness by considering router's cache capacity (inside the RTT range) and at the same time giving more priority to the nodes that are nearer to the users. A limited cooperative request forwarding mechanism, which is the part of our proposal, plays a role to improve users' QoE by providing the popular requested contents quickly. We have intensively simulated our proposed cache management and request forwarding scheme. The simulation results show that our proposed solution outperforms the current cache management schemes and improve the cache utilization. Also our proposed scheme decrease the traffic flowing inside the network by eliminating the request flooding and providing the requested contents from the nearby location to the users. The proposed scheme provides video faster to the users, specially the mandatory base layer is provided very quickly to the users.

✉ Choong Seon Hong
cshong@khu.ac.kr

Saeed Ullah
saeed@khu.ac.kr

Kyi Thar
kyithar@khu.ac.kr

[1] Department of Computer Science and Engineering, Kyung Hee University, Yongin, Republic of Korea

Keywords Information centric networking (ICN) · Scalable video streaming · Cooperative caching in CCN · Request forwarding

1 Introduction

Internet has grown many folds in the last couple of decades and this growth is to be continued. The current Internet structure which is end-to-end communication between users and content server is not suitable to provide quality services to users because it does not fully utilize the enriched resources of modern networking nodes i.e., routers. To take full benefits from the advanced routers, Information Centric Networking (ICN) [1] is introduced.

Content Centric Networking (CCN) [8] is one of the premium instance of ICN on which we are focusing in this paper. Among many features of CCN the most attractive is the facility of content distribution by further utilization of in-network data storage and computational resources. CCN users send Interest packet that contains name of the Data chunk, user is needing. Every CCN router on the way, checks it Content Store (CS) to find the requested Data chunk. If requested chunk is present there, the requesting node is replied with the Data chunk on the reverse path and the Interest is discarded. Otherwise, the Interest is forwarded towards the content provider if the outgoing face is present in Forward Information Base (FIB). The Interest is flooded if no such face is founded in the FIB that could lead it towards the requested Data. Information of the unresolved/forwarded Interests is saved in Pending Interest Table (PIT) in order to handle the corresponding Data chunk as well as to aggregate similar Interests.

On the other hand, According to [5] globally, total Internet video traffic will be 77 % of all Internet traffic in 2019, up from 59 % in 2014. 50.8 % of that will be HD video, which was 30.5 % in 2014. Scalable Video Coding (SVC) based adaptive video streaming [33], which we call Scalable Video Streaming (SVS) in this paper, is considered very promising for video delivery over the Internet because it provides video to different users with different quality from a single file. SVC encodes video in layers which is consisted of a mandatory Base Layer (BL) and several optional Enhancement Layers (ELs). Users are provided as many layers as they can afford according to network situation and/or their device capabilities. Quality of the video is dependent on the number of layers the user downloads, the more layers the better quality video and vice versa.

In SVC based adaptive video streaming there is strong dependency between the layers. In order to use higher layer(s) all the lower layers must be present. Furthermore, data downloaded is proportional to the number of layers downloaded, more layers downloading means more data downloading. Therefore, some users who have low specifications devices or low speed access network will request only BL. Users with higher specs devices and higher speed access network will request BL along with some EL(s) and users with high specs devices and highest speed access network will request all the layers present in a video. This phenomenon encourages us to cache the BL nearer to the users, as it is needed by many users, and ELs away from the users according to their hierarchy.

In this paper, we propose a caching mechanism in which the edge router specify a Round Trip Time (RTT) range on the bases of layer requested in the Interest. The corresponding Data chunk is cached by one router inside the specified RTT range. Routers inside the RTT range take decision probabilistically based on its cache capacity with respect to the total cache capacity of all the routers in the specified RTT range, to cache the content in its CS. Selection of the RTT range is dependent on the layer of the requested video. BL is cached in routers near to the user while the ELs are cached up in the hierarchy. Furthermore, we also propose a cooperative Interest forwarding mechanism, according to which if local popularity of a cached content reaches to a threshold, it informs one hope neighbor node about it and if a request comes to the neighbor for this content, rather than forwarding the Interest towards the server, it directs the request to the node have cached the content. Thus we utilize the cache more efficiently and reduce traffic inside the network.

We have extended chunk level simulator ccnsim [30] to implement our proposed mechanism. We have performed intensive simulations to compare our proposal with other similar proposals in the literature. Our proposed solution show that the mandatory BL is provided very quickly to the users. Also our results show reduced traffic inside the network and better cache hit rate inside the network.

Rest of the paper is organized as follows. In Section 2 we cover some related work, while in Section 3 we give some background of CCN and SVS and have derived motivation for this paper. Section 4 discusses the system architecture and assumptions. We present our proposed mechanism in Section 5 and performance analysis of the proposed scheme, as compared to the other similar proposals, is covered in Section 6. The paper is concluded in Section 7.

2 Related work

In this section we cover some of the related work done in the field of video streaming and caching/forwarding in CCN. In order to deliver the video over the network, many things need to be considered. Like, users' device capabilities, network condition etc,. Furthermore, network condition usually changes over the time and also differ from user to user. These things can be achieved in two ways, i.e., by using Adaptive Bitrate Streaming (ABS) [40] or Scalable Video Streaming (SVS) [33]. In ABS a single video source file is transcoded into multiple files with different scalability like, spatial, temporal and SNR (Signal to Noise Ratio). Video is provided to the user from a file that is suitable for the network condition and/or users device capabilities. ABS is used for video delivery over the network very widely. Different companies having their own proprietary standards for ABS like Microsoft's smooth streaming, Adobe dynamic streaming for flash and Apple HTTP Adaptive Streaming etc.,

In the recent past ABS is standardized as MPEG-DASH [34]. Instead of writing one big file for each video quality, MPEG-DASH store small segments for each quality. Because of these small segments, changing from one quality to another quality is very easy. MPEG-DASH is a pull based scheme, in which user is provided a meta-data file which contains information of all the available qualities of the video in the server. User analyze the network condition and on the basis of it and his device capabilities, requests the most suitable piece of video. MPEG-DASH has been proposed for CCN in [15] and been research very much in ICN in the last three years. On the other hand, SVS which provide the scalability through layering, as we have discussed in detail in Section 3.2, is envisioned to be very beneficial for CCN [12, 24]. SVC encoder transcodes video in layers and store all the layers in one file. Users are provided as much layers as they need from this file. Thus, it reduces video storing space by a great margin as compare to ABS in which for each quality a separate file is stored. SVS provides storage space efficiency to the content provider i.e., server. Similarly, it provides cache space efficiency in CCN [12].

Many researchers have taken interest in dealing with ABS as well as SVS in CCN in the very recent past. In [19], the authors presented time-shifted TV in CCN and proposed cooperative in-network caching using hash based scheme for video contents. In [20], the authors combined the traditional hash-based and directory-based cooperative caching schemes to treat the large video streams with on-demand access. In [17], the authors proposed an algorithm for SVC in which the enhancement layer(s) downloading is discarded if they are unable to be downloading in a specific time. In [16] and [15] the authors introduced DASH in CCN and evaluated its performance using the real-world mobile bandwidth traces to compare it with other proprietary systems. In [7], the authors raised the issue of RTT fluctuation in CCN and proposed a new timeout estimation algorithm to quickly adjust the timeout value and perform retransmission of video streaming in CCN in wireless networking environment. In [16], the authors considered exploiting the CCN characteristic of providing the contents from multiple sources as well as over multiple links in parallel. The authors proposed to choose the best performing link for downloading the multimedia contents. In [29], instead of supposing per-defined popularity for video contents, the authors find the important points in a video and proposed algorithm to give priority to cached these important points. Network traffic and energy efficiency have an important impact on the video, which is considered in [9, 10] and [11].

Cache Decision is a strategy in which the router decides whether to cache Data (Content Object) or not to cache. Several different type of cache decision algorithms are proposed for different purposes. Such as Leave Copy Everywhere (LCE) and Leave Copy Down (LCD) [39]. Caching scheme of the original CCN [8] is LCE, which is very simple but it causes redundancy of Data in the network that effect the caching efficiency. By using LCD, duplicate data can be reduce on the request path. LCD is proposed in [13] for web caching. Progressive caching [14] extended the LCD to cache popular chunks only. In [28], the authors have used probabilistic caching scheme termed as ProbCache. The probability of caching the content is increasing as content is passing on a path towards the user.

Forwarding Strategy is to make a decision to choose the interface(s) of current router to forward the Interest, when the Interest matching is unsuccessful in the CS and no entry for the same Interest is found in PIT. In original CCN [8], routers flood the Interest to all physical interfaces to find the Data inside the network. In [19] and [3], each router has ranked the interfaces and choose the best path to retrieve the requested Data. If the requested Data is not delivered by the best path, router selects the second best interface to forward the Interest again. Hash-based routing is used in [20, 31] and [38]. In [41], the same number of routers are formed as groups or clusters and each group stored Data in dedicated router by using modulo hashing. One of the major problem in this method is network scalability. Another problem is the load balancing, i.e., some routers are dumped with the Data and other remain very less loaded.

In [6], authors consider network as one AS in which edge nodes make decision to store and forward the Data. By using this scheme, if the AS is large, performance will decrease and the network can face the scalability issue. In [4] the authors have tried to solve caching problem of duplicate chunks around one hop neighbors by cooperative redundancy elimination method. In [41], the authors have introduced the Availability Info Base (AIB) to know which router stores which Data.

In [35], the authors have classified the web caching replacement strategies as five main categories, recency-based, frequency-based, recency/frequency-based, function based and randomized strategies. Least Recently Used (LRU) is based on recentness of the Data. When the router use LRU, the least recently access Data is replaced with the new Data in the situation when cache is fulled. Ponnusamy and Kathikeyan [29] have used Function based strategies which used Greedy dual algorithm for Data replacement in the cache.

In ICN/CCN, the bandwidth cannot be predicted precisely if caching exists. This bandwidth unpredictability severely affects the video quality selection of AVC-based encoded videos. [24] has introduced the usage of SVC-based encoded videos in ICN/CCN to overcome the quality selection problem of AVC. The paper shows advantages of SVC in ICN over AVC both theoretically as well as through simulations. In [2], the authors have discussed the potential of ICN paradigm as a networking solution for connected vehicles. Their analysis shows that ICN structure can benefit VANETs in getting their goals and applications.

Authors of [18, 21] have proposed caching the dynamic adaptive streaming contents at the optimum locations in the network. They have termed their proposal as DASCache. The proposed scheme is binary integer programing based optimization problem for updating cache of all the networking nodes periodically. They have solved the optimization problem centrally, considering Zipf distribution based popularity of the contents. However, solving the optimization problem for a very large network may not be feasible in CCN environment, where caching decision is needed to be taken at the line-speed. We have compared our proposal with DASCache, the results are discussed in Section 6.3.

3 Background and motivation

In this section we provide an overview of the CCN architecture and scalable video streaming encoding/decoding mechanism to give a better understanding of this work to the readers. At the end of the section, we draw a motivation for the proposed mechanism.

3.1 Content centric networking (CCN)

Current Internet is IP based which is location specific. CCN changes the architecture of current Internet by making it information specific. In CCN users generate request packet, called Interest, whenever they need a piece of information. Interest packet is carrying name of the required data. These names are variable-length hierarchal identifiers similar to URIs or the file system path names like a/b/c.mpg etc. These hierarchal names are used to direct the Interest towards the content provider [8]. On reception of the Interest message, the content provider reply with the Data packet which is also called Data chunk which is sent back on the reverse path. CCN routers on the path, cache the Data chunk in order to provide it to similar requests in the near future. The content caching capability of CCN router helps in reducing traffic inside the network by replying with the requested content locally if a copy is present in its cache.

There are three main data structures in a CCN router namely Content Store (CS), Pending Interest Table (PIT) and Forward Information Base (FIB). When an Interest is received by a CCN node/router first it checks the local CS, if a copy of the requested content is present it sends it on the reverse path and discards the Interest. If CS search in unsuccessful, the router checks PIT for a similar Interest that is forwarded and the Data delivery for which is still pending. If an entry is already present for the same Data chunk in PIT, the face on which Interest is received is added to the PIT entry and the Interest is discarded. When the router receives the corresponding Data packet it duplicates the Data packet on all the requested ports. In case the router does not find PIT entry, it searches the FIB and forwards the Interest on the potential face(s) that can lead towards the content provider. The Interest is flooded in the network via broadcasting on all the faces if there is no information available in the FIB about the potential content server.

Data delivery is relatively simple. Every router on receiving Data, checks PIT entry. If PIT entry is present the router sends Data on all the faces and also cache a copy of it in

CS according to its caching policy. If no PIT is found for the received Data, it is discarded because it is either considered redundant or is considered as out dated. Interested readers are directed to [8] for further details of CCN.

3.2 Layered scalable video streaming

In this paper we are focusing on video objects that are encoded with H.264/SVC (Scalable Video Coding (SVC)) [32]. We call the video stream as Scalable Video Streaming (SVS) which is encoded with H.264/SVC. We use SVC and SVS interchangeably throughout this paper.

H.264/AVC (Adaptive Video Coding (AVC)) [22] encodes a single video in multiple bitrates and resolution and store each instance in a separate file. Video from appropriate file is provided to the users according to their needs/requests or according to their device/bandwidth capabilities. Unlike AVC, SVC provides scalability by encoding the video in a single file. Strength of SVC is that it enables a single video stream to support different users with various device capabilities/requirements like, screen resolution, bandwidth requirements, playback quality etc.

SVC achieves scalability by encoding video into multiple layers. Each SVC encoded video has a mandatory Base Layer (BL) and optional multiple Enhancement Layers (ELs) [33]. BL is encoded with the lowest temporal, spatial and quality parameters of the video stream. Only BL is capable of decoding the video with the minimum quality. Enhancement layers are used for to increase temporal quality (i.e., video frame rate), spatial quality (i.e., resolution) or overall quality improvement. Users fetch base layer and additional enhancement layers according to its device capability and/or network condition. Layers of the SVS are strongly interdependent. To use any layer the decoder must have all the lower layer present with it, i.e., enhancement layer 1 can only be used if the base layer is present and enhancement layer 2 can only be used if base layer and enhancement layer 1 are already present and so on. SVS can be more beneficial in CCN than in the IP network because of the caching capability of CCN routers [12]. There will be less duplicate streams for the same video inside the network as a result of utilizing the previously cached Data inside CCN routers.

3.3 Motivation for the proposed scheme

SVC encodes a video in a single file and users are provided as many layers as they need according to their network connectivity and device capabilities. Li et al. [21] has shown superiority of SVC over AVC in CCN environment. Figure 1 is showing an example of delivering one video to 3 different users, with different device capabilities and network connectivity, over the Internet. Let's suppose smart phone is using $3G$ laptop is using WiFi and desktop is using wire connection to access the Internet. Here, we can see that a user with smart phone will need lower quality video because of its device specification and network capacity. So it is provided base layer of the video only. User with the laptop can afford higher quality video because of higher specification device and better connection, so it is provided an additional enhancement layer (EL1) along with base layer. The desktop user is provided highest quality (BL with 2 ELs) because of high end device and high speed link. Here the important thing to notice is that BL is needed by all the users and EL1 by lesser users and EL2 by more lesser users.

We use this hierarchal popularity of the video layers as the baseline guidance for proposing the cache decision policy in this paper which we present in the next session. Popularity

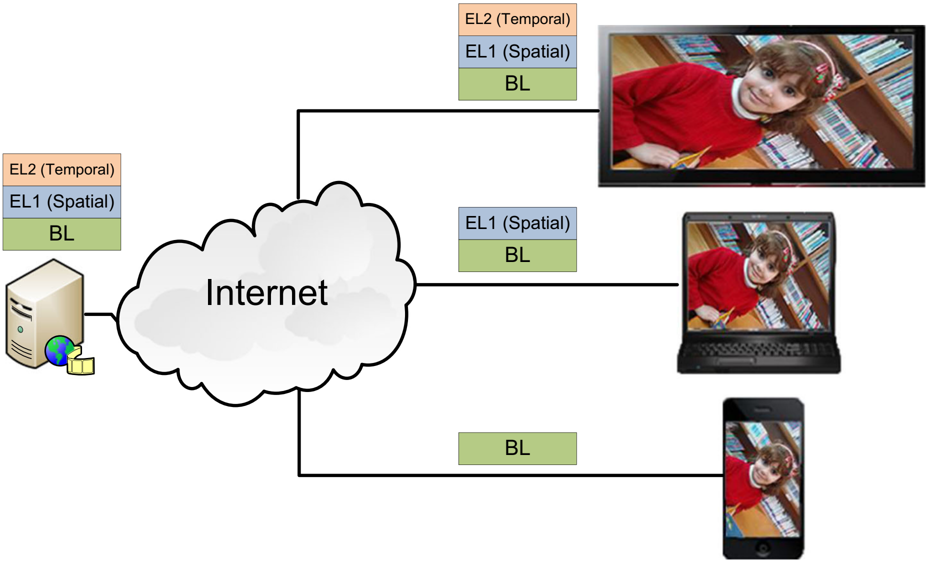


Fig. 1 Layered video delivery to various devices with different capabilities/bandwidth

based cache decision may not directly apply to SVS because it consider the whole one content as a unit and assigns a single popularity value to it. While in SVS different parts of a single video file have different popularity as been discussed in the above example. This phenomenon encourages us to cache the BL (the most popular) as near to the users as possible and ELs (lesser popular) faraway from the users according to their hierarchy.

On the other hand, as we have discussed in Section 2, each request forwarding scheme has its own pros and cons. In this paper we are presenting a light-weighted cooperated Interest forwarding scheme. According to this scheme, when number of hits for a specific content reaches to a threshold in a CCN node, the node give information of this content to its parent node. The parent node stores this information in a table that we call Cooperated Popular Content Store (CPCS) in order to direct request for this content to its child node. This scheme will decrease the traffic inside the network by exploiting the cache of neighbor nodes.

4 System architecture and assumptions

Our proposed cache management and Interest forwarding scheme for scalable video streaming is for a group of ICN nodes that are under a single administrative domain or simply an Autonomous System (AS). We are considering tree topology for our proposed scheme like used by [23, 26, 37], in which the content provider (server) is attached with the root of tree and users are connected with the leaf nodes as shown in Fig. 2. The proposal is applicable on any kind of tree topology. A request path is consisted of all the nodes between a user and server of the requested content or the node where cache hit occurs. Our proposed scheme caches each video layer V_k in a specific RTT range according to the layer k on the request path. The edge router extracts layer information from the Interest name and make decision

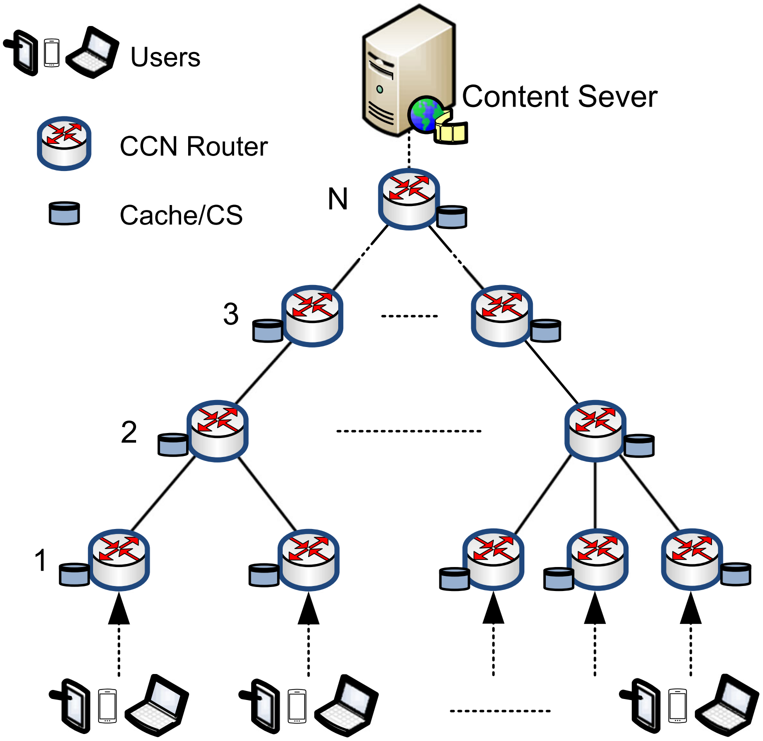


Fig. 2 System diagram

on the bases of requested layer of the video. For SVS, as suggested in [12], the Interest name is modified by adding layer ID information after the segment number in Interest packet e.g., stream/segments/%00%01/layer1. If the last component (layer1 in the above example) is not mentioned in an Interest, it by default will mean the user is requesting BL. The publisher/server needs to append information of all the available layers to BL Data packet in order to keep the playback synchronized. Therefore, when a user gets the BL of a video, it also receives complete knowledge of all the available ELs in the server for that specific video. To request ELs, the user is required to issue explicit Interest packet(s) mentioning requesting layer. In order to make our proposal workable, we need some extra parameters to be carried by the Interest and Data packets. These parameters and their usage procedure is discussed in Section 5 (Table 1).

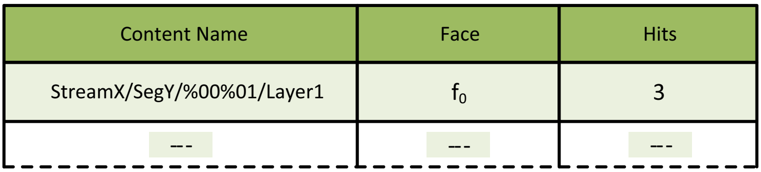
In our proposed scheme the Data chunk is cached at most by only one node on the return path of Interest. The baseline goal of our system design is to cache the important parts of video, that is needed by a large number of users, in a place that is nearer to the users. For this purpose, our proposed system specifies a range of RTT for each layer. The BL is kept in the range that includes the edge router/node and nearby routers. Similarly, the upper layers are cached up in the hierarchy. In order to keep the much needed content nearer to the users, even inside the RTT range, more weightage is given to the routers nearer to the users for caching the chunk. Additional fairness is achieved by giving weightage to the routers according to their cache capacity inside the range for caching the contents.

Table 1 Symbols and abbreviations

Symbol	Meaning
k	Requested layer of video content($k = 1$ to K)
K	Total layers of video content
c_i	Cache capacity of a node i
C_T	Cache capacity of total routers in γ_k range
C_R	Cache capacity of the remaining routers in γ_k range
γ_k	RTT range for layer k
RTT_N	Total RTT from client to server
t_s^i	The time when Interest is received by edge node i
t_c^i	The time Interest is received by node i
M	Total contents
β	Threshold of cache hit for copy down and CPCS update
α	Zipf parameter
N	Total nodes in a path from edge node to the server
CI	Cache Indicator
CM	Cache Marking
CH	Cache Hits (a counter in CS)
$CPCS$	Cooperated Popular Content Store
IT	Interest Type
TD	Type of the Data Chunk
n	Total nodes in a γ_k range
V_i^k	Layer k of content V_i

5 Proposed cache management scheme

Our proposed scheme is consisted of mainly three parts. First is cache decision process in which an RTT range is defined for caching the content and then routers inside the rage decides whether to cache the contents or not. Second part is the Interest packet forwarding i.e, to decide on which face the Interest to be forwarded and third part is the cache eviction response, i.e., in case of deleting a popular content from the content store (CS) the CPCS needs to be updated in order to keep it consistent. All these are discussed in detail in the following subsections. But first, we discuss some augmentation in the CCN protocol that are necessary for the implementation of our proposal.



Content Name	Face	Hits
StreamX/SegY/%00%01/Layer1	f_0	3
---	---	---

Fig. 3 Structure of CPCS

5.1 Augmentation in CCN protocol

In this section, we present a few amendments that are needed in CCN protocol for the implementation of our proposed mechanism. The changes are consisted of adding an additional table in CCN, slight changes in the request forwarding mechanism and adding a column in PIT and CS. A few more modification regarding the cache decision are discussed in Section 5.2.

- **Cooperated Popular Content Store (CPCS):** Each CCN router (except the leaf nodes) have a special table (CPCS) that have information of the popular contents with its one hope child node. CPCS has a very simple structure that is consisted of three columns. First column is for the content name, second column is for the interface towards the child node that have the requested content in it's CS and third column is the hit counts. Structure of the CPCS is visualized in Fig. 3.
- **New columns in CS and PIT:** In this proposal we have added a new column CH to the CS for counting the cache hits. This value is used to measure local popularity of the content. Similarly for our proposed scheme, we need to add a column to the PIT that will hold a one bit value. We name this column as CM (Cache Marker). If the router is in the γ_k range of a chunk of video V_i^k, the Interest packet will put 1 in the CM field according to the procedure discussed in Section 5.2 otherwise the default value of this column is 0.
- **Interest Packet structure:** Interest packet is carrying two extra values and an Interest Type (IT) field. First value is γ_k which is the RTT range that is used in cache decision process. Second value is the cache capacity (C_T) of routers inside the γ_k range. Usage of both these values is discussed in Section 5.2. IT field is used to differentiate four different types of Interest packets. Usage of IT field is discussed in detail in Section 5.3. Optional bits in Interest header can be used, for carrying these values, as been discussed in the Interest packet header structure in the CCN reference implementation in [25].
- **Data chunk structure:** The Data chunk is carrying three additional values. First value is a one bit Cache Indicator (CI) value that is showing the caching status of the Chunk in current flow. Second value is C_R, which is the cache capacity of the remaining path inside the γ_k range. Usage of these values is discussed in Section 5.2. Third value is the Type of Data (TD) field which is used for giving a copy to the down node for a popular content. Usage of this value is discussed in detail in Section 5.3.

5.2 Cache decision process

In this subsection, we are discussing our proposed cache decision process. In the legacy CCN [8] the content is cached in each and every router on the path, which creates too many redundant copies of the content on the path. This mechanism of caching is called Leave Copy Everywhere (LCE). In situation when the cache of a router is full (which is almost always) some Data must be deleted from the CS to make space for the new Data. It is quite possible that a popular content is replaced by a non popular content. Many approaches have been proposed in the literature to solve this problem like [4, 27, 28] however, non of them is specialized for SVS. Our proposal insures to cache the content at most only one router in a call on the way and cache each layer in γ_k rage according to the layer k of the video been carried in the Data packet. Our aim is to keep the base layer nearer to the users and the enhancement layers in the upper router according to its hierarchy.

Algorithm 1 Parameters Setting via Interest Packet

```
if ($IT = 1$ ) [Hints: 1 for CPCS entry in the parent node] then
    Search CPCS
    if Entry for the chunk is already present then
        Ignore the Interest and discard it
    else
        Creat an entry for the chunk in CPCS
    end if
else if ($IT = 2$ ) [Hints: 2 for CPCS forwarded Interest] then
    Search CS for the requested chunk
    if Found then
        reply with the Data
    else {Hint: requested Data not found in CS. Here we need to update the parent
    CPCS as well as forward the Interest to the server}
        Set $IT = 3$ and return back the Interest on the receiving face
    end if
else if ($IT = 3$ ) [Hints: 3 for CPCS forward Interest is bounced back] then
    Search CPCS and delete the entry
    forward the Interest toward the server following FIB
else {Hints: for default IT which is "0"}
    if (Requested Data is present in $CS$) then
        Reply with the Data as discussed in algorithm 3
    else
        At Edge node only:
        Lower bound of $\gamma_k$: $\gamma_k^l = (k-1) * (\frac{RTT_N}{K})$
        Upper bound of $\gamma_k$: $\gamma_k^u = (k) * (\frac{RTT_N}{K})$
        At Every node including edge node:
        $RTT_i = 2(t_s^i - t_c^i)$
        if ($\gamma_k^l \leq RTT_i \leq \gamma_k^u$) then
            Mark $CM = 1$ in $PIT$
            $C_i = C_i + c_i$
            Forward the Interest
        else
            Forward the Interest
        end if
    end if
end if
```

5.2.1 *Copy down and CPCS entry*

We take the local popularity of the contents into account by recording the number of hits that it gain in CS. These hits are recorded by the Cache Hit (CH) field that is added to CS. When the number of cache hit reaches to the half of a threshold β i.e., ($CH = \lceil \beta/2 \rceil$), the router sends an update message to its parent router to inform it about the important Data. Line 5 to 8 in Algorithm 3 is showing this process. Upon reception of the message, the parent node creates an entry in the $CPCS$ and set the output face of this child node in the CPCS for the entry (Structure of CPCS is shown in Fig. 3). Moreover, when a node has to replace an old chunk with a new chunk, according to LRU policy, it check the CH field. If CH is more than or equal to $\lceil \beta/2 \rceil$, the node sends a copy of the deleting chunk to the parent node for keeping the parents CPCS updated. On reception of the CPCS update message (Data chunk) the parent node caches the chunk in its CS and deletes the packet and also delete the entry for the chunk in CPCS. The process of recording and updating the CPCS entry is discussed in Section 5.3 in detail.

Furthermore, if CH for any chunk reaches to threshold β, in a CCN node more than two hopes away from the user, the router marks that Data chunk which is cached by the router one hope above the access router on the way back. Similarly, when CH reaches to a threshold β_2 the chunk is cached by access router on the way. This process is discussed in detail in Section 5.2.3.

5.2.2 *Parameters setting via interest message*

Interest packet carries layer information of the requesting chunk in its name [12]. When an access CCN node, i.e., node that is connected with user(s), receives an Interest packet it extract the layer k information from the Interest name. On the basis of layer k the edge CCN node defines lower bound and upper bound of RTT threshold γ_k according to (1).

$$(k-1)*(\frac{RTT_N}{K}) \le \gamma_{\mathbf{k}} \le (k)*(\frac{RTT_N}{K}) \tag{1}$$

Where γ_k is the RTT range for video of layer k. K is the number of total layers in the video and k is the layer requested in current Interest packet. RTT_N is the RTT between the edge node and the content server or the content provider. Edge node keeps the most recent value of RTT_N with itself that it has calculated from the last Data chunk that had been delivered from the original content provider or the server. The lower bound and upper bound of γ_k are attached to the Interest message. When Interest message is received by a node, (node i), inside the γ_k range it compares RTT between the Edge node and the current node according to (2) (we call this RTT_i).

$$RTT_i = 2(t_s^i - t_c^i) \tag{2}$$

Where t_s^i is the timestamp attached to the Interest by the edge node as it is received, while t_c^i is the current time i.e., the time Interest is received by node i. If RTT_i is in γ_k range router set the Cache Marker (CM) value in PIT to 1. The Data chunk will be considered for caching in only those routers where the CM column's value in PIT is set to 1. Since, our cache decision that we discuss in Section 5.2.3, depends upon the cache capacity of the routers inside the γ_k range. Therefore, every router in γ_k range updates the total cache capacity C_T by adding its own cache capacity c_i to it. Total cache capacity from first node inside the γ_k to the last node in γ_k range is represented by the following equation.

$$C_T = \sum_{i=1}^{n} c_i$$

Where c_i is the cache capacity of current router inside the γ_k range and n is the number of routers inside the γ_k range. Parameter setting via Interest message is presented in Algorithm 1.

5.2.3 *Cache decision for data chunk*

There are possibly two cases how a Data can be acquired. First is that the Data is taken from the content provider or server and second case can be that the Data is taken from a CCN router's cache i.e., CS. Mechanism of cache hit and server hit is given in Algorithm 3. Here we discuss both these cases in the following subsections.

Algorithm 2 Action Upon Receiving Data Chunk

```
Check TD field
if (TD = 1 & current router is at the second last level) [Hints: Type 1 is for the copy down to one hope above access router] then
    Set "$TD = -1$", $CI = 0$"
    Forward the chunk
    Cache the Chunk in CS
else if TD = 2 & current router is at the last level i.e., access router) [Hints: Type 2 is for the copy down to the access router] then
    Forward the chunk
    Cache the Chunk in CS
else if ($TD = -1$ or $CI = 0$ or $CM = 0$ ) then
    Forward the Data Chunk
else {Hints: $TD = 0$, CI = 1 and CM in PIT is also = 1}
    Calculate Probability given in eq (3)
    if Probability = $'True'$ then
        Set $CI = 0$
        Forward the chunk
        Cache the Chunk in CS
    else {Hints: Probability is not true}
        Forward the chunk
        update $C_T$ according to eq (4)
    end if
end if
```

Case of server hit Interest message has marked all the routers in γ_k range by setting CM value to 1 in PIT, only one router out of these will cache the Data chunk. Upon receiving the Interest, server prepare Data packet in which it set the CI to 1 and TD to 0. Moreover, the content server, takes C_T (capacity of all the routers inside γ_k range) from the Interest and append it to the Data packet. In the Data packet we term this value as C_R (Cache capacity of all the remaining routers in γ_k range).

When a Data chunk is received by any router, it first check the TD bit in the chunk header. If the TD value is 0, the router will check the CI value. If it is 0, then the router will not consider it for caching. So the router will forward the chunk on the way back according to PIT entry. If the CI value is 1 then the router checks the CM value in the PIT, If it is 0 then it means this router is not in γ_k range. In this case, the router just forwards the packet according to PIT entry on its way back. If this CM value is also 1 in the PIT table, then it means this router is in γ_k range and should consider the chunk for caching.

In cache decision our aim is to cache the Data chunk in router according to its cache capacity inside the γ_k range and at the same time we also want to give weightage to the routers according to its cache size. To get these preferences, our cache decision take place in the marked routers according to formula given in (3).

$$P_i\left(V_k\right) = \frac{c_i}{C_R} \tag{3}$$

The above formula gives probability P for caching the Data chunk V_i^k at current router. Here C_R is the cache capacity of the remaining routers on the path inside the γ_k range i.e.,

$$C_R = C_T - \sum_{j=1}^{J} c_j$$

Here J represent number of all the routers inside γ_k range that are traversed by the Data. C_R is updated by every router inside the γ_k range according to the following formula.

$$C_R = C_R - c_i \tag{4}$$

If probability in (3) becomes true at any node, it caches the content in CS and changes the (CI) value from 1 to 0. The remaining router will not calculate the probability, when they see $CI = 0$. Thus, with this CI flag, we reduce complexity of cache decision because the calculation is done only by a subset of routers in γ_k range. Formula in (3) always give probability of 1 at the edge router which ensures to cache the content if it is not cached by any other node in γ_k range. The process of caching the Data chunk in γ_k range is shown in line 11 to 19 of Algorithm 2.

Algorithm 3 Action Upon Server/Cache Hit

```
if (Server hit or cache hit outside $\gamma_k$ range) then
    Set CI = 1
    Append $C_T$ (Taken from Interest message) to the Data chunk
    Reply with the chunk
else if ($CH = \beta/2$) [Hints: CPCS entry message to the parent] then
    Set CI = 0
    Reply with the chunk to the requested node
    Send Interest with $IT = 1$ to the parent node
else if ($HC >= \beta$)[Hints: Copy Down to node one level above to the access router] then
    Set TD = 1
    Reply with the chunk
else if ($HC >= \beta_2$)[Hints: Copy Down to the access router] then
    Set TD = 2
    Reply with the chunk
else
    Reply with the chunk
end if
```

Case of cache hit If there is a cache hit inside the network for a chunk there can be two situations. First is that the router at which the cache hit occurred is outside the γ_k range. This situation is straightforward, the hit node is considered as server. The router will reply with the Data chunk and will set the parameters as discussed above. There can be many causes of cache hit outside the γ_k range, e.g., network condition is changed over time or the request is arrived from a different place than the previous one etc.

Second case is that the node at which the cache hit occurred is inside the γ_k range. In this case the router at which the cache hit occurred will increment the CH value in CS by 1 and will check the CH value against the thresholds β/β_2. When the number of cache hit reaches to the half of a threshold β i.e., $(CH = \lceil \beta/2 \rceil)$, the router sends an Interest message, setting IT equal to 1 to the parent node for recording an entry in the CPCS. The process of CPCS entry making and updating is discussed in Section 5.2.1. In case CH is less than β and not equal to $\beta/2$, The router will reply with the Data chunk without changing the default values of header fields. If CH is equal to β, The router will reply with the Data chunk by setting TD field in chunk header to 1 and in case, the CH is equal to β_2 the router sets TD field to 2 and reply with the Data chunk.

The down node when receives the Data chunk, first it checks the TD in the Data header. If the TD bit is carrying 1, the router will see, if it is the second last level router, it will cache the Data and will change the TD to -1, CI to 0 and will forward it to the node down. In case, the TD value is 2, it means the Data is for copy down to the access router. The router will see, if it is access router, will cache the Data and will forward it to the requested user(s). Line 1 to 8 in Algorithm 2 is showing this process.

5.3 Request forwarding and CPCS update

The new table added, which we call Cooperated Popular Content Store (CPCS), is used similarly as Content Store (CS). However, it don't have any data but only the forwarding interface towards the child node that has the requested Data. As been mentioned in Section 5.1, CPCS has 3 columns that holds content name, interface for reaching the contents and hit counts.

In our proposed scheme we introduce a slight modification in the Interest forwarding mechanism of CCN. The Interest message is of 4 types. Each Interest is identified by the Interest Type (IT) header value. If the IT value is 0 then this interest is forwarded normally with the method as discussed in Section 5.2 and shown in line 21 to 31 of Algorithm 1. IT value 1 is an update message from child node to its parent node to inform it about a popular content. This message is sent by the child node in situation when the CH i.e., number of cache-hits for a content reaches to the β limit. Upon receiving this update message an entry is created in CPCS of parent node, it also set the forward interface to this child node and set CH to 0. When the CH for any CPCS entry reaches to $\lceil \beta/2 \rceil$ in a parent CCN node, it caches the content in its own CS and deletes the CPCS entry for that content. Interest of type 2 is the Interest forwarded by the parent node towards the child node due to CPCS hit. If the requested Data chunk is not present in the CS for type 2 Interest, the child node change IT to 3 and sends back the Interest on the receiving face to the parent node. Upon reception of the type 3 interest, the parent node deletes the corresponding entry from $CPCS$, changes IT to 0 and sends the Interest towards the server following FIB. Forwarding Interest of type $1, 2$ and 3 is presented in line 1 to 20 of Algorithm 1.

Here is an important thing to be noted that, in case of CPCS hit, the node where CPCS hit is occurred will make PIT entry before forwarding the Interest towards the child node (that have the content) just like normal Interest forwarding case. The child node which have the actual content will be considered as the cache hit node. In case CPCS is full and Interest of type 1 is received, an old content is deleted to make space for the new one. Selection of the content to be deleted is done according to Least Recently Used (LRU) method.

5.4 Cache eviction

CCN router has a limited cache size which almost always remain full. In order to cache a new content the router must delete some of the existing one. The ideal scenario is to delete the least important content among the cached content to make space for the new one. One of the simplest and widely used method is the Least Recently Used (LRU). Here in our proposed system we use the same LRU, which is also used in the original CCN [8]. However, in our proposal, if new content has to replace a popular content (content whose CH is greater than or equal to $\lceil \beta/2 \rceil$) according to LRU, the CCN node must inform its parent node about this, so that the CPCS in parent node remain consistent. Before replacing the deleting chunk is sent to the parent node with TD field value is fixed with "2". The parent node treats this Data as been discussed above.

6 Results and discussion

In this section we present the performance evaluation of our proposed mechanism to show the relative performance analysis of our scheme as compare to other similar proposals.

6.1 Simulation and analytical study

To measuring the performance of our proposed schemes, we are focusing on three aspects of the caching and forwarding schemes i.e., cache hit ratio, hit distance and download delay.

6.1.1 Cache hit ratio

When a copy of the requested Data is found in the cache of a CCN router, we call it cache hit. This is one of the most important parameter to measure performance of the caching scheme. Most of the time, operators have to pay for transiting the data to a third party network. High cache hit means lower server hits or less data is transited outside the network and vice versa. Cache hit is measured with the following formula:

$$CacheHit = \frac{CacheHit}{CacheHit + CacheMiss} \tag{5}$$

There are a number of factors that affect cache hit ratio.

Caching scheme If the caching scheme keeps the most important (frequently requested data) in the cache, it will increase cache hit ratio.

Interest forwarding scheme Intelligent Interest forwarding plays a vital role in cache hits. Higher cache hitting can be achieved, if the forwarding scheme forwards the Interest towards a node, where the requested data can be found. Flooding the Interest can achieve higher cache hit. However, flooding produces a large amount of unnecessary redundant traffic in the network which may degrade the performance. Our intelligent Interest forwarding for popular data, through CPCS, achieves improved performance which can be seen in the simulation results in the following sections.

Cache size Bigger size cache can give high cache hit rate. However, cache cannot be of unlimited size because it is expensive and secondly caching and forwarding decision are needed to be taken on the line speed. Therefore, cache size should be according to the processing capability of the router.

6.1.2 Hit distance

Hit distance is the distance (most of the times measured in terms of number of hops) traveled by Interest to get a copy of the requested Data. Hit distance give an idea of the traffic flow inside the network. Shorter hit distance means the Interest and corresponding Data packets have to travel shorter inside the network and vice versa. Hit distance performance can be improved by keeping important data in the nearest location to the users. Our proposed caching and forwarding schemes achieve this by exploiting important parts inside a video contents.

6.1.3 Download delay

Download delay is the most important parameter from the users' point of view. Lower download delay yields higher QoE and vice versa. Hit distance is one of the factors affecting download delay. However, complexity of caching and forwarding decisions also effect download delay. Moreover, link capacity may also have a role in the download delay. Load on a router is another important factor that can effect the download delay. Higher load will increase the queuing delay which will result in higher download delay.

6.2 Simulation setup

We have implemented our proposed cache management and request forwarding mechanism by modifying chunk-level simulator, ccnsim [30] that is developed over Omnet++ simulator. Source code of the simulator is available at [36]. All the links in the network have the same capacity and RTT remains the same during the simulations. Table 2 shows the parameter used in the experiments. We are considering tree topology for our simulations. The tree constructed is a level 4 binary tree i.e., there are total 15 nodes. Server is connected with the root of the tree while clients are connected with the leaf nodes. Clients generate requests (Interest packets) in a random manner governed by zipf distribution with different parameter 0.8, 1, 1.2, 1.4. There are total of 10 thousand videos with the server. Each video is consisted of 4 layers i.e., 1 base layer and 3 enhancement layers. Users request either only BL or BL and 1 EL or BL and 2 ELs or BL and 3 ELs on random basis. There are total of 20 chunks in one content i.e., 5 chunks for each layer. Each chunk represents one segment of the video.

We have compared our proposal with the following caching and forwarding schemes.

Caching schemes The following four prominent caching schemes are used to show the performance analysis of our proposed scheme in the simulations.

Leave copy everywhere This is the universal caching approach proposed in [8]. According to this scheme, every router caches every Data chunk that passes through it. This scheme

Table 2 Parameters used in the experiments

Parameters	Values
num; repositories	1
replicas	1
num; routers	15
num; user groups	8
λ	1
max no; of layers in a video	4 (1 BL, 3 ELs)
α	$0.8 \sim 1.4$
β	2
β_2	3
Total Videos	10^4
Chunks per video (BL+ELs)	20
CDS	LCE, LCD, PROB
Replacement policy	LRU
Forwarding schemes	fixed, NRR1, SPR, Proposed

caches redundant copies of the Data in the network due to which the cache utilization may be effected as the one-timer Data chunk may replace the popular one.

Leave copy down This scheme was proposed in [13]. In this scheme, whenever cache hit occurs in a router, it gives a copy of the Data to the one hop down node.

ProbCache ProbCache [28] caches Data in the routers with probability that is dependent on the cache size and the path length. Routers nearer to the users have high probability to cache the chunk. This scheme do not guarantee the elimination of redundant content caching.

Probabilistic algorithm This scheme is also called as FixedP. According to this scheme every router caches a chunk with the pre-defined fixed probability.

DASCache DASCache [21] takes caching decision for the content on a path periodically on the basis of contents' popularity (Zipf distribution). The caching decision is formulated as binary integer programing based optimization problem which is solved centrally for each path periodically and the contents are placed at the optimum location in the network.

Forwarding schemes We compare our proposed scheme with the following Interest forwarding schemes.

Shortest path routing According to this strategy, router chooses the shortest path repository and sends packets on the corresponding face.

Nearest replica routing (NRR1) In this scheme, router sets up an exploration phase, in which the node floods the neighborhood with the request for the given object. the node that have the requested data in it's cache reply and thus the Data is acquired. This scheme generates extra traffic in the network which may degrade the performance in the rush hours. for simplicity we term this scheme as NRR in the following sections.

6.3 Simulation results

In this section we discuss results of our simulation.

6.3.1 Cache hit performance

Firstly, we take the cache hit (the requested content is found in a CCN router's cache) parameter to compare the performance of our proposed mechanism with the existing similar proposals. Figure 4 is showing cache hit ratio. Here we can see that hit rate for our proposed scheme is higher than LCD, LCE and ProbCache in both NNR1 (for simplicity we call it NRR) and SPR routing mechanisms. All of these three caching schemes suffers from caching redundant copies of data in routers which reduces cache space utilization. SPR is the worst performing forwarding scheme because of its inability to take the Data from the routers that are not on the shortest path to the server. NRR performance is relatively better. However, it floods the Interest, which produces too much useless traffic inside the network. Cache hit rate for DASCache is also lower. There are two reasons for it. Firstly, DASCache follows SPR for Interest forwarding which only search the router for data that fall on the shortest path to the server. Secondly, DASCache solves optimization problem for

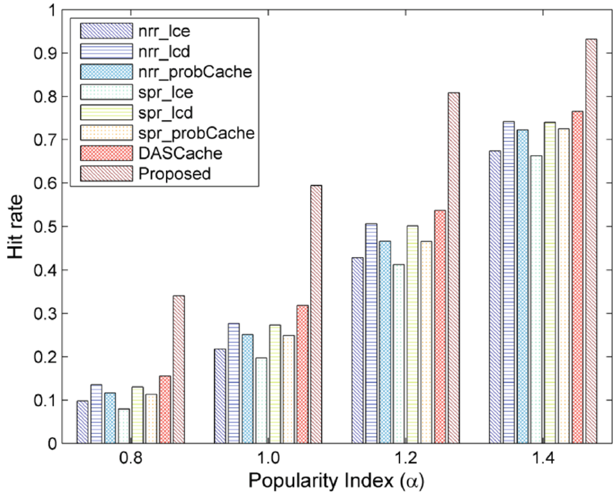


Fig. 4 Comparative analysis of cache hit rate

each path, in which the most popular contents (on the basis of Zipf distribution) are cache in the nearest router to the users. Eventually all the nodes of the same level (in case of tree topology) will have exactly the same contents. This is because, Zipf considers global popularity for the content which remains the same, for a given content, everywhere in the network. Therefore, we can say that DASCache is poor performer regarding redundant data reduction in the cache of nearby nodes. In contrast, due to CPCS, our proposed scheme caches distinct popular contents in the nearby routers (especially same level router in tree topology). Moreover, instead of global popularity of the content, our proposed scheme takes cache decision on the basis of local popularity of the contents (the number of cache hits). Hence, our proposed scheme outperforms all other schemes.

6.3.2 Hit distance performance

Hit distance shows the number of hops that the Interest/Data travels. Hit distance reflects traffic inside the network. Figure 5 is showing average hit distance. Due to lower cache hit rate, all of the compared schemes have to download the requested Data from the server which results in longer hit distance. Our proposed intelligent caching and forwarding schemes outperforms all the compared schemes. CPCS have a vital role in performance improvement. ProbCache and DASCache suffer with redundant contents caching and inability of exploiting cache of the nearby routers. Interest must have to pick up the requested data from server although a copy of the same data may be present with the nearby node. This is because, DASCache follows SPR for Interest forwarding.

6.3.3 Download delay performance

Download delay measures the time for getting the requested contents. By average delay we mean the time taken, on average, from issuing Interest message until getting the requested Data. Figure 6 is showing the average delay for the download of requested content. Y-axis of the figure is showing the average download delay per-chunk in seconds. In the figure,

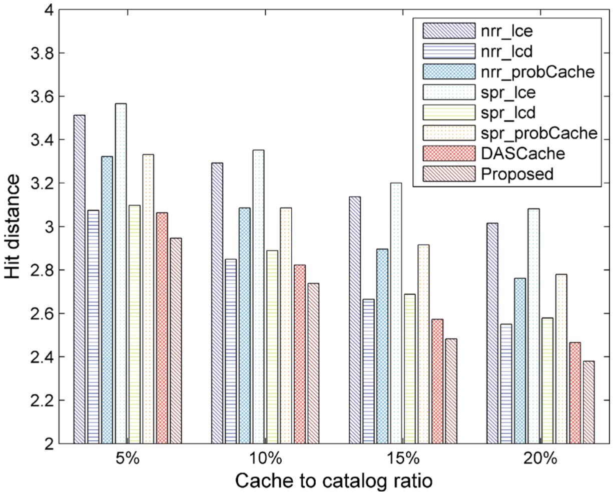


Fig. 5 Comparative analysis of hit distance

we see that our proposed scheme has the lowest delay in all the compared scenarios. This is due to caching the important, popular and distinct contents in the nearer by routers to the users. Moreover, our forwarding scheme exploits cache of the child nodes with the help of CPCS of the parent nodes. The effect of lower cache hit, longer hit distance and complexity of the caching and forwarding schemes affects the download delay of all the other compared schemes. DASCache delay performance is relatively very bad. The possible reason for this is the complexity of cache decision process. DASCache performance will be the worst in larger networks due to the delay caused by solving the cache decision optimization problem.

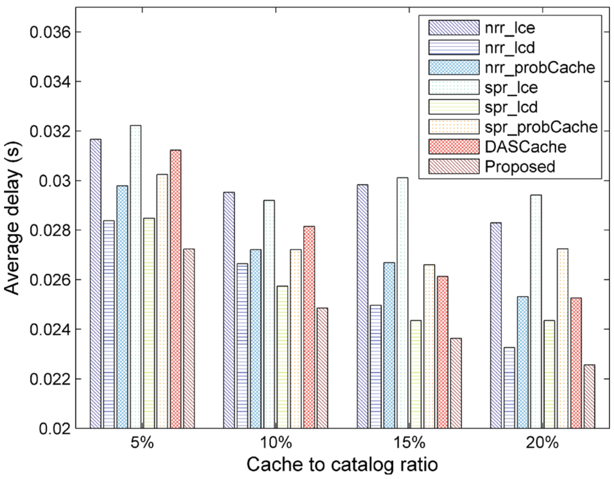


Fig. 6 Comparative analysis of download delay

Table 3 Summary of comparative analysis

	nrr_lce	nrr_lcd	nrr_prob-Cache	spr_lce	spr_lcd	spr_prob- Cache	DASCache	Proposed
Hit rate	0.218	0.277	0.251	0.197	0.273	0.249	0.318	0.594
Hit distance	3.292	2.849	3.086	3.352	2.910	3.085	2.823	2.738
Delay(s)	0.0295	0.0266	0.0272	0.0292	0.0257	0.0272	0.0282	0.0249

6.3.4 Summary of comparative analysis

We presented comprehensive comparative analysis of different caching and forwarding schemes. Table 3 is showing summary of our comparative analysis. Results shown in the table are for $\alpha = 1$ and cumulative cache size of the network is 10 %.

6.3.5 Analysis of different video layers

In this section, we discuss analysis of different layers of the video. As we mentioned earlier, we have considered video that are consisted of 4 layers, i.e., one base layer and 3 enhancement layers. The users randomly select to request a video either of lowest quality i.e., only base layer, base layer with 1 enhancement layer, base layer and 2 enhancement layers or base layer and 3 enhancement layers. In the following subsections we discuss the layered video analysis from different perspectives.

Download delay for different video layers Figure 7 is showing the average download delay of our proposed scheme for different layers of the video. Y-axis of the figure is showing the download delay per-chunk in seconds. We cache BL of the video at the nearest location to the users. Therefore, we can see in the figure that download delay for

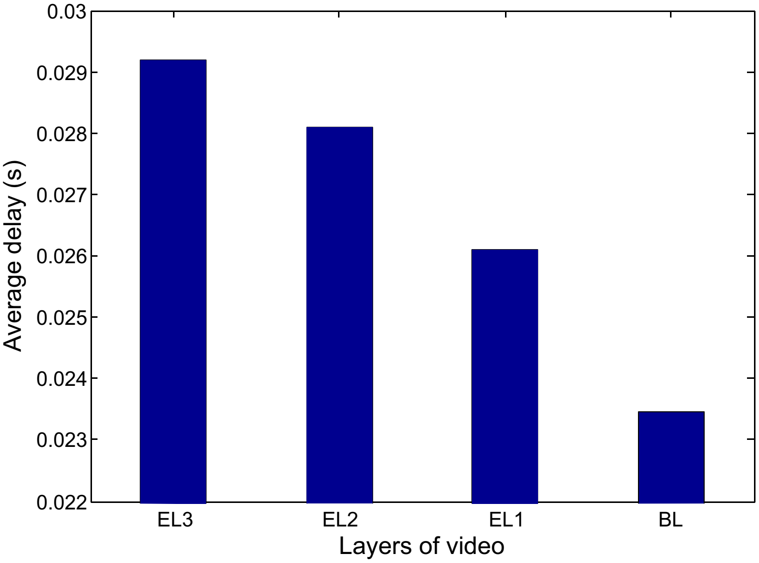


Fig. 7 Average download delay of the proposed scheme for different layers of the videos

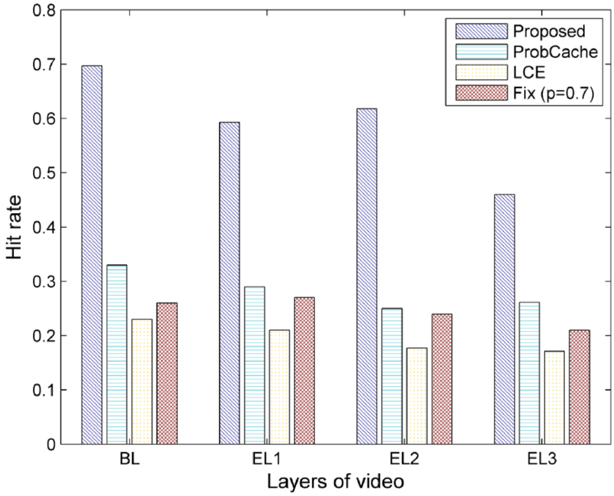


Fig. 8 Comparative analysis of hit rate for different layers of the videos

the BL is the lowest. Similarly, the first enhancement layer's (EL1) delay is more than the BL and upper enhancement layers' delay is in increasing order. Achievements of our proposed scheme is exactly according to our motivation, presented in Section 3.3, that BL is needed by all the users regardless of their Internet connectivity and device specification and demand for the upper layers is in decreasing order. On average, BL takes 21 % lesser time in downloading than $EL3$ in the simulated scenario.

Hit rate for different video layers Figure 8 is showing comparison of the average hit rate for different layers of the videos. In the figure, we can see that our proposed scheme outperforms all the compared schemes for the first 3 layers of the videos. For the 4th layer

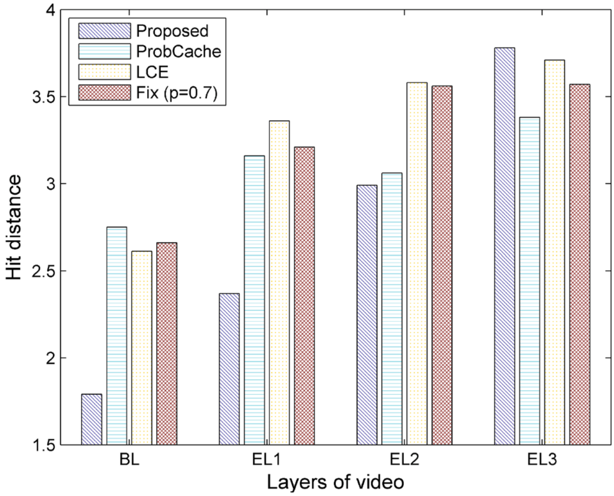


Fig. 9 Comparative analysis of hit distance for different layers of the videos

our scheme showing the similar behavior as the other schemes. Reason of the phenomenon is that in our proposed scheme, the last layer is almost always stored in the root node and we did not implement CPCS for the root node. Purpose of not defining CPCS in the root node is that both the server and the CPCS would been at one hop distance from the root node. So instead of taking Data from the child node via CPCS, it is better to request it from the sever as both are one hope distance. In this way we can reduce burden of maintaining CPCS in the root node. In the scenario where the network service provider pay to the content provider then maintaining a CPCS with the root node will be beneficial.

Hit distance for different video layers Figure 9 is showing hit distance for different video layers. Base layer is the mandatory layer and majority of the users request it as we have discussed in Section 3 in detail. our proposed scheme shows significant less hit distance for the base layer and enhancement layer 1 as compared to the other schemes. Hit distance of our proposed scheme for $EL2$ and $EL3$ is not much different from the other schemes. This is because we cache these layers away from the users. Since a subset of users is requesting these layers so overall impact of this is very little.

Comparative analysis of download delay for different video layers Figure 10 is showing comparative analysis of average download delay for the proposed scheme with other proposals for different layers of the videos. Y-axis of the figure is showing the download delay per-chunk in seconds. Here, we can see that our proposed scheme is having less delay for downloading the requested base layer and enhancement layer 1 as compare to all other schemes been analyzed in these experiments. Download delay of our proposed scheme for BL and $EL1$ is significantly lower than all the other compared schemes while, download delay for $EL2$ and $EL3$ is relatively the same as of the other schemes. Since the upper ELs are needed by a subset of users, therefore, its impact is very minor.

6.3.6 Summary of layers' analysis

We presented analysis for different layer of the video by different caching and forwarding schemes. Table 4 is representing summer of different layers of the videos.

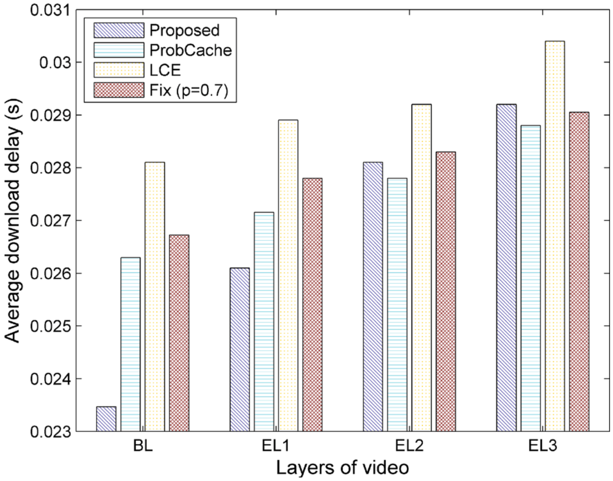


Fig. 10 Comparative analysis of average download delay for different layers of the videos

Table 4 Summary of layers' analysis

	BL			EL1			El2			EL3		
	Hit rate	Hit distance	Delay	Hit rate	Hit distance	Delay	Hit rate	Hit distance	Delay	Hit rate	Hit distance	Delay
Proposed	0.70	1.79	0.0235	0.59	2.37	0.0261	0.62	2.99	0.0281	0.46	3.78	0.0292
ProbCache	0.33	2.75	0.0263	0.29	3.16	0.0272	0.25	3.06	0.0278	0.26	3.38	0.0288
LCE	0.23	2.61	0.0281	0.21	3.36	0.0289	0.18	3.58	0.0292	0.17	3.71	0.0304
Fix(p=0.7)	0.26	2.66	0.0266	0.27	3.21	0.0278	0.24	3.56	0.0283	0.21	3.57	0.0291

7 Conclusion

In SVC-based video streaming the mandatory base layer is required by every user who want to watch the video while demand for the enhancement layers is decreasing as we go up in the hierarchy. Demand for the enhancement layers depend upon users' device capability/budget or network condition. This phenomenon encouraged us to present a caching scheme in CCN that cache the content according to its layer information. In this paper we presented a cache scheme that cache the base layer nearer to the users and enhancement layers away from the users according to its hierarchy. Interest packet make a group of routers on the path by marking it that is considered for caching the corresponding Data chunk. The group is consisted of all the router in a specific RTT range. RTT range is selected by edge router on the basis of layer of video requested in the Interest. Data chunk upon reaching the marked routers are cached probabilistically on the basis of current router's cache capacity and cache capacity of the remaining routers in the group. Furthermore, our proposed scheme brings the popular content nearer to the users by copying it down to the routers in two hops range from the users. Thus, our caching strategy bring the video layer requested by most of the users nearer to the users and less requested layers away from the users according to its hierarchy.

We also propose a request forwarding scheme in which parent node have a limited knowledge of popular contents in cache of its one hop children. Child node give information of popular contents that are cached in its CS to its parent node. Parent node stores this in a table we call it CPCS. Instead of forwarding the request towards the server the parent node directs Interest for the popular contents to its child node if it is present there.

We simulated the proposed scheme by extending chunk level simulator, ccnsim and compare it with other similar proposals. Our simulation results show high cache hit in the network, shorter hit distance and lower download delay. Also our proposal provide the mandatory base layer quickly to the users. By caching the most demanding part of video nearer to the users, our proposed solution decrease the traffic flow inside the network by providing the requested contents from nearby location. For our future work, we aim to implement our idea in the real environment and evaluate the proposal with other parameters like received video quality etc.

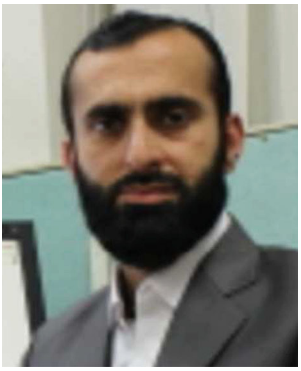
Acknowledgments This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (R0126-16-1009, Development of Smart Mediator for Mashup Service and Information Sharing among ICBMS Platform). CS Hong is the corresponding author.

References

1. Ahlgren B, Dannewitz C, Imbrenda C, Kutscher D, Ohlman B (2012) A survey of information-centric networking. IEEE Commun Mag 50(7):26–36
2. Amadeo M, Campolo C, Molinaro A (2016) Information-centric networking for connected vehicles: a survey and future perspectives. IEEE Commun Mag 54(2):98–104

3. Carofiglio G, Gallo M, Muscariello L, Perino D (2011) Modeling data transfer in content-centric networking. In: Proceedings of the 23rd International Teletraffic Congress. International Teletraffic Congress, pp 111–118
4. Chai WK, He D, Psaras I, Pavlou G (2012) Cache "less for more" in information-centric networks. In: NETWORKING 2012. Springer, pp 27–40
5. Cisco 2019 prediction: http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index vni/vni-forecast.html
6. Guo S, Xie H, Shi G (2012) Collaborative forwarding and caching in content centric networks. In: NETWORKING 2012. Springer, pp 41–55
7. Han L, Kang S-S, Kim H, In HP (2013) Adaptive retransmission scheme for video streaming over content-centric wireless networks. IEEE Commun Lett 17(6):1292–1295
8. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL (2009) Networking named content. In: Proceedings of the 5th international conference on emerging networking experiments and technologies. ACM, pp 1–12
9. Jiang D, Xu Z, Lv Z (2015) A multicast delivery approach with minimum energy consumption for wireless multi-hop networks. Telecommun Syst:1–12
10. Jiang D, Shi L, Zhang P, Ge X (2016) Qos constraints-based energy-efficient model in cloud computing networks for multimedia clinical issues. Multimedia Tools Appl 1–22
11. Jiang D, Yuan Z, Zhang P, Miao L, Zhu T (2016) A traffic anomaly detection approach in communication networks for applications of multimedia medical devices. Multimedia Tools Appl 1–25
12. Kulinski D, Burke J (2012) Ndnvideo: Random-access live and prerecorded streaming using ndn, Technical Report NDN, Tech. Rep.
13. Laoutaris N, Syntila S, Stavrakakis I (2004) Meta algorithms for hierarchical web caches. In: IEEE international conference on performance, computing, and communications, 2004. IEEE, pp 445–452
14. Laoutaris N, Che H, Stavrakakis I (2006) The lcd interconnection of lru caches and its analysis. Perform Eval 63(7):609–634
15. Lederer S, Mueller C, Rainer B, Timmerer C, Hellwagner H (2013) An experimental analysis of dynamic adaptive streaming over http in content centric networks. In: IEEE international conference on multimedia and expo (ICME), 2013. IEEE, pp 1–6
16. Lederer S, Mueller C, Rainer B, Timmerer C, Hellwagner H (2013) Adaptive streaming over content centric networks in mobile networks using multiple links. In: IEEE international conference on communications workshops (ICC), 2013. IEEE, pp 677–681
17. Lee J, Hwang J, Choi N, Yoo C (2013) Svc-based adaptive video streaming over content-centric networking. KSII Trans Internet Inf Syst (TIIS) 7(10):2430–2447
18. Li W (2015) Popularity-driven caching strategy for dynamic adaptive streaming over information-centric networks
19. Li Z, Simon G (2011) Time-shifted tv in content centric networks: the case for cooperative in-network caching. In: 2011 IEEE international conference on communications (ICC), pp 1–6
20. Li Z, Simon G (2014) Cooperative caching in a content centric network for video stream delivery. J Netw Syst Manag 1–29
21. Li W, Oteafy SM, Hassanein HS (2015) Dynamic adaptive streaming over popularity-driven caching in information-centric networks. In: 2015 IEEE international conference on communications (ICC). IEEE, pp 5747–5752
22. Marpe D, Wiegand T, Sullivan GJ (2006) The h. 264/mpeg4 advanced video coding standard and its applications. IEEE Commun Mag 44(8):134–143
23. Muscariello L, Carofiglio G, Gallo M (2011) Bandwidth and storage sharing performance in information centric networking. In: Proceedings of the ACM SIGCOMM workshop on information-centric networking. ACM, pp 26–31
24. Petrangeli S, Bouten N, Claeys M, De Turck F (2015) Towards svc-based adaptive streaming in information centric networks. IEEE
25. Project ccnx: http://www.ccnx.org/
26. Psaras I, Clegg RG, Landa R, Chai WK, Pavlou G (2011) Modelling and evaluation of ccn-caching trees, pp 78–91

27. Psaras I, Chai WK, Pavlou G (2012) Probabilistic in-network caching for information-centric networks. In: Proceedings of the second edition of the ICN workshop on Information-centric networking. ACM, pp 55–60
28. Psaras I, Chai W, Pavlou G (2013) In-network cache management and resource allocation for information-centric networks. IEEE Trans Parallel Distrib Syst
29. Ponnusamy SP, Kathikeyan E (2012) Hp proxy: Hot-point proxy caching with multivariate sectoring for multimedia streaming. Eur J Sci Res 68(1):21–35
30. Rossini G, Rossi D (2013) ccnsim: a highly scalable ccn simulator. In: IEEE ICC
31. Saino L, Psaras I, Pavlou G (2013) Hash-routing schemes for information centric networking. In: Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking. ACM, pp 27–32
32. Schwarz H, Wien M (2008) The scalable video coding extension of the h. 264/avc standard. IEEE Signal Process Mag 25(2):135
33. Schwarz H, Marpe D, Wiegand T (2007) Overview of the scalable video coding extension of the h.264/avc standard. IEEE Trans Circuits Syst Video Technol 17(9):1103–1120
34. Sodagar I (2011) The mpeg-dash standard for multimedia streaming over the internet. IEEE MultiMedia 4:62–67
35. Sourlas V, Gkatzikis L, Flegkas P, Tassiulas L (2013) Distributed cache management in information-centric networks. IEEE Trans Netw Serv Manag
36. Source cod for ccnsim: http://www.infres.enst.fr/drossi/index.php?n=software.ccnsim
37. Suksomboon K, Tarnoi S, Ji Y, Koibuchi M, Fukuda K, Abe S, Motonori N, Aoki M, Urushidani S, Yamada S (2013) Popcache: Cache more or less based on content popularity for information-centric networking. In: IEEE 38th conference on local computer networks (LCN), 2013. IEEE, pp 236–243
38. Wang JM, Bensaou B (2012) Progressive caching in ccn, pp 2727–2732
39. Wang JM, Zhang J, Bensaou B (2013) Intra-as cooperative caching for content-centric networks. In: 3rd ACM SIGCOMM workshop on Information-centric networking, pp 61–66
40. Wiegand T, Sullivan GJ, Bjøntegaard G, Luthra A (2003) Overview of the h. 264/avc video coding standard. IEEE Trans Circuits Syst Video Technol 13(7):560–576
41. Yi C, Afanasyev A, Wang L, Zhang B, Zhang L (2012) Adaptive forwarding in named data networking. ACM SIGCOMM Comput Commun Rev 42(3):62–67



Saeed Ullah received his B.S. degree in Information Technology in 2006 from IBMS/CS, Peshawar and M.S. degrees in Information Technology from National University of Sciences and Technology (NUST), Islamabad, in 2010. He is current working for his PhD degree in Department of Computer Science and Engineering, Kyung Hee University, South Korea. His research interest includes, Multimedia Communication, Scalable Video Streaming Routing and In-network Caching, and Future Internet.

Kyi Thar received Bachelor of Computer Technology in 2007 from University of Computer Studies, Yangon, Myanmar. He is currently working towards the Ph.D. degree at Department of Computer Science and Engineering, Kyung Hee University, Rep. of Korea. His research interest includes Information Centric Networking, Routing and In-network Caching

Choong Seon Hong (AM'95–M'07–SM'11) received the B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, South Korea, and the Ph.D. degree from Keio University, Tokyo, Japan, in 1983, 1985, and 1997, respectively. In 1988, he joined KT, where he worked on broadband networks as a Member of the Technical Staff. In September 1993, he joined Keio University. He had worked for the Telecommunications Network Laboratory, KT as a Senior Member of Technical Staff and as a Director of the Networking Research Team until August 1999. Since September 1999, he has been a Professor with the Department of Computer Engineering, Kyung Hee University. His research interests include future Internet, ad hoc networks, network management, and network security. He has served as a General Chair, TPC Chair/Member, or an Organizing Committee Member for International conferences such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN, TINA, SAINT, and ICOIN. Also, he is now an Associate Editor of the IEEE TRANSACTIONS ON SERVICES AND NETWORKS MANAGEMENT, International Journal of Network Management, Journal of Communications and Networks, and an Associate Technical Editor of the IEEE Communications Magazine. He is a member of ACM, IEICE, IPSJ, KIISE, KICS, KIPS, and OSIA.