

# Reducing channel zapping time in live TV broadcasting over content centric networks

Behrouz Shahgholi Ghahfarokhi<sup>1</sup> · Neda Moghim<sup>1</sup> ·  
Shayan Eftekhari<sup>1</sup>

Received: 14 March 2016 / Revised: 8 August 2016 / Accepted: 4 October 2016 /  
Published online: 16 November 2016  
© Springer Science+Business Media New York 2016

**Abstract** Channel zapping delay is a big challenge in delivering TV service over the Internet infrastructure. Previous research works have studied this delay, its components, and solutions to decrease it. Unfortunately, the best proposed solutions reduce the delay at the expense of increasing bandwidth usage or decreasing the received video quality. After channel switching, the Set Top Box (STB) or player application should buffer sufficient frames before starting to play the received video. However, the buffering process takes place at the playback rate and leads to a delay which is inversely related to the buffer duration. Regarding Information Centric Networking (ICN) paradigm, this paper introduces a new channel zapping protocol that aims to remove the synchronization and buffering delays while maintaining the bandwidth utilization and also the received video quality. The general idea of the proposed solution is to exploit the in-network caching feature of the ICN to retrieve the frames from the network at the network speed. Although the analyses show that the proposed zapping protocol eliminates the delay dependency to the buffer duration, network throughput becomes the bottleneck instead. So, novel solutions have been proposed to reduce the queuing delay as the main component of network delay. These solutions include two new caching algorithms, a new cache replacement algorithm, and applying scheduling methods to the forwarding queues. Simulation results show that increasing link rates, using the proposed caching and cache replacement algorithms, and applying an appropriate scheduling method will greatly reduce the zapping delay without sacrificing the bandwidth or video quality.

---

✉ Behrouz Shahgholi Ghahfarokhi  
shahgholi@eng.ui.ac.ir

Neda Moghim  
n.moghim@eng.ui.ac.ir

Shayan Eftekhari  
eftekhari@eng.ui.ac.ir

<sup>1</sup> Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran

**Keywords** Information centric networking · TV broadcast · Channel zapping · In-network caching · Video quality

## 1 Introduction

With the advent of video hosting services like YouTube and IPTV networks such as Netflix and Hulu, it is anticipated that 80 to 90 % of Internet traffic will be devoted to video by the end of 2018 [5]. However, distribution of television content on current IP-based networks encounters many constraints and challenges. Weakness of IP protocol in guaranteeing the required QoS level for multimedia services is one of the basic limitations in this regards. On the other hand, considering the Quality of user's Experience (QoE) is a vital criterion in multimedia and television service provisioning, which is directly affected by the network performance parameters.

Execution delay of the users' commands, such as channel zapping request (that is addressed in this paper), is one of the basic factors that has a negative impact on the quality of experience and users' satisfaction level. In traditional broadcast TV, the content of all channels are broadcasted to the covered receivers. Due to bandwidth limitation, it's not possible to broadcast all channels in an IP-based network. Instead, multicast communications are used to deliver a channel to the interested customers. Therefore, channel zapping includes leaving the current multicast group and joining the new one. These two phases can lead to an approximately 100–200 ms of delay [7, 17, 42–44] which makes channel zapping time remarkable.

An MPEG video usually contains three types of frames: Intra Frame (I-frame), Predicted Frame (P-frame), and Bidirectional Frame (B-frame). I-frames contain the data of the baseline image while P-frames and B-frames contain the temporal differential data. The number of frames between two I-frames is called Group of Pictures (GoP). Image reconstruction requires receiving an I-frame. Thus, the receiver should wait for a while, which is called synchronization time, to capture the first I-frame of the video. Furthermore, receiver needs to buffer the captured video frames, before starting playback, to overcome the packet loss and jitter. In average, the mentioned synchronization and buffering times increase the channel zapping time for 1 to 2 s [17, 43, 44]. Finally, processing delay or decoding delay, that is not usually greater than 50 ms, will be added to this total delay. It is worth noting that the maximum acceptable channel zapping time is around 430 ms [24].

Many research works have been done to improve the users' QoE in video streaming and to reduce the interaction delay with TV. The most serious ones try to reach this end for the price of increasing the bandwidth usage [8, 10, 22, 26, 28, 30, 31, 37, 40–42, 46]. Solving this problem is somewhat difficult owing to the current Internet's host-to-host architecture where hosts are the main entities of the network and data packets do not have identity without them. Therefore, it is not feasible to make a profit of storing the packets in network for future use. With the rapid increase of the multimedia traffic, Internet users' expectations are changing and they care about the content irrespective of its location. Such shortcomings have led to the appearance of Information-Centric Networking (ICN) concept as a possible architecture for future Internet [3]. In ICN, each data object is named and the hosts are replaced by these Named Data Objects (NDOs), as the main network entities. In addition, in-network caching capability is another feature of ICN that allows several copies of an NDO to be stored in intermediate nodes' caches, in addition to the end hosts. Due to

the names' stability, each data segment can be requested from the network and the network will be in charge of its provisioning.

First attempts for developing an information-centric networking architecture relates to TRAIID project in 1999 [47]. After that, many projects have been defined with the same idea, which are independently extending. Currently, major architectures on ICN are Content-Centric Networking (CCN) [20], Data-Oriented Network Architecture (DONA) [25], Network of Information (NetInf) [35], Publish-Subscribe Internet Technology (PURSUIT) [36], and Content-Centric Inter-Networking Architecture (CONET) [14]. CCN has attracted more attentions among these architectures considering the success of CCNx [9] and NDN [6] projects and so is considered in this study.

Some limited studies are performed on multimedia streaming and live TV broadcasting over ICN. Lederer et al. [27] have exploited DASH (Dynamic Adaptive Streaming over HTTP) standard over ICN (instead of HTTP) for multimedia communications. Liu et al. [33] address the effect of CCN caching mechanisms on quality of perceived video. Similarly, reference [32] evaluates the effect of caching and cache replacement strategies of CCN on quality of live TV broadcasting. To the best of our knowledge, channel zapping time has not been investigated yet in live TV broadcasting over ICN. However, channel zapping time is one of the major parameters that affect the QoE of the TV users. Synchronization and buffering times are the dominant components of the channel zapping time. Furthermore, other parameters such as routing and network congestion may also increase this time.

In this paper, a novel protocol is proposed that decreases the channel zapping time of live TV broadcasting over CCN. Moreover, the paper presents an analytical model of the proposed channel zapping protocol. Based on this analysis, some caching strategies and cache replacement methods are proposed to decrease the channel zapping time while improving the QoE of the perceived video. The proposed method reduces the synchronization and buffering time by effectively exploiting the in-network caching of ICN and modifying the method of channel zapping request. Although the Gartner networking hype cycle shows that named data networking is now on the rise [19] and it takes time to be considered as a commercial off-the-shelf technology, the share of ICN in future internet architecture seems inevitable. Therefore, the proposed method would be operational in the future implementations of TV broadcasting infrastructures.

The rest of the paper is organized as follows: In the next section, we describe the ICN technology and specially focus on CCN architecture as the basis of the proposed method. In section 3, previous works are presented. Section 4 introduces the proposed CCN-based channel zapping protocol and its analyses, and also presents the proposed caching methods. Section 5 demonstrates the simulation results and finally the paper is concluded in Section 6.

## 2 Information centric networking

Information centric networking architectures leverage in-network caching and exploit communication models that decouple senders and receivers to achieve more efficient content distribution comparing to the current Internet architecture and its overlay communication systems such as Peer to Peer (P2P) networking and proprietary Content Delivery Networks (CDN).

Although the first effort on ICN is related to the TRIAD project (which defines an explicit content layer over IP that provides content routing, caching, etc.) [47], DONA is the first

architecture that completely involves the data centric features [48]. NetInf project [13] is the other ICN project (part of 4WARD and SAIL projects on future Internet architecture) that presents a distinct architecture. PURSUIT [36] is another ICN project based on PSIRP architecture that aims to replace the IP protocol stack with a novel publish/subscribe based one. CONVERGENCE [12] is the other famous one that is based on CONET architecture [14]. Finally, CCN [20] is one of the most welcomed architectures, introduced by PARC. NDN and CCNx projects are of the famous ICN projects that are based on CCN architecture. Regarding the feasibility of being replaced as the future internet architecture, both projects currently have laboratory implementations that are exploited in the worldwide testbeds. Those are active projects that are supported by both academic and commercial partners. Thus, CCN has an increased chance of being commercialized in the future. Therefore, CCN architecture has been selected in this research and so will be described in more details.

Naming is hierarchical in CCN. Each individual's name is composed of a number of components. The human readable representation of the names is somehow similar to URLs. CCN communication is started by the consumer which broadcasts an *Interest* packet for a named data. Any node that receives the *Interest* packet and has a copy of the requested data, returns *Data* packet to the consumer. Otherwise, the request will be forwarded to the others. A named data satisfies the *Interest* packet if the *Name* field of the *Interest* packet matches the prefix of the *Data's Name*. Figure 1 shows the forwarding engine of a CCN node. It has three main data structures: Forwarding Information Base (FIB), Content Store (CS), and Pending Interest Table (PIT). FIB is responsible for forwarding the *Interest* packets to the potential source(s). It is much similar to the IP FIB table except the fact that CCN FIB allows forwarding *Interest* packets on multiple faces (as the named data may be available at multiple content sources). CS is the cache of the node and PIT keeps track of the *Interest* packets, forwarded toward content source(s), to route the data packets back to the requester(s) [20].

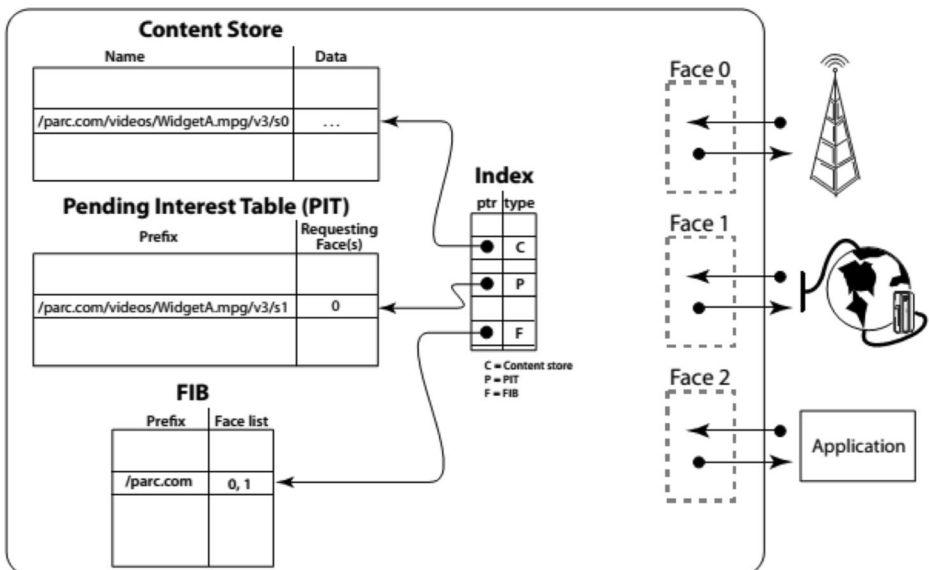


Fig. 1 CCN forwarding engine structure [20]

When an *Interest* packet is received by a face, a longest-match lookup is accomplished on its *Name* field to find it in CS, PIT, or FIB, respectively. If a matching is found in CS, the stored *Data* packet will be returned back on the incoming face towards the requester and the *Interest* packet will be discarded. If a PIT matching occurs, it means that an *Interest* packet for this *Name* has already been sent and so the new *Interest* packet will just be dropped. Otherwise, if there is a matching FIB entry, the *Interest* packet should be forwarded towards the potential data source(s) by removing the incoming face from the faces list of the FIB entry and sending the packet on all the remaining faces. Moreover, a new PIT entry is added to store the *Interest* and its arrival face [20].

In CCN, only *Interest* packets are routed. Therefore, *Data* packets just follow the reverse path of the corresponding *Interest* packet, regarding the PIT entries in PIT table of the nodes. For each *Data* packet, a longest-match lookup is done. A CS match means that the *Data* packet is duplicated and must be discarded while any PIT matching means that the data has already been requested by *Interest* packet(s). In PIT matching case, the packet is added to CS (after validating its name-data integrity) and is forwarded on all faces that are recorded in the matching PIT entry (except the incoming face). Then, the matching entry is removed from the PIT. In the case that no PIT matching is found, the *Data* packet will be discarded [20].

### 3 Related work

In this section, we review the previous works that have tried to improve the channel zapping time of the TV users. Firstly, the improvements in traditional IP-based TV broadcasting are studied and then the researches on the ICN-based ones are discussed.

#### 3.1 Channel zapping time in IP-based TV broadcasting

The dominant components of the channel zapping time are the waiting time for receiving the next I-frame of the video stream (synchronization time) and the buffering time of the receiver. Therefore, previous works have mostly focused on reducing these delay components.

A simple method for decreasing the synchronization time is shrinking the GoP (reducing the distance between two consecutive I-frames). However, this is equivalent to the increase of the number of I-frames, i.e. reducing the compression rate, which leads to the decrease of bandwidth efficiency [40]. Kalman et al. [21] focused on reducing the buffering delay, while frequent change of frame rate is the shortcoming of this method, which results in QoE degradation.

Some previous efforts [8, 29, 46] employ auxiliary streams to mitigate the zapping time of IPTV. In these methods, some dedicated channel zapping servers are considered that receive TV channels (similar to normal users) and cache the video frames. Whenever a Set Top Box (STB) requests joining a new multicast group, it also requests its local channel zapping server for a temporary unicast stream of the new channel that it wants to join. This method removes the synchronization time by streaming the video just from the next I-frame, but the cost of zapping servers and also the bandwidth utilization of unicast communications (especially in the presence of large number of channel switching requests) are its drawbacks. Reference [8] suggests those zapping servers to multicast the time shifted version of TV channels (instead of unicasting). Although this method reduces the overhead comparing to previous ones, the synchronization delay will not be zero.

Another category of research efforts [2, 4, 10, 22, 26, 28, 30, 31, 34, 37, 38, 41, 42] try to predict the next expected TV channels and pre-join them. Herein, the STB receives predicted channels in addition to the current channel and so the synchronization time will be removed, if the next channel is between the predicted ones. Reference [28] proposes a channel prediction method based on channels' popularity. In reference [4], fuzzy logic is exploited to evaluate the interest level of users to the channels. Ramos [41] has shown that the next selected channel is more probably between the neighboring channels (the channels that are adjacent to the current channel) and suggests pre-joining those neighboring channels by STB. Moreover, to reduce high overhead of pre-joining multiple channels, the time is divided into zapping period (the period of time that the user is surfing the channels) and viewing period (the period that the user stays on a channel and watches). Whenever users change the channel, they will enter the zapping period and whenever stay on a channel for a specific while, they will enter the viewing period. Therefore, pre-joining the neighboring channels is only performed during the zapping period which results in lower bandwidth wastage. However, considerable zapping delay is inevitable after exiting a viewing period.

The major drawback of prediction-based methods is their bandwidth waste due to receiving multiple channels concurrently. Some research works [28, 34] exploit video coding features to reduce this overhead. Similar to [41], the user is either in preview mode (switching between channels) or in watching mode. During the time that user is in preview mode, predicted channels are streamed to the STB with lower quality to reduce the bandwidth consumption. However, the effect of wrong prediction is still an important issue in these methods [2].

### 3.2 ICN-based TV broadcasting

In this subsection, we survey some researches that aim to deliver multimedia streams and TV channels over ICN. The related works are mostly proposed on CCN architecture and have mainly focused on adaptive video streaming over ICN.

Lederer et al. [27] addressed multimedia streaming over ICN-based mobile networks. Their research employs DASH (Dynamic Adaptive Streaming over HTTP) standard over CCN protocol stack (instead of HTTP). In DASH, multimedia streaming originates at the receiver and the latest GoPs will be requested at a bit rate (quality) corresponding to the current network quality (network bandwidth). Lederer et al. studied the performance of DASH in the presence of multiple links and also the Scalable Video Coding (SVC). It was showed that the potential multipath principles of CCN leads to a good performance of multimedia streaming in such networks.

Reference [15] presented a P2P over CCN based multimedia streaming method for mobile devices that provides concurrent receive of stream over multiple links (using the inherent capabilities of CCN).

Han et al. [18] designed and implemented AMVS-NDN (Adaptive Mobile Video Streaming on NDN) framework for adaptive mobile video streaming in NDN. This framework utilizes the mobile station's interfaces (3G/4G or WiFi links) opportunistically. It has been shown that this framework leads to high video quality in the time varying quality of the wireless links in a mobile environment.

Liu et al. [33] proposed an adaptive streaming mechanism over CCN architecture, called DASC. Thereafter, they investigate the effect of CCN caching strategies on the quality of the perceived video. They have shown that the consecutive interests for a data segment (except the first one) are responded more rapidly, using the CCN features. This results in overall

improvement of the perceived video. They also expressed that the CCN overhead is still large and needs improvement.

Reference [39] investigated the effect of different forwarding strategies and adaptation mechanisms on the adaptive multimedia streaming performance in NDN. Furthermore, the influence of various caching policies and different cache sizes were studied on the streaming performance. It has been shown that NDN's potential of multipath and its caching capability easily lead it to outperform the multimedia streaming over TCP/IP. This improvement can be reachable, if suitable forwarding strategy and enough cache sizes are utilized. It has also been mentioned that rate based adaptation cannot be used in NDN, as the available bandwidth could not be estimated accurately due to the multipath transmission and inherent caching of NDN. Therefore, client-based adaptation was used in which the nodes exchange their adaptation information.

Li et al. [32] evaluated the effect of CCN caching and cache replacement strategies on quality of time-shifted TV service. The in-network caching of CCN decreases the bandwidth utilization of the network in time-shifted TV. However, as their research shows, caching redundancy in a CCN domain leads to lower efficiency of the caches. Therefore, they proposed a cooperative caching strategy to better utilize the caching capability of the routers of the CCN domain. The proposed strategy reduces the caching redundancy.

Another effort that addresses TV broadcasting over ICN is presented in Reference [11]. This paper introduces a live TV broadcast system over CCN and studies the effect of caching strategies and cache replacement methods on live TV quality. Evaluation results have shown that in-network caching feature of CCN has no considerable effect on quality of live TV broadcast, as the cached frames will be outdated soon.

To the best of our knowledge, channel zapping has not been regarded in the previous ICN based TV broadcasting researches.

## 4 Proposed method

In this section, we describe our proposed method for live TV broadcasting over CCN that employs a novel protocol to reduce the synchronization and buffering delays. In the proposed method, buffering of the video frames is moved from the receivers to the network routers and even to the source. Therefore, recent GoPs of each video stream are cached by some of the content routers in CCN network (regarding in-network caching feature) or even by the source. This method accelerates channel zapping process since the receiver starts receiving frames just from the next I-frame and buffers them with the network speed, instead of buffering with video frame rate. It is worth noting that frame rate is always much lower than network speed. Hence, as the last GoP(s) has been cached in at least one of the routers of the path or eventually in the source, both the synchronization and traditional buffering delays are removed. However, download time of the GoP(s) has still remained as the bottleneck.

In the next subsection, the proposed CCN-based channel zapping protocol is introduced and then in subsection 4.2, channel zapping time is analytically modeled and compared to the traditional IP-based network. Regarding the importance of caching and cache replacement strategies in the proposed protocol, subsection 4.3 presents some novel strategies for improving the QoE of video channels.

### 4.1 Proposed channel zapping protocol

The proposed channel zapping protocol defines a new field in *Interest* and *Data* packets, called *ToS*. Using this field, the protocol defines two types of CCN *Interest* packets for video, namely *channel zapping request* and *video chunk request*. Similarly, the protocol defines two types of CCN *Data* packets for video, *channel zapping information* and *video chunk*. Whenever the user requests for a new channel, the STB sends a *channel zapping request* packet to the TV channel provider, indicating the volume of frames needed to be buffered before playback. The provider responds with a *channel zapping information*, indicating the range of chunk numbers (recent cached GoPs) that should be requested by the STB. In fact, to be synchronized with the live TV channel, the STB obtains the chunk numbers of the last GoPs from the TV channel provider. However, the video chunks are requested from the CCN network not the channel provider necessarily. The *channel zapping information* packets are not cached by any CCN router during the path.

When STB is informed about the chunk numbers, it sends consecutive *video chunk request* packets to the CCN network. Each router that contains a copy of the requested chunk (in its CS), or eventually the channel provider, responds with a *video chunk* packet. When all the requested chunks are received, STB starts playing the video stream of the new channel and simultaneously requests for the later chunk numbers. Routing of the *Interest* packets and forwarding of the *Data* packets are performed as determined in CCN architecture. Figure 2 shows the sequence diagram of the proposed protocol in detail. It is noteworthy that non-video *Interest/Data* packets are also distinguished, using *ToS* field.

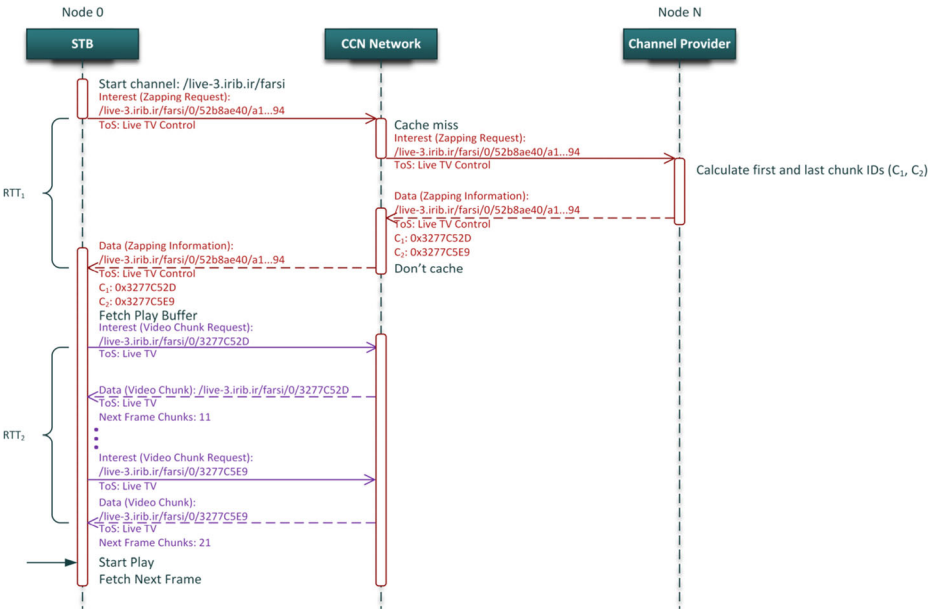


Fig. 2 Proposed channel zapping protocol (assuming that video chunks are cached in CCN routers)



## 4.2 Channel zapping time analysis

As it was mentioned, synchronization and buffering delay will be removed with the use of the proposed protocol. But network delay should still be considered due to the fetching of the buffered frames. In this section, an analytical comparison is provided for the channel zapping times of the IP-based method and the proposed method. Regarding the results of this comparison, suitable content caching strategies will be determined for the ICNs.

### 4.2.1 Zapping delay analysis of the proposed protocol

Following assumptions are considered in the formulation of the network delay as the main delay component of the channel zapping time in the proposed protocol:

1. Nodes' processing delay is neglected.
2. Links are assumed full duplex and symmetric.
3. Transmission queue of each node's interface contains different types of traffic with different transmission priorities. Weighted Fair Queuing (WFQ) is considered as the scheduling algorithm which divides the link's bandwidth between different types of traffic, according to their pre-assigned data rate.

As it is shown in Fig. 2, network delay is composed of two delay components,  $RTT_1$  and  $RTT_2$ .  $RTT_1$  is the delay of sending the *channel zapping request* packet to the sender and getting the *channel zapping information* packet back. Regardless of the nodes' processing delay, each of these terms is composed of queuing, transmission, and propagation delays. Eq. (1) shows the maximum queuing delay that each *channel zapping request* packet may experience on the traversed path ( $Dq_{zr}$ ).

$$Dq_{zr} = \sum_{k=1}^{N-1} \frac{Bs_k}{Rs_k} \quad (1)$$

Here,  $Bs_k$  is the maximum buffer length assigned to the signaling packets and  $Rs_k$  is the bit rate assigned to the signaling traffic, both at Node  $k$ . To calculate this delay, the maximum queuing delays of the nodes are added together. It should be noted that index  $0$  is assigned to the receiver and the sender's index is  $N$ . If the receiver wants to send *channel zapping request*, it will evacuate its transmission queue. Thus, no queuing delay is considered at the receiver, i.e. for  $k = 0$ .

Eq. (2) calculates the overall transmission delay of the *channel zapping request* packet on the route:

$$Dt_{zr} = \frac{L_{zr}}{R_0} + \sum_{k=1}^{N-1} \frac{L_{zr}}{Rs_k} \quad (2)$$

Here,  $L_{zr}$  is the *channel zapping request* packet's length and  $R_0$  is the link's bit rate of Node  $0$ . In the channel zapping time, the receiver's transmission queue is empty. Therefore, total bit rate of Node  $0$  ( $R_0$ ) is used to transmit the request packet. The case is different at the other nodes and a pre-assigned rate is allocated to the signaling traffic ( $Rs_k$ ) in this regard.

Finally, propagation delay of the *channel zapping request* packet is obtained by Eq. (3):

$$Dp_{zr} = \frac{d}{C} \quad (3)$$

In the above equation,  $d$  is the distance between the sender and the receiver and  $C$  is the signal's propagation speed.

According to the above equations, the elapsed time from the *channel zapping request* packet's transmission by the video receiver till its reception by the video sender can be calculated by Eq. (4):

$$T_{zr} = Dq_{zr} + Dt_{zr} + Dp_{zr} = \frac{d}{C} + \frac{L_{zr}}{R_0} + \sum_{k=1}^{N-1} \frac{L_{zr} + Bs_k}{Rs_k} \quad (4)$$

The elapsed time duration between the *channel zapping information* packet's transmission by the sender and its reception by the receiver is also similarly calculated by Eq. (5). Queuing delay, propagation delay, and also transmission delay at the sender are included in this equation.

$$T_z = \frac{d}{C} + \frac{L_z + Bs_N}{Rs_N} + \sum_{k=1}^{N-1} \frac{L_z + Bs_k}{Rs_k} \quad (5)$$

Here,  $L_z$  is the length of the *channel zapping information* packet and  $Bs_N$  is the maximum buffer length assigned to the signaling packets and  $Rs_N$  is the bit rate assigned to the signaling traffic, both at Node  $N$  (sender). Therefore,  $RTT_1$  delay can be calculated by Eq. (6):

$$RTT_1 = T_{zr} + T_z = \frac{2d}{C} + \frac{L_{zr}}{R_0} + \frac{L_z + Bs_N}{Rs_N} + \sum_{k=1}^{N-1} \frac{L_{zr} + L_z + 2Bs_k}{Rs_k} \quad (6)$$

The sender uses Eq. (7) to obtain the index of the first and last video frames ( $F_f$  and  $F_l$ ), required by the receiver. Thus, the number of required chunks can be calculated.

$$\begin{aligned} F_f &= I - i - \left\lfloor \frac{P \times F}{G} \right\rfloor \times G \\ F_l &= I \end{aligned} \quad (7)$$

In the above equation,  $I$  is the last broadcasted video frame by the channel,  $i$  shows the relative distance of the channel's last broadcasted video frame and the beginning of GoP,  $P$  is the temporal capacity of the receiver's playback buffer,  $F$  is the video frame rate and  $G$  is the length of GoP in terms of the number of frames.

The last video frame, transmitted for the receiver is the channel's last broadcasted video frame ( $I$ ). Temporal capacity of the receiver's playback buffer ( $P$ ) is multiplied by the video frame rate to calculate the number of receiver's required video frames. As the frames should be transmitted in complete GoPs, the result is shown as a coefficient of GoP's length. Finally, the number of receiver's required video frames and the number of broadcasted video frames of the broadcasting GoP ( $i$ ) are subtracted from the index of the channel's last broadcasted video frame ( $I$ ) to calculate the index of the first required

video frame. After that, the number of video chunks, required at the receiver to start playing the video, can be calculated by Eq. (8):

$$M = \left( i + \left\lceil \frac{P \times F}{G} \right\rceil \times G \right) \times \overline{fl} \tag{8}$$

In the above equation,  $\overline{fl}$  stands for the average number of video chunks included in a video frame and  $i$  is equal to  $G/2$  in average. Therefore, Eq. (8) can be rewritten by Eq. (9):

$$M = \left( \left\lceil \frac{P \times F}{G} \right\rceil + \frac{1}{2} \right) \times G \times \overline{fl} \tag{9}$$

The chunks of a video may be provided either by the sender or different routers. Thus, the time duration between transmitting the *video chunk request* packet and receiving the *video chunk* is a function of the probability of providing the video chunk at the traversed nodes. Eq. (10) will show this time duration ( $T_i$ ) for a *video chunk* packet, if it is provided by Node  $i$ .

$$T_i = \frac{2d_i}{C} + \frac{L_{cr}}{R_0} + \frac{\overline{Lc} + Bv_i}{Rv_i} + \sum_{k=1}^{i-1} \frac{Lcr + \overline{Lc} + 2Bv_k}{Rv_k} \tag{10}$$

Here,  $d_i$  is the distance of the receiver and Node  $i$ ,  $L_{cr}$  is the length of the *video chunk request* packet,  $\overline{Lc}$  represents the *video chunk* packet’s average length,  $Bv_k$  is the maximum buffer length assigned to the video packets and  $Rv_k$  is the bit rate assigned to the video traffic, both at Node  $K$ .

The reception of a video chunk is simultaneous with the transmission of a request for the next chunk and so  $RTT_2$  is not a coefficient of  $T_i$ . Therefore, the time duration between starting the request for the video chunks and the reception of the  $j^{th}$  video chunk, provided by Node  $i$ , is represented as  $T_{ij}$  and calculated by Eq. (11). Transmission delay of the  $(j-1)$  previous *video chunk request* packets is added to Eq. (10) to calculate  $T_{ij}$  in Eq. (11).

$$T_{ij} = \frac{2d_i}{C} + \frac{jL_{cr}}{R_0} + \frac{\overline{Lc} + Bv_i}{Rv_i} + \sum_{k=1}^{i-1} \frac{Lcr + \overline{Lc} + 2Bv_k}{Rv_k} \tag{11}$$

This equation will be obtained, if the  $j^{th}$  video chunk is provided by Node  $i$ . The video chunks are at least available at the sender. Thus, the average time elapsed between the request of the video chunks and the reception of the  $j^{th}$  video chunk can be written as Eq. (12). In this equation,  $P_{ij}$  represents the probability of providing the  $j^{th}$  video chunk by Node  $i$ .

$$\overline{T_j} = \sum_{i=1}^N P_{ij} T_{ij}, \text{ where } \sum_{i=1}^N P_{ij} = 1 \tag{12}$$

As the video chunks may be provided by the sender or different routers or they may even traverse different routes to the receiver, the order of their reception is not determined. Therefore,  $RTT_2$  is not equivalent to  $T_M$  and it can be obtained by Eq. (13):

$$RTT_2 = \text{Max}(\overline{T_j}); j \in [1..M] \tag{13}$$

Considering the proposed protocol (Fig. 2), channel zapping delay can be totally formulated by Eq. (14):

$$D_N = RTT_1 + RTT_2 \quad (14)$$

To analyze the effect of network delay on the channel zapping time in the proposed protocol, it is necessary to model the channel zapping time in a similar IP-based scenario as discussed in the next subsection.

#### 4.2.2 Zapping delay analysis of IP-based TV broadcasting

Channel zapping time in IP networks is composed of processing time, network delay, synchronization delay, and buffering delay. Processing delay is neglected in this section as well. Some assumptions are made in this section for the network delay calculation:

1. Nodes' processing delay is neglected.
2. Links are assumed full duplex and symmetric.
3. WFQ is considered to schedule the transmission queues of the nodes.
4. Routers use Reverse Path Forwarding (RPF) to avoid loops in multicast routing.

The average synchronization delay is equal to the time needed to receive half of one GoP's frames.

$$D_S = \frac{G}{2F} \quad (15)$$

The buffering delay is also equal to the time needed to receive the whole frames of  $W$  GoP.

$$D_B = \frac{WG}{F} \quad (16)$$

In the above equations,  $F$  is the number of broadcasted video frames per second,  $G$  represents the number of frames inside a GoP, and  $W$  is the number of GoPs needed in a receiver to start playing.

The time duration from the transmission of a join group request till the reception of the first video frame should be considered for the network delay calculation. The delay of receiving later video frames is considered in the synchronization and buffering delays. Eq. (17) shows the network delay in the condition of joining the multicast tree at Node  $i$ . It is worth noting that the reception of the first video frame is desired, not the first video chunk. Thus, the average number of video chunks that are included in a video frame is multiplied by the average length of a video chunk to calculate the average video frame's length.

$$T_i = \frac{2d_i}{C} + \frac{L_{Join}}{R_0} + \frac{\bar{fl} \times \bar{Lc} + Bv_i}{Rv_i} + \sum_{k=1}^{i-1} \frac{L_{Graft} + Bs_k}{Rs_k} + \sum_{k=1}^{i-1} \frac{\bar{fl} \times \bar{Lc} + Bv_k}{Rv_k} \quad (17)$$

Here,  $L_{Join}$  is the length of the IGMP *membership report* packet and  $L_{Graft}$  is the length of the RPF *graft* packet that is sent to graft the pruned branch to the multicast tree.

If  $p_i$  shows the probability of joining the multicast group at node  $i$ , Eq. (18) will be established. It is noteworthy that joining the multicast group will definitely happen at the sender.

$$\sum_{i=1}^N p_i = 1 \quad (18)$$

Therefore, Network delay is calculated by Eq. (19):

$$D_N = \bar{T} = \sum_{i=1}^N P_i T_i \quad (19)$$

And finally, channel zapping time can be obtained as:

$$D_Z = D_S + D_B + D_N \quad (20)$$

#### 4.2.3 Comparing channel zapping time of the proposed protocol to the IP-based TV broadcasting

Some assumptions are made for this comparison:

1. Network topology is the same for both scenarios and the same link rate ( $R$ ) is assigned to the live TV application in the nodes.
2. As the rate of channel zapping requests is lower than the rate of video chunk requests, buffer length of the signaling packets is assumed less than its length for the video traffic. It should be noted that signaling traffic has higher priority.
3. The maximum buffer length, allocated to the signaling traffic is equal to  $B_s$  in the network nodes.
4. The maximum buffer length, allocated to the video traffic is equal to  $B_v$  in the network nodes.
5. The nodes and their corresponding links do not change during the channel zapping process. Shortest path routing algorithm is used in the network and thus, it is probable that all the video chunks traverse the same route and reach the receiver in the right order.
6. To be fair, the node that has provided the video chunks in the proposed method (Node  $a$ ) is the same node in which the receiver has joined the multicast tree in the IP-based method.

According to the second assumption, signaling traffic has higher service priority. Thus, if signaling traffic exists in the network, the whole link rate ( $R$ ) will be assigned to it and  $R_s$  in Eq. (6) will be substituted by  $R$ . Considering assumptions 1, 3 and 4, the corresponding index of nodes' number will not be needed and Eq. (6) can be rewritten as Eq. (21):

$$RRT_1 = \frac{2d}{C} + \frac{L_{zr}}{R} + \frac{L_z + B_s}{R} + (N-1) \frac{L_{zr} + L_z + 2B_s}{R} \quad (21)$$

Providing that all the video chunks are received sequentially (assumption 5), Eq. (12) with  $j = M$  can be used to calculate  $RRT_2$  in the proposed protocol. If all

the video chunks are received from Node  $a$ , the probability of providing the  $j^{\text{th}}$  video chunk by Node  $i$  ( $P_{ij}$ ) is equal to 1 only for  $i = a$ , and otherwise, it is equal to 0. If the first four assumptions are also considered,  $RTT_2$  can be calculated as:

$$RTT_2 = T_M^- = T_{aM} = \frac{2d_a}{C} + \frac{M \times L_{cr}}{R} + \frac{\bar{L}_c + B_v}{R} + (a-1) \frac{L_{cr} + \bar{L}_c + 2B_v}{R} \quad (22)$$

According to two above mentioned equations and with substituting  $M$  by Eq. (8), channel zapping time will be written as follows:

$$D_{Z1} = \frac{2(d + d_a)}{C} + \frac{N(L_{cr} + L_c) + (2N-1)B_s + \left[ \left( W + \frac{1}{2} \right) G \times \bar{f}l + a-1 \right] L_{cr} + a \times \bar{L}_c + (2a-1)B_v}{R} \quad (23)$$

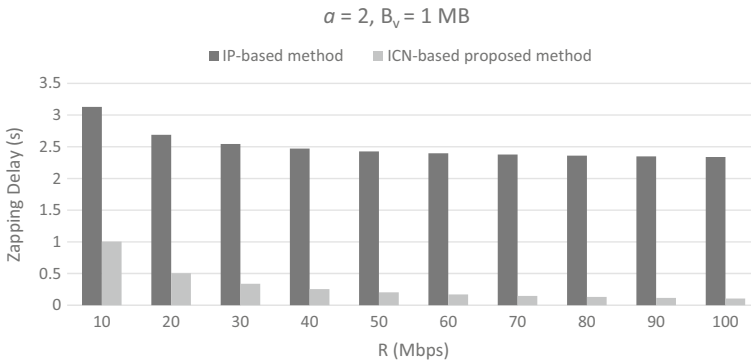
With similar analysis, channel zapping time in an IP-based TV broadcasting can be calculated by Eq. (24):

$$D_{Z2} = \frac{2d_a}{C} + \frac{L_{Join} + a(\bar{f}l \times \bar{L}_c + B_v) + (a-1)(L_{Graft} + B_s)}{R} + \frac{\left( W + \frac{1}{2} \right) G}{F} \quad (24)$$

As can be seen in Eq. (23), channel zapping time includes two terms. The first term is related to the propagation delay and can be mitigated by shortening the distance between the receiver and the node storing the video chunks. The second term reflects the effect of transmission and queuing delays that can be lowered by decreasing  $a$  and increasing  $R$ . The other variables in this equation are almost unchangeable. On the other hand, Eq. (24) shows that channel zapping time in an IP-based network includes three terms. The first term is related to the propagation delay and can be lowered by decreasing  $a$ . The second term relates to the transmission and queuing delays and can be mitigated by decreasing  $a$  and increasing  $R$ . But the third term that reflects the effect of the buffering and synchronization delays cannot be improved.

**Table 1** Parameters setting for the numerical analysis

Parameter	Value
$d$	$5 \times 10^5$ m (500 Km)
$d_i$	$10^4$ m (10 Km)
$C$	$2 \times 10^8$ m/s (200,000 Km/s)
$N$	7
$L_{zr}$	2400 bit (300 B)
$L_z$	4000 bit (500 B)
$B_s$	81,920 bit (10 KB)
$W$	4
$G$	15
$\bar{f}l$	5
$L_{cr}$	1200 bit (150 B)
$\bar{L}_c$	81,920 bit (10 KB)
$L_{Join}$	480 bit (60 B)
$L_{Graft}$	480 bit (60 B)
$F$	30 fps



**Fig. 3** The effect of the link's bit rate on the channel zapping time

Following, the effect of  $R$ ,  $\alpha$ , and  $B_v$  on the channel zapping time of both methods will be compared using numerical analysis. Table 1 shows the parameter settings for this comparison. Figures 3, 4, and 5 show the effect of  $R$ ,  $\alpha$ , and  $B_v$  on the channel zapping time, respectively.

Delay analysis shows that channel zapping time is directly dependent to the caching location of the video chunks in the proposed algorithm. Channel zapping time will decrease, if the video chunks' caching location comes nearer to the receiver. Therefore, suitable caching and cache replacement strategies are sought for the proposed method in the next section.

It is noteworthy that although the proposed protocol could be adaptable for current IP network, its superiority tightly depends on caching location of packets. However, current internet caches are located at limited positions of the edge of the network and this does not fully comply with the inherent multicasting feature of ICN.

Network congestion is another important factor that directly affects channel zapping time and received video quality. While differentiated service classes are needed for TV broadcast to improve delay and received video quality, there is no service classification field in the CCN packets format and CCN nodes do not inherently support QoS. Thus, some changes should be made in the packets to support such ability in CCN. Therefore,  $ToS$  field is added to the CCN packets to create the possibility of flow differentiation in the intermediate routers as mentioned earlier.

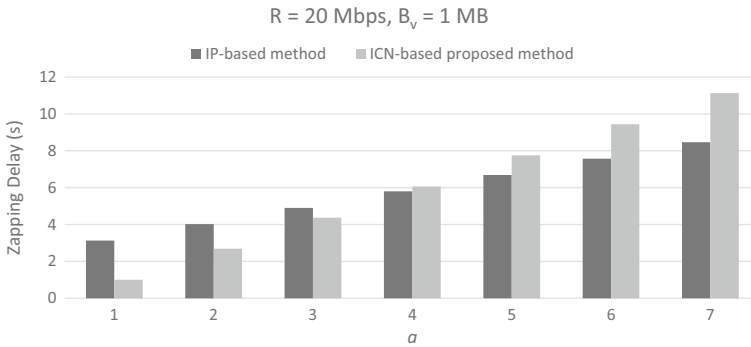
### 4.3 The proposed caching and cache replacement strategies

As mentioned before, network delay can be reduced by decreasing the distance of the receiver and the caching location. Therefore, indefinite caching strategies such as fixed probability, probable cache and the strategies that leads to the data caching far from the receiver (such as LCD,<sup>1</sup> BTW<sup>2</sup>) [1] are not suitable for this purpose. LCE<sup>3</sup> caching strategy caches data packets all over the route. Although this strategy leads to low network delay, it is not an efficient method in routers' cache usage. Video frame caching in the edge routers requires a new caching strategy. Thus, Cache Video on Edge (CVE) is proposed as its pseudo code is shown

<sup>1</sup> Leave Copy Down

<sup>2</sup> Betweenness Centrality

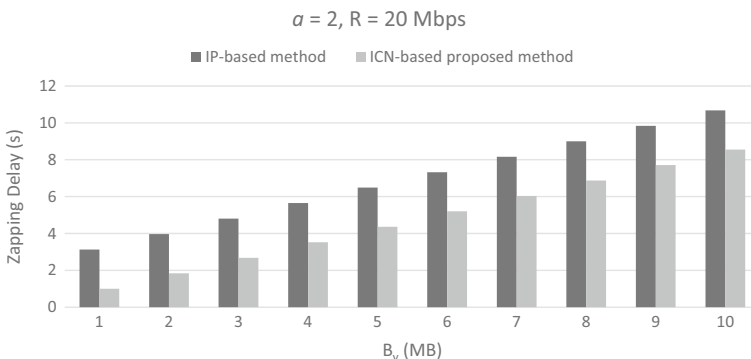
<sup>3</sup> Leave Copy Everywhere



**Fig. 4** Channel zapping time versus the distance between the video receiver and the provider node

in Fig. 6. All the network routers use this strategy for live TV application, while the caching strategy for the other applications is indicated independently as determined by *cache\_strategy* argument (which its indication is out of the scope of this paper). Router specifies its own location on the route, based on the hop field value. Adding the *ToS* field in the header of the data packets and also the addition of hop field to the header of data and interest packets are required to implement this strategy.

Video reconstruction at least depends on the last played internal frame and its next frames. However, to overcome the network jitter, a window of GoPs is cached in the proposed method where  $W$  determines this window's length. CVE only caches the received video chunks in the routers near the receivers. This caching strategy has some drawbacks: first of all, lots of copies of a video chunk are cached in the network. Secondly, every edge router that is receiving a similar request for the first time will start a separate connection with the sender. Although the inherent characteristic of CCN in aggregating the interests decreases the upstream bandwidth requirement, the asynchronous requests cannot be aggregated. Asynchronous requests may occur due to the delay variation of different routes or the difference in the playback buffer size of the receivers. In such conditions, intermediate routers that do not have a copy of the video chunks, will receive them iteratively. Thus, bandwidth usage will increase. Figure 7 shows these two problems. In this figure,  $STB_2$  has started receiving stream  $S_2$  at time  $t_1$ . After a while,  $STB_3$  requests the same stream but Router  $R_1$  do not store the stream because of the



**Fig. 5** Channel zapping time versus the maximum buffer length assigned to the video traffic



```

function SHOULD_CACHE(data_packet, cache_strategy, W, gop_len, fps): bool
  # Don't cache live TV control messages
  if data_packet.tos = live_tv_control then
    return false
  # Devolve caching decision of non live TV data packets to cache_strategy
  if data_packet.tos <> live_tv then
    return cache_strategy(data_packet)
  # Don't cache live TV data packets on core routers
  if data_packet.hop <> 1 then
    return false
  # Don't cache expired frames
  frame ← data_packet
  timeout ← W × gop_len / fps
  if now - frame.time > timeout then
    return false
  return true
end function
    
```

Fig. 6 CVE's pseudo code

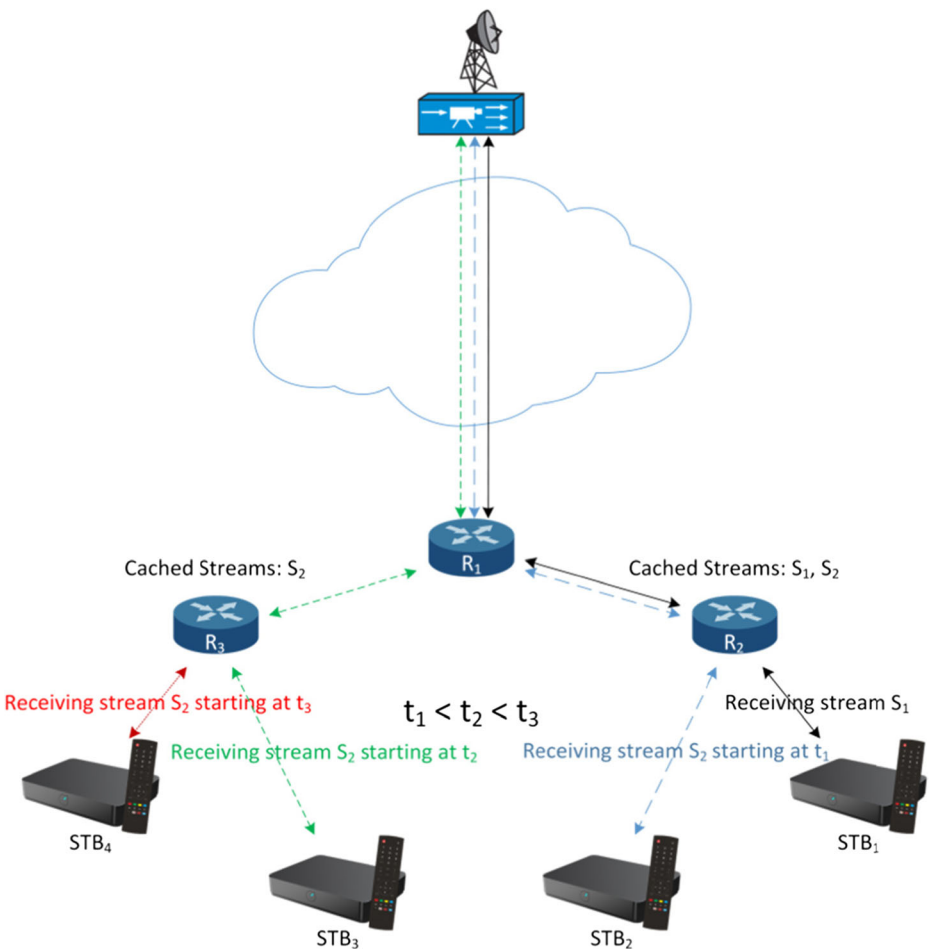
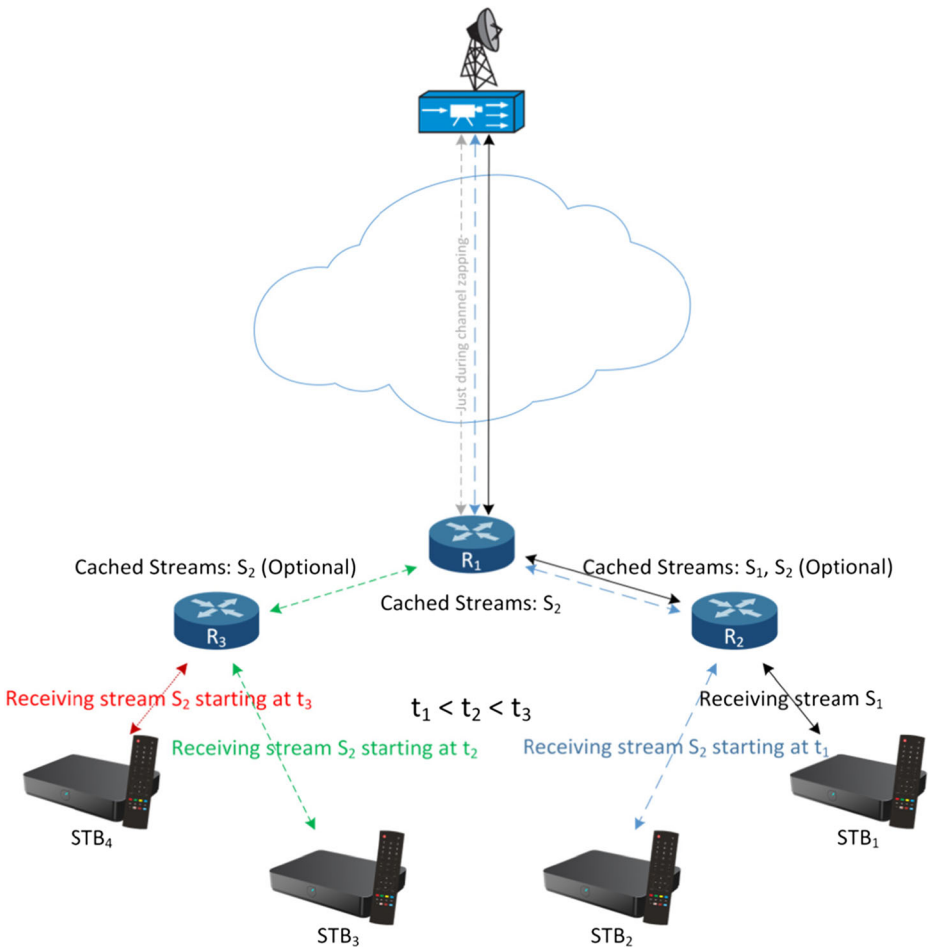


Fig. 7 The side effects of CVE usage



**Fig. 8** The effect of DVC usage

employed caching strategy. Thus, a new stream is initiated. Later requests for the same stream from STB<sub>4</sub> will be responded by R<sub>3</sub>.

Therefore, another caching strategy is proposed to solve the mentioned problems. In the new strategy, called DVC,<sup>4</sup> all routers decide to cache the new chunks of live TV dynamically. However, the decision about the caching of other applications' data packets is taken by the other caching strategy. Figure 8 shows an example of using this strategy and its effect on the scenario showed in Fig. 7. In this scenario, STB<sub>1</sub> is receiving the S<sub>1</sub> flow and STB<sub>2</sub> is receiving the S<sub>2</sub> flow and R<sub>2</sub> should cache the video chunks of both flows. When STB<sub>3</sub>, which is connected to R<sub>3</sub> requests for receiving S<sub>2</sub> at time t<sub>2</sub>, it finds S<sub>2</sub>'s cached video chunks neither in R<sub>3</sub> nor in R<sub>1</sub>. Therefore, it takes action to receive them from the main sender. At this time, R<sub>1</sub> that finds the request for S<sub>2</sub> flow from multiple interfaces, caches the returned video chunks and lets R<sub>2</sub> to cache them optionally. The next request for S<sub>2</sub> flow that is issued by R<sub>3</sub> will be responded by either R<sub>3</sub> or R<sub>1</sub>. As a result, bandwidth usage will decrease.

<sup>4</sup> Dynamic Video Caching

```

function SHOULD_CACHE(data_packet, continuous_cache, cache_strategy, W,
gop_len, fps): bool
  # Don't cache live TV control messages
  if data_packet.tos = live_tv_control then
    return false
  # Devolve caching decision of non live TV data packets to cache_strategy
  if data_packet.tos <> live_tv then
    return cache_strategy(data_packet)
  # Don't cache data packets which explicitly marked as don't cache
  if not data_packet.cache then
    return false
  # Don't cache expired frames
  frame ← data_packet
  timeout ← W × gop_len / fps
  if now - frame.time > timeout then
    return false
  # Decide to cache live TV data packets on core routers based on the number
  # of requesting faces
  if data_packet.hop <> 1 then
    if pit.find(sid).requesting_faces.size() > 1 then
      data_packet.cache ← continuous_cache
    else
      return false
  return true
end function

```

**Fig. 9** DVC algorithm

Figure 9 shows DVC caching strategy. This strategy is also used in all the network routers and is only applied to the live TV broadcast. It is worth noting that caching strategy for the other data packets is determined by *cache\_strategy* argument and old video chunks will not be cached. Core routers take their caching decision based on the number of flow requests. When a router caches a frame, it will determine if the next routers need to store it or not, based on *continuous\_cache* parameter. *ToS* field of the header of data packets, hop field of data and interest packets, and cache field of data packets are also required to implement this strategy.

Video chunks' replacement strategy is also important in these networks. Old video frames lose their importance in live TV service, unlike the video on demand service. Thus, a suitable replacement strategy should preserve the last cached video chunks. The same as caching, indefinite replacement strategies (such as random replacement) are not suitable. FIFO replacement strategy also has an important problem. In a network that TV service is presented besides other applications, using such a strategy means the possibility of fast replacement of cached video chunks by the entered data packets of the other applications. LRU<sup>5</sup> replacement strategy acts similar to FIFO for live TV application. As live video packets have short life time and will never be requested again, the least recently used packet, that should be replaced, has entered sooner than the other packets. The simplest method that can solve this problem is partitioning the cache and allocating a specific partition to the live TV application. But this method decreases the cache efficiency. Therefore, a new replacement strategy, namely Keep Intra-Pictures (KIP) is proposed in this study. Figure 10 shows the procedure that is called for releasing the cache whenever it is full. This procedure uses KIP strategy for the cache release.

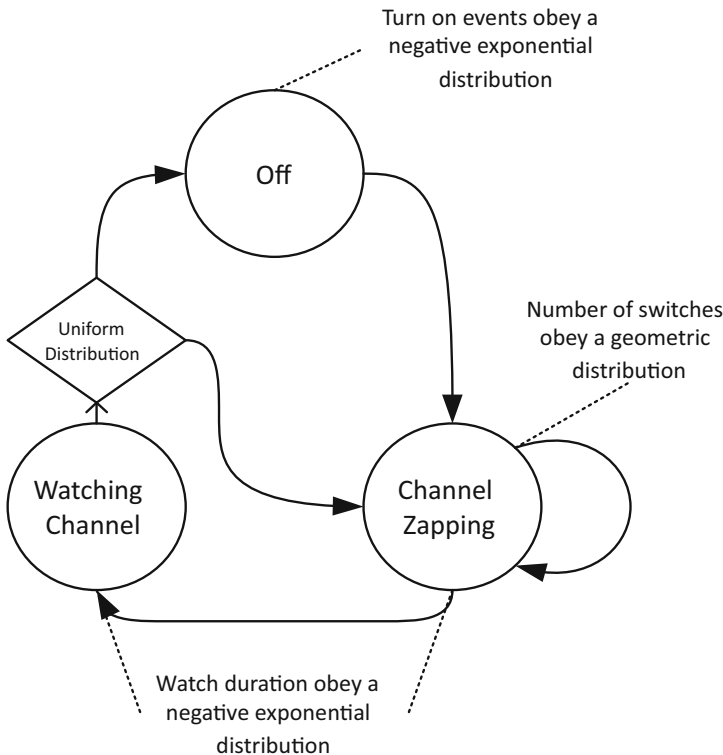
<sup>5</sup> Least Recently Used

**Fig. 10** KIP cache replacement strategy

```

procedure CLEANUP(W, gop_len, fps, rep_strategy)
  cleaned  $\leftarrow$  false
  # Cleanup each stored stream from expired frames
  timeout  $\leftarrow$  (W + 1)  $\times$  gop_len / fps
  for each stored_stream do
    for each frame do
      if now - frame.time > timeout then
        remove(frame)
        cleaned  $\leftarrow$  true
      end if
    if not cleaned then
      if non_live_tv_data_packets.size() > 0 then
        # Cleanup non live TV data packets
        rep_strategy(non_live_tv_data_packets)
      else
        # Cleanup recent frame
        remove(recent_frame)
      end if
    end if
  end procedure
  
```

First of all, this strategy releases the video chunks of frames which are located out of the window of GoPs (for all of the cached flows). If no video chunk is found for deletion, the other



**Fig. 11** State machine of TV receiver



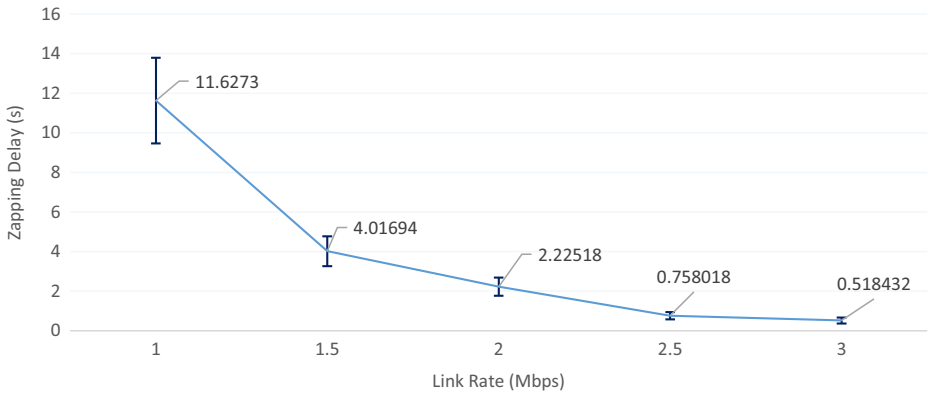
**Fig. 12** Simulation scenario

applications’ data packets will be deleted. Deletion of these data packets follows the strategy determined by *rep\_strategy* argument. Finally, if no data packet is found for deletion, the chunks of newest cached video frame will be deleted. This decision has two important advantages:

1. The newest cached video frame is the newest played one too. Thus, it is located far from the frame that should be played (regarding playback buffer of receivers) and there is

**Table 2** Simulation parameters

Parameter	Value
Number of users in each STB	25
Links’ Bit Error Rate (BER)	$10^{-9}$
Number of channels	100
PIT entry time-out	2.0 s
Capacity of the caches	5120 chunks
Maximum buffer length of the interfaces	1 MB
Routing strategy	Shortest Path
Caching strategy	LCE
Cache replacement strategy	LRU
Scheduling strategy	FIFO
Average off time of the STBs	20 min
Average number of channel switching before entering the watch state	5
Average watching time between consecutive zapping requests	10 s
Average watching state’s residence time	2.5 min
Simulation time	1 h



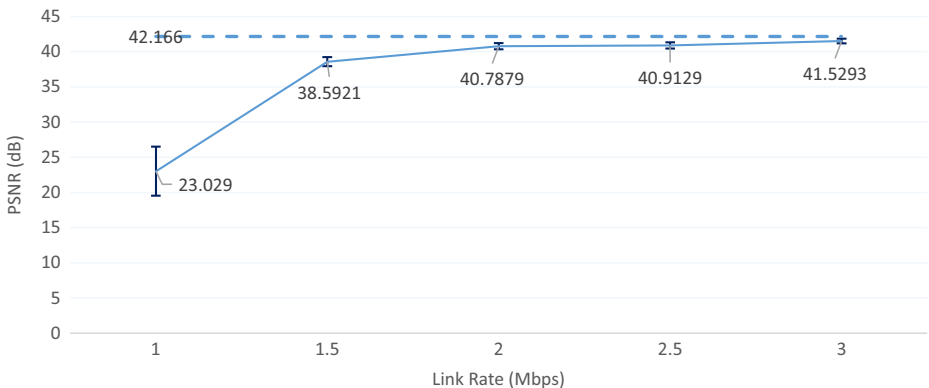
**Fig. 13** Channel zapping time versus speed of links

enough time to retrieve it again. Furthermore, as it is novel, its next receipt requests will more probably come from others and so the router is probably triggered to receive it again.

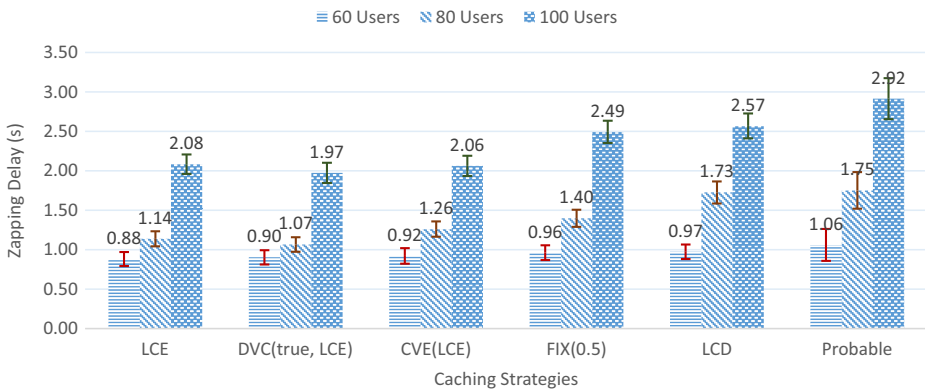
- Sequential deletion of the chunks of video frames, starts with the newest received one, deletes the worthless frames (i.e. P-frames and B-frames) sooner than worthy I-frames. This will continue till a full GoP is deleted from the cache. The mentioned prioritization does not need any video frame differentiation. Thus, no changes are required in the packet format or in the name of the content and name of the content will remain opaque for the router. The window of GoPs has the highest priority in this strategy and the other video frames have no importance. As the number of TV flows passing through each router is not high and since the number of video frames cached for each flow is not significant, this prioritization will not be harmful for the other applications and can be used all over the network either in the edge or in the core routers.

### 5 Performance evaluation

This section demonstrates the simulation results of the proposed method. Simulations have been performed by the Otanes simulator [16], which is a CCN simulator over the OMNeT++



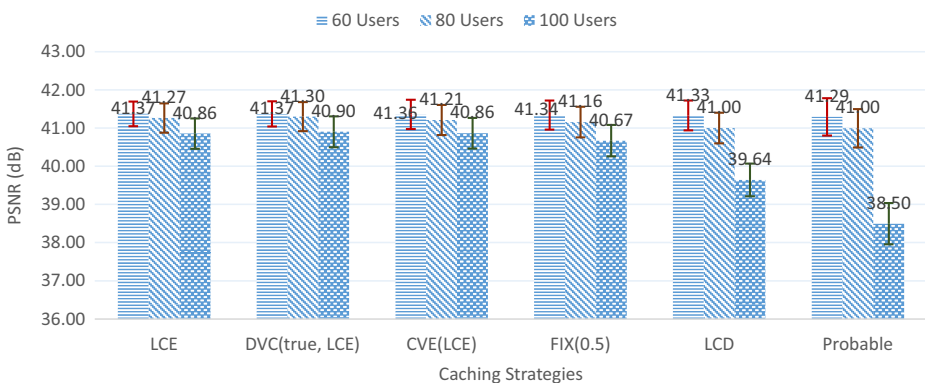
**Fig. 14** Average PSNR level of the received video frames versus speed of links



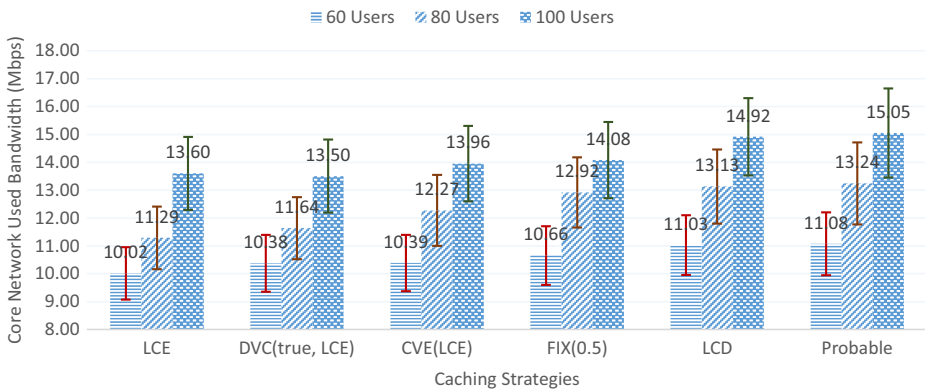
**Fig. 15** Zapping delay versus various caching strategies under three different network loads

framework. Otanes supports video reconstruction to measure the received video quality using the Evalvid framework [23]. This simulator supports tree types of nodes: Providers, Consumers, and Content Routers. Providers are either file provider or TV provider. In our simulations, file provider is used to simulate the background traffic of the network. Each TV provider broadcasts multiple TV channels simultaneously where MPEG-4 video files are used to simulate the live channel contents. Consumers are also divided into three types: file receiver, TV receiver, and probe. File and TV receivers simulate multiple users simultaneously to simplify the simulation of large networks. Probe is a special TV receiver that simulates one user. It provides the capability of re-constructing real video stream from the simulation traces, using the Evalvid tool. Therefore, probe node estimates the QoE of the real video stream by assessing the PSNR level of received video frames.

Each file receiver uses a Poisson process to randomly request for a file. The state machine, shown in Fig. 11 is used to model the behavior of TV users. Users switch between off, zapping, and watching states. The time that STB stays in off state follows a negative exponential process. After exiting the off state, STB sends several channel switching requests, the number of which complies with geometric process. The time that STB stays in watching or zapping states follows a negative exponential process too. The channel that is selected by a probe/TV



**Fig. 16** Average PSNR level of the video frames versus different caching strategies under various network loads



**Fig. 17** Network bandwidth utilization versus different caching strategies under various network loads

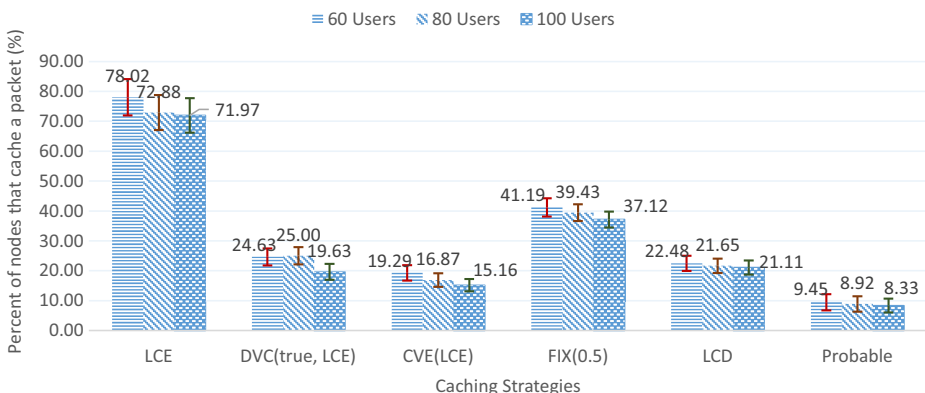
receiver (and also the file that is selected by the file receiver) follows the discrete popularity distribution of Zipf-Mandelbrot.

### 5.1 Simulation environment

The topology of the simulation scenario is Abilene network model [45]. Several TV providers (TVP), file providers (FP), TV receivers (STB), file receivers (FR), and probe have been attached to it as shown in Fig. 12. The frame rate of the video streams (that model live TV channels) is 30 fps and each GoP contains 60 video frames. The size of the largest chunk is 10 KB. Table 2 shows the simulation parameters in detail. The simulation results are average of multiple runs and include 95 % confidence intervals.

### 5.2 Evaluating the effect of links speed

As stated in Section 4, the proposed protocol removes synchronization and buffering delays, while the effect of the network throughput on the GoPs’ fetching time must be regarded. Therefore, we change the speed of the links between the network nodes and evaluate its effect on the channel zapping time and also PSNR of the video streams. Figure 13 shows the effect of



**Fig. 18** Average percent of nodes that cache a packet along the path in various caching strategies



**Table 3** Modified simulation parameters for evaluation of file transfer application

Parameter	Value
Number of users in each FR	20, 12, 4
Number of files	10 <sup>6</sup>
Average length of files (geometric distribution)	100 MB
Transmission window of FR	10
Receiver timeout (waiting for a file segment)	1.0 s
Average inter-arrival time of the file requests for a user	2.5 min
Links rate	3 Mbps

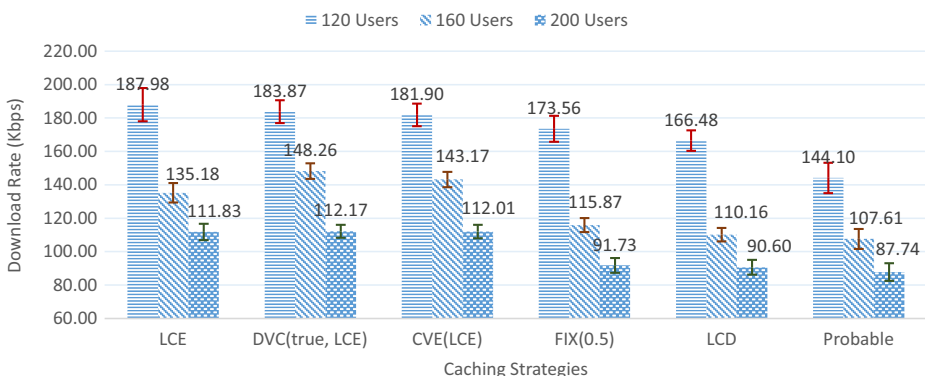
links’ speed on the zapping delay. It is clear that as the speed of links increases, the zapping time decreases exponentially. This trend has also been approved by analytical results, shown in Fig. 13.

Figure 14 shows the effect of links’ speed on the PSNR level of video streams. The reference PSNR level of the original video stream (regarding its uncompressed version) is around 42 dB, as shown in this figure. It is clear that the PSNR level increases dramatically as the bandwidth of the links increases. This is due to the fact that higher link speed reduces the drop and delay of the video chunks.

### 5.3 Evaluating the effect of caching strategies

In Section 4, we show the effect of caching location on the channel zapping time. As the location where video chunks are cached depends on the caching strategy, this subsection demonstrates the effect of caching strategies on the zapping time in the proposed protocol. Alongside, we consider the effect of caching strategies on the PSNR level of the video streams and quality of non-TV applications to study the probable tradeoff in this regard. In the following simulations, we assume the speed of links to be 2Mbps.

Figure 15 shows the effect of different caching strategies on the channel zapping time under various network loads (i.e. total number of TV users). LCE strategy stands for Leave Copy Everywhere and caches the content in all the routers of the path while LCD (Leave Copy Down) caches the content in the routers that are far from the receiver. FIX(0.5) stands for



**Fig. 19** Download rate of the file users versus caching strategies under various loads

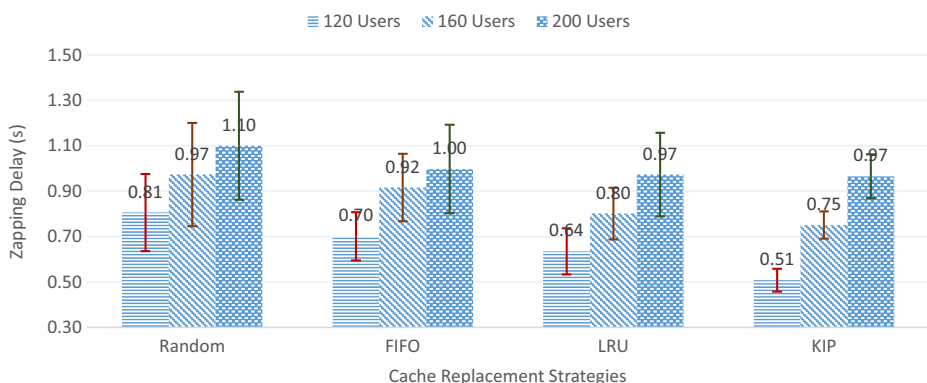
**Table 4** Modified simulation parameters for the evaluation of cache replacement strategies

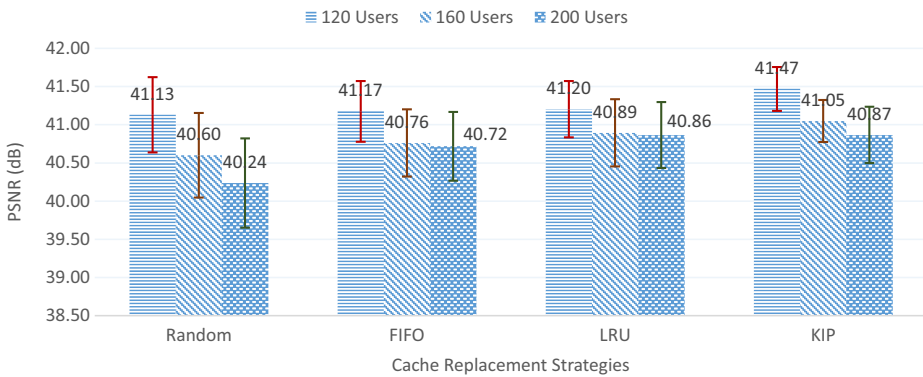
Parameter	Value
Number of TV channels	10
PIT entry time-out	4.0 s
Cache size	2048 chunks
Caching strategy	LCE
Links rate	20 Mbps
Receiver timeout (the time waiting for a file segment)	2.0 s

probabilistic caching with fix caching probability of 0.5. DVC(true, LCE) indicates the proposed DVC caching strategy that allows caching of the video chunks in all the routers of the path (continuous cache) and uses LCE caching for the non-video packets. CVE(LCE) demonstrates the proposed CVE caching strategy for the video traffic and LCE strategy for the non-video ones. As the results show, LCE, DVC, and CVE strategies that certainly cache the video chunks at the routers which are closer to the STBs, demonstrate lower zapping delay especially in higher load conditions.

Figure 16 shows the average PSNR level of the video frames versus the above mentioned caching strategies. As it is shown, LCE, DVC, and CVE caching strategies have not almost affected the average PSNR level of the video stream. This illustrates that using those strategies with the proposed channel zapping protocol does not impose side effects on the QoE of the streaming. For LCD and probable caching, such effect is noticeable under higher network loads. This is due to the fact that in LCD, the content is more probably cached in the routers located near the providers (than receivers) and in probable caching, the probability of cache hit is low in overall. Therefore, more routers should be traversed by the packets and the network congestion will further increase.

While proposed caching strategies improve the zapping delay, their effects on the bandwidth utilization must also be considered. It is obvious that as the cache hit rate increases and the cached content comes closer to the receiver, the bandwidth utilization of the network increases. Figure 17 shows the aggregate bandwidth usage of the core network under various caching strategies. An advantage of DVC caching strategy is its capability to aggregate non-concurrent *interests* for content and so decreasing the bandwidth usage under higher network

**Fig. 20** Zapping delay versus cache replacement strategies under various network loads



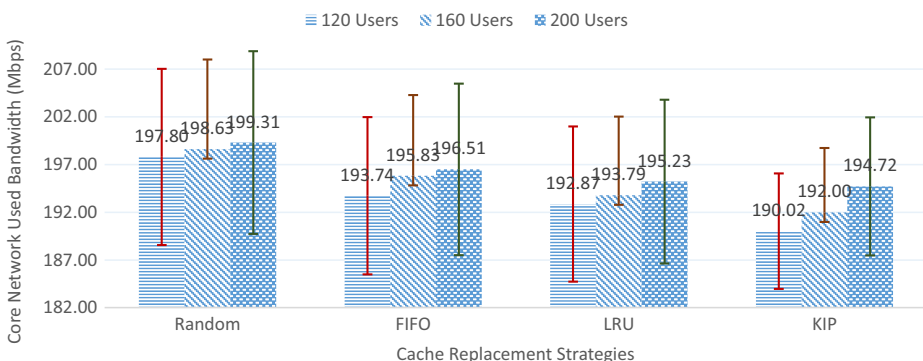
**Fig. 21** Average PSNR level of perceived TV channels versus cache replacement strategies

loads. This is due to the fact that more users may request for a chunk in higher network loads and the chance of aggregation will increase. Simulation results confirm the above mentioned claim. However, the performance of caching strategies is also related to the other parameters such as routing mechanism that its evaluation is out of the scope of this paper.

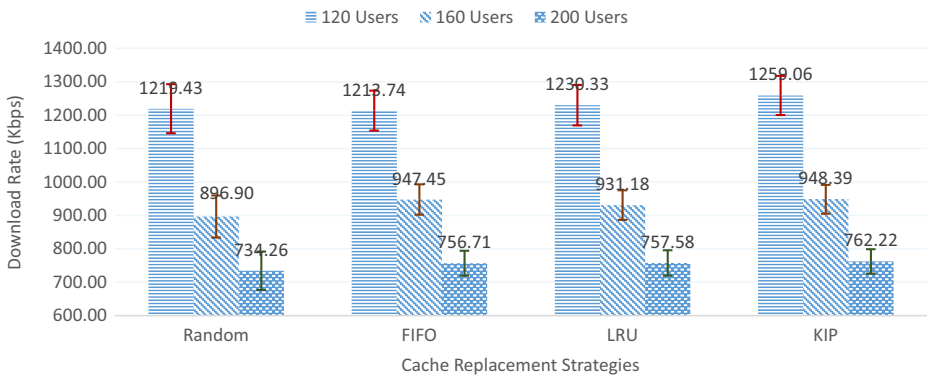
From the above assessments, it may be concluded that LCE, DVC, and CVE are almost similar in performance. However, the key difference is in their memory usage that DVC and CVE provide this level of performance with much lower memory consumption. Figure 18 shows the average percent of routers that will cache a data packet along the path. It is clear that cache utilization of LCE is higher than the proposed strategies.

The proposed caching strategies for TV application must not affect other ICN applications. Therefore, the effect of those strategies on the file download is also evaluated. For this evaluation, the simulation parameters are modified according to Table 3. In this simulation, different network loads are also modeled by changing the number of users. Since the number of STBs is 4 and the number of FRs is 5, 200 users means 25 users per each TV receiver (STB) and 20 users per each file receiver (FR). Similarly, in 160 and 120 users’ scenarios, the number of users is 12 and 4 per each FR respectively, while the number of users is 25 per each STB.

Figure 19 shows the effect of the strategies on the average download rate of FR users. As the figure shows, the proposed strategies improve download rate comparing to the traditional ones, especially FIX, LCD, and probable caching.



**Fig. 22** Network bandwidth consumption versus cache replacement strategies



**Fig. 23** File download rate versus cache replacement strategies

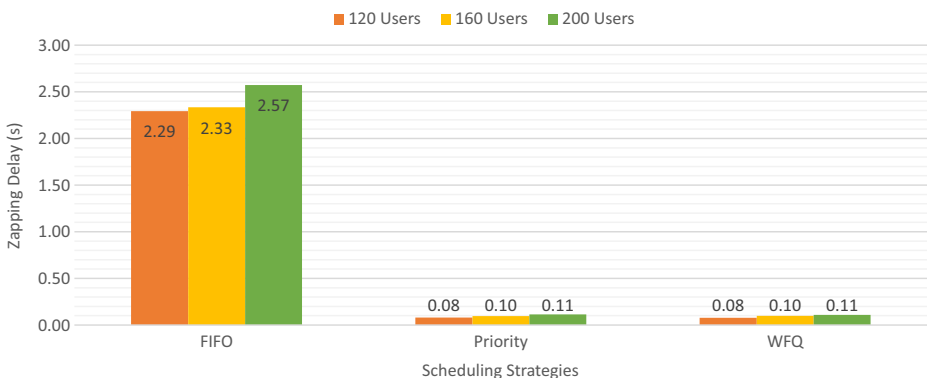
### 5.4 Evaluating the effect of cache replacement strategies

In this section, we evaluate the effect of proposed cache replacement strategy on the channel zapping time and PSNR level of video channels comparing to the conventional strategies. Table 4 shows the simulation parameters that are different from Table 1.

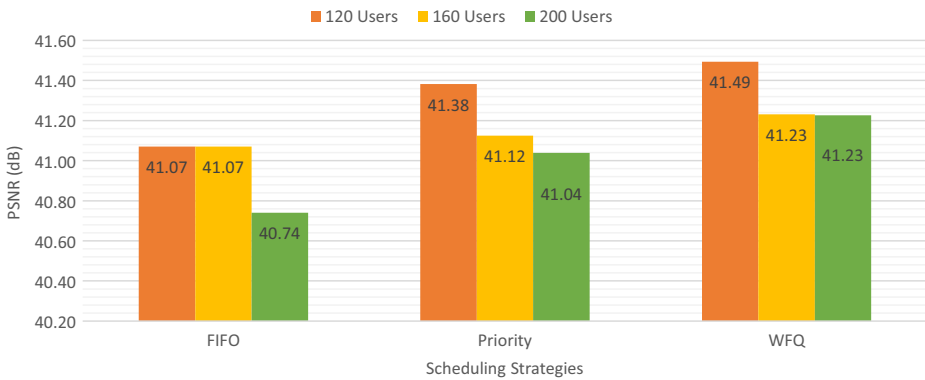
Figure 20 compares the effect of cache replacement methods on channel zapping time. Random strategy removes the cached packets randomly under the overflow conditions. FIFO removes the packets according to their arrival order and LRU strategy removes the packets that are requested infrequently. The results show that proposed KIP strategy demonstrates lower zapping time, although the performance is much similar to LRU, especially when the network load is high.

Video perceived quality is also compared in different cache replacement strategies, as illustrated in Fig. 21. As KIP tries to hold more recent packets in the cache, subsequent *interests* for a video chunk are more possibly responded by the routers’ caches (which are almost near the receiver). Therefore, TV channels experience less frame delay and frame drop.

Figure 22 compares the strategies in term of network bandwidth consumption. KIP method reduces the overall consumed bandwidth. Because, the increasing chance of cache hit reduces the volume of *Interest/Data* packets that are propagated in the network core. We also have investigated the effect of cache replacement strategies on the file download traffic as shown in



**Fig. 24** Channel zapping delay versus three different scheduling algorithms



**Fig. 25** Average PSNR level of the video frames versus 3 different scheduling algorithms

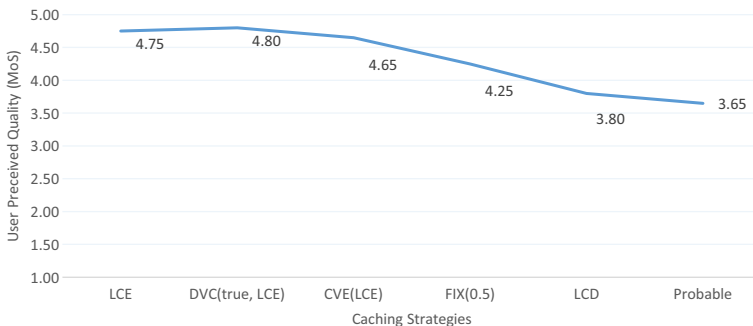
Fig. 23. The results show that the proposed method has improved the download rate comparing to the traditional ones, although this improvement is not significant.

### 5.5 Evaluating the effect of scheduling strategies

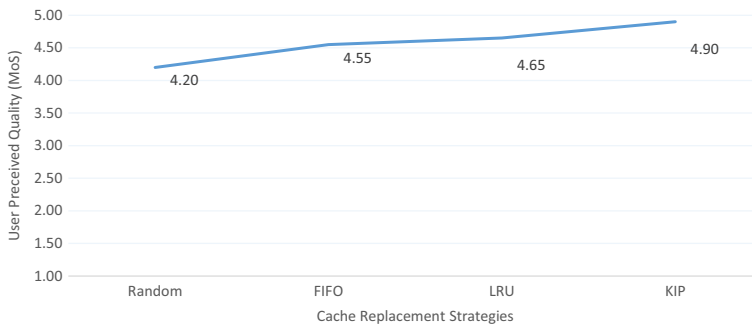
As discussed in Section 4, proposed protocol defines *ToS* field in CCN packets and packets are scheduled according to their *ToS*. In this section, we evaluate the effect of packet scheduling methods on the proposed protocol. For this evaluation, links’ rate is assumed to be 5Mbps and LRU cache replacement strategy is exploited. Figure 24 compares three famous scheduling algorithms in term of zapping delay. It is obvious that both WFQ and Priority Queuing (PQ) mechanisms improve the zapping delay since prioritizing the video traffic reduces the network delay as the main component of zapping time. Based on the same argument, PSNR level of the video frames improves under PQ and WFQ methods dramatically, as shown in Fig. 25.

### 5.6 Subjective evaluation of caching and cache replacement strategies

As the PSNR based objective evaluation may not reflect the real perception of users, we have evaluated the proposed caching and cache replacement strategies via subjective evaluations too. Doing so, the caching strategies have been compared regarding the simulation parameters of Table 2 as shown in Fig. 26. The comparison has been performed for the case with 100 video users and the results are the average of MOS levels of received videos that are assessed



**Fig. 26** Average MOS level of the received video streams versus the caching strategies



**Fig. 27** Average MOS level of the received video streams versus cache replacement strategies

by 10 subjects (real users). Similar to objective assessment (Fig. 16), the results indicate that DVC(true,LCE) caching strategy is the best and the probable caching strategy is the worst.

Figure 27 has compared the cache replacement methods for the case with 200 video users. Similarly, the results are the average MOS level of received videos that are assessed by 10 real users. As the results show, the KIP method better satisfies the users.

## 6 Conclusion

Channel zapping delay is a challenging issue in delivering TV services over the Internet infrastructure that highly affects the QoE of TV users. Therefore, a new protocol was proposed in this paper, which is based on the CCN architecture. The proposed protocol replaces the synchronization and buffering delays of the channel zapping (which depend on the playback rate) with a new delay component that depends on the network throughput. The general idea of the proposed solution is using the CCN features to buffer the frames of recent GoPs in the routers or even in the TV providers and sending them to the requesting STBs at the network speed. Analytical modeling of zapping delay shows that using the proposed protocol eliminates the dependence of this delay to the synchronization and buffering. However, caching location of the packets, cache hit ratio, and queuing delay become important. Regarding these challenging issues, two new caching strategies and a new cache replacement algorithm were proposed and scheduling methods were also applied to the CCN routers. In addition to analytic evaluation, some simulations were conducted. Simulation results showed that increasing the links rates, using the proposed caching strategies and cache replacement algorithm, and applying an appropriate scheduling method decrease the zapping delay significantly. Furthermore, bandwidth utilization and video quality are not sacrificed. Considering the effect of routing algorithms on the proposed channel zapping protocol is suggested as the future work.

## References

1. Abdullahi I, Arif S, Hassan S (2015) Survey on caching approaches in information centric networking. *J Netw Comput Appl* 56:48–59
2. Ahmad MZ, Qadir J, Rehman N, Baig A, and Majeed H (2009) Prediction-based channel zapping latency reduction techniques for IPTV systems—a survey, *International conference on, emerging technologies*, pp. 466–470.

3. Amadeo M, Molinaro A, Ruggeri G (2013) E-CHANET: routing, forwarding and transport in information-centric multihop wireless networks. *Comput Commun* 36:792–803
4. Ardissono L, Gena C, Torasso, P, Bellifemine F, Difino A, and Negro B (2004) *User modeling and recommendation techniques for personalized electronic program guides*: Springer
5. Arumathurai M, Chen J, Monticelli E, Fu X, and Ramakrishnan KK, (2014), Exploiting ICN for flexible management of software-defined networks, in *Proceedings of the 1st international conference on Information-centric networking*, pp 107–116
6. Named Data Networking (11/05/2016) Available: <http://www.named-data.net>
7. Begen AC, Glazebrook N, and Steeg WV (2009) A unified approach for repairing packet loss and accelerating channel changes in multicast IPTV. In *6th IEEE consumer communications and networking conference (CCNC)*, pp 1–6
8. Bejerano Y and Koppol PV (2009) Improving zap response time for IPTV. in *INFOCOM, IEEE*, pp. 1971–1979.
9. CCNx project, <http://blogs.parc.com/ccnx/>. Accessed October 2016
10. Cho C, Han I, Jun Y, and Lee H, (2004) Improvement of channel zapping time in IPTV services using the adjacent groups join-leave method. *The 6th international conference on, advanced communication technology*, pp. 971–975
11. Ciancaglini V, Piro G, Loti R, Grieco LA, and Liquori L (2013) CCN-TV: a data-centric approach to real-time video services, *27th international conference on, advanced information networking and applications workshops (WAINA)*, pp. 982–989
12. CONVERGENCE project, <http://www.ict-convergence.eu>. Accessed October 2016
13. Dannewitz C, Kutscher D, Ohlman B, Farrell S, Ahlgren B, Karl H (2013) Network of information (NetInf) - an information-centric networking architecture. *Comput Commun* 36:721–735
14. Detti A, Melazzi NB, Salsano S, Pomposini M (2011) CONET: a content centric inter-networking architecture. In: *Proceedings of the ACM SIGCOMM workshop on Information-centric networking Toronto*. ACM, Ontario, pp. 50–55
15. Detti A, Ricci B, and Blefari-Melazzi N (2013) Peer-to-peer live adaptive video streaming for Information Centric cellular networks. *IEEE 24th international symposium on, personal indoor and mobile radio communications (PIMRC)*, pp. 3583–3588.
16. Eftekhari S, Ghahfarokhi BS, and Moghim N (2015) Otones: a live tv simulator for content-centric networking. *7th conference on, information and knowledge technology (IKT)*, pp 1–6
17. Fuchs HF (2008) Optimizing channel change time in IPTV applications. *2008 I.E. international symposium on, broadband multimedia systems and broadcasting*, pp 1–8
18. Han B, Wang X, Choi N, Kwon T, Choi Y (2013) AMVS-NDN: Adaptive mobile video streaming and sharing in wireless named data networking. *2013 I.E. conference on, computer communications workshops (INFOCOM WKSHPs)*. Turin, pp.:375–380
19. <https://www.gartner.com/doc/3100229/hype-cycle-networking-communications->. Accessed July (2016)
20. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, and Braynard, RL (2009) Networking named content. In: *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pp 1–12
21. Kalman M, Steinbach E, Girod B (2004) Adaptive media playout for low-delay video streaming over error-prone channels. *IEEE Trans Circuits Syst Video Technol* 14:841–851
22. Kim Y, Park JK, Choi HJ, Lee S, Park H, Kim J, Lee Z, and Ko K (2008) Reducing IPTV channel zapping time based on viewer’s surfing behavior and preference. *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pp 1–6
23. EvalVid - a video quality evaluation tool-set, <http://www2.tkn.tu-berlin.de/research/evalvid/fw.html>. Accessed October 2016
24. Kooij RE, Kamal A, and Brunnström K (2006) Perceived quality of channel zapping, in *communication systems and networks*, pp 156–159
25. Koponen T, Chawla M, Chun BG, Ermolinskiy A, Kim KH, Shenker S, and Stoica I (2007) A data-oriented (and beyond) network architecture. *ACM SIGCOMM computer communication review*, pp 181–192
26. Kyryk M, Pleskanka N, and Silyuchenko M (2012) Reducing channel zapping time based on predictive tuning method. In *proceedings of international conference on modern problem of radio engineering, Telecommunications and Computer Science*
27. Lederer S, Mueller C, Rainer B, Timmerer C, and Hellwagner H (2013) Adaptive streaming over content centric networks in mobile networks using multiple links. *2013 I.E. international conference on communications workshops (ICC)*, pp 677–681
28. Lee J, Lee G, Seok S, and Chung B (2007) Advanced scheme to reduce IPTV channel zapping time, in *Managing Next Generation Networks and Services*: Springer, pp. 235–243.
29. Lee Y, Lee J, Kim I, Shin H (2008) Reducing IPTV channel switching time using H. 264 scalable video coding. *IEEE Trans Consum Electron* 54:912–919

30. Lee CY, Hong CK, Lee KY (2010) Reducing channel zapping time in iptv based on user's channel selection behaviors. *IEEE Trans Broadcast* 56:321–330
31. Lee S, Moon H, Bahn H, Kim T, Kim I-S (2011) Popularity and adjacency based prefetching for efficient IPTV channel navigation. *IEEE Trans Consum Electron* 57:1135–1140
32. Li Z and Simon G (2011) Time-shifted TV in content centric networks: the case for cooperative in-network caching. *IEEE international conference on, communications (ICC)*, pp 1–6
33. Liu Y, Geurts J, Point JC, Lederer S, Rainer B, Muller C, Timmerer C, and Hellwagner H (2013) Dynamic adaptive streaming over CCN: a caching and overhead analysis. *IEEE international conference on, communications (ICC)*, pp. 3629–3633
34. Mandal SK and MBuru M (2008) Intelligent pre-fetching to reduce channel switching delay in IPTV systems, Dept. of Comp. Sci., Texas A & M University, tech. rep
35. SAIL Project (11/05/2016) Available: <http://www.sail-project.eu>
36. PURSUIT Project (2013, 11/05/2016) Available: <http://www.fp7-pursuit.eu/PursuitWeb>
37. Oh U, Lim S, Bahn H (2010) Channel reordering and prefetching schemes for efficient IPTV channel navigation. *IEEE Trans Consum Electron* 56:483–487
38. Qiu T, Ge Z, Lee S, Wang J, Xu J, and Zhao Q (2009) Modeling user activities in a large IPTV system. *Proceedings of the 9th ACM SIGCOMM conference on internet measurement conference*, pp. 430–441.
39. Rainer B, Posch D, Hellwagner H (2016) Investigating the performance of pull-based dynamic adaptive streaming in NDN. *IEEE Journal on Selected Areas in Communications* PP(99):1
40. Ramos FM (2013) Mitigating IPTV zapping delay. *IEEE Commun Mag* 51:128–133
41. Ramos FM, Crowcroft J, Gibbens RJ, Rodriguez P, and White IH, (2010) Channel smurfing: minimising channel switching delay in IPTV distribution networks. *Multimedia and expo (ICME), IEEE international conference*, pp 1327–1332
42. Ramos F, Crowcroft J, Gibbens RJ, Rodriguez P, White IH (2011) Reducing channel change delay in IPTV by predictive pre-joining of TV channels. *Signal Process Image Commun* 26:400–412
43. Siebert P, Van Caenegem TN, Wagner M (2009) Analysis and improvements of zapping times in IPTV systems. *IEEE Trans Broadcast* 55:407–418
44. Sun W, Lin K, Guan Y (2008) Performance analysis of a finite duration multichannel delivery method in IPTV. *IEEE Trans Broadcast* 54:419–429
45. The Abilene Network (2002) University of Texas System
46. Tian X, Cheng Y, Shen X (2013) Fast channel zapping with destination-oriented multicast for IP video delivery. *IEEE Trans Parallel Distrib Syst* 24:327–341
47. TRIAD.(4/13/2015) Available: <http://gregorio.stanford.edu/triad>
48. Xylomenos G, Ververidis C, Siris V, Fotiou N, Tsilopoulos C, Vasilakos X, Katsaros K, Polyzos G (2013) A survey of information-centric networking research. *IEEE Commun Surv Tutor* 16:1024–1049



**Behrouz Shahgholi Ghahfarokhi** received his B.Sc. in Computer Engineering (2004), his M.S. in Artificial Intelligence (2006), and his Ph.D. in Computer Architecture (2011) from University of Isfahan. He joined to University of Isfahan in 2011 and he is now assistant professor at Faculty of Computer Engineering. His research interests are computer networks, mobile communications, and intelligent systems.





**Neda Moghim** received the B.S. and M.S. degrees both from Isfahan University of Technology, Iran, Isfahan in 1999 and 2002 respectively and the Ph.D. from Amirkabir University of Technology, Iran, Tehran in 2009. She is the author of several technical papers in telecommunications journals and conferences. Currently she is an assistant professor with the Department of Information Technology Engineering, University of Isfahan, Iran. Her research interests are in the area of admission control, bandwidth management and traffic engineering for QoS-enabled IP networks, next generation networks, and wireless mobile/fixed networks.



**Shayan Eftekhari** is an experienced software architecture. He received his B.Sc. in Software Engineering from IAUN (2008), and his M.Sc. degree in Information Technology Engineering from the University of Isfahan (2015). His current research interests include information centric networking, multimedia communications, and distributed systems. From 2007 to 2008 he was a research assistant at the Information and Communication Technology Institute (ICTI). Since then, he is with PayamPardaz Company as software architecture.