Proceedings of 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN 2018)

Coupling DAS, SVC and NDN: an SVC-aware cache and forwarding policy for NDN routers

Wei Liu, Guoqiang Zhang, and Qian Gao
School of Computer Science and Technology, Nanjing Normal University
Email: {1300371834@qq.com; guoqiang@ict.ac.cn; gaoqian@njnu.edu.cn}

***Abstract*—Video streaming traffic has grown sharply in recent years. *Dynamic Adaptive Streaming* (DAS) is widely used in video streaming because of its ability to cope with the network dynamics and terminal heterogeneity. Compared with AVC(advanced video coding), SVC(scalable video coding) can better adapt to network conditions, reduce server storage and improve users' playback QoE. However, the present TCP/IP network is communication-oriented, focusing on addressing rather than content, which can not fully realize the potential of DAS and SVC. *Named Data Networking* (NDN) is a new network architecture that centers around content. Its name based routing, in-network caching, hop-by-hop forwarding and built-in multicast support naturally fits the pattern of DAS and SVC. As the last piece of jigsaw puzzle, coupled with DAS and SVC, NDN is believed to be able to provide better video streaming performance. In this paper, we proposed a customized SVC-aware cache and forwarding policy for ordinary NDN routers to support video streaming applications. Simulation results based on *ndnSIM2.3* showed that the proposed policy can improve the average playback bitrate and reduce the video stall frequency as well as duration.**

***Index Terms*—Dynamic Adaptive Streaming, Scalable Video Coding, Named Data Networking, Forwarding, Caching.**

I. Introduction

In recent years, people's demand on high-quality video increases significantly. According to Cisco VNI report [1], Internet video traffic will account for 82% of the total Internet traffic. To offer better video playback experience, *Dynamic Adaptive Streaming* was proposed to deal with the dynamics of network and the heterogeneity of networked devices. DAS replaces traditional push-based video streaming with pull-based streaming, which allows users to dynamically adapt to network changes based on their sense of network conditions. Various proprietary products are developed, e.g., apple's HLS, Adobe's HDS and Microsoft's MMS. To offer an interoperable solution, MPEG-DASH was standardized by MPEG [2]. In MPEG-DASH, the video to be streamed is encoded in different bitrates and segmented into small chunks. Each chunk has a very short duration, e.g., 2 seconds. A client first retrieves the MPD(*Media Presentation Description*) file which describes all the available presentations of the given video, including bitrates, duration, and URLs. The client can then download appropriate chunks based on its current network condition. MPEG-DASH, however, does not regulate how the client-side schedule should be performed. Traditional DAS uses advanced video coding(AVC), which encodes the video into various independent versions of bitrates. The independence among these encoded videos poses challenges for both the server and intermediate caches. The server has to store all these video versions. Also, since the chunks from different bitrate versions are independent, to achieve high cache hit ratio, the cache has to store different versions of the same chunk. The independence between different version chunks also exacerbates the bitrate oscillation problem. Scalable Video Coding (SVC) [3] can partly solve these problems by encoding the video into one base layer(layer 0) and several enhancement layers(layer 1, 2, etc). Each enhancement layer can then be used together with lower layers to enhance the video playback quality. Compared with AVC, SVC allows the server to have much less storage. Also, SVC can improve the cache utilization, mitigate bitrate oscillation and reduce client-side video stalls.

However, the present TCP/IP network is host-centric, which poses a lot of obstacles to fully utilize the potential of DAS and SVC. Named Data Networking [4] is a new network architecture that centers around content. Its name based routing, in-network caching, hop-by-hop forwarding and built-in multicast support naturally fits the pattern of DAS and SVC. As the last piece of jigsaw puzzle, coupled with DAS and SVC, we believe NDN is able to provide better video streaming performance. For example, instead of relying on client-side bandwidth prediction, NDN's hop-by-hop forwarding allows the network to provide explicit feedback to the client, which in turn can make more accurate adjustment. And, since each layer depends on the base layer to decode, the cache policy in NDN routers can prioritize the caching of base layer chunks. In this paper, we proposed a customized SVC-aware cache and forwarding policy for ordinary NDN routers to support video streaming applications. Particularly, we first proposed a *probabilistic caching policy* (PC) that assign a cache probability to each chunk based on its layers. A lower-layer chunk is more likely to be cached. We then proposed a *buffer and reorder forwarding policy* that prioritizes the forwarding of lower SVC data chunks. We evaluated its performance using *ndnSIM2.3*. Results show that our proposed policy can improve the average playback bitrate and reduce the video stall frequency as well as duration.

II. RELATED WORK

A. DAS and SVC

MPEG-DASH [2] regulates the interaction between DASH client and DASH server, however, it leaves the adaption logic for different vendors. The adaption logic can be broadly classified into two classes: bandwidth based [5]-[10] and buffer based [11]. Bandwidth based approaches predicts the currently available bandwidth of the network, and based on the prediction to decide which video quality chunks to fetch. The bitrate of the fetched video chunk should not exceed the predicted bandwidth. The downside of this approach is frequent network dynamics and the existence of cache could lead to frequent bitrate switches and even playback stalls [12]-[14]. Buffer-based approaches intend to stabilize the buffering level of the player to avoid playback stalls. They make the decision based on the player's buffering level. Usually, these adaption algorithms classified the buffering length into several levels, and fetch predefined video quality according to the present buffering level. These algorithms, however, cannot adapt to network dynamics in time. Present researches proposed several hybrid approaches, intending to accommodate the advantages of these two classes of algorithms. Spiteri et al.[15] formulated bitrate adaptation as a utility maximization problem and devised an online control algorithm called BOLA that uses Lyapunov optimization techniques to minimize rebuffering and maximize video quality. Some researches reveal that SVC can augment the adaption logic [16], [17]. In fact, SVC encoding introduced two dimensions for the adaption logic: the time dimension, and the layer dimension. DASH with SVC can significantly reduce the server side storage burden, and can easily cope with diverse terminals. Compared with AVC, it is demonstrated that SVC can further improve users' QoE, reduce client-side buffer, better adapt to network dynamics and achieve high cache utilization.

B. *DASH over CCN/NDN*

DASH and CCN/NDN have many common grounds, e.g., receiver-driven transmission mode, independent names for each chunked content, and content cache[18]. However, DASH is working on the application layer, whose underlying host-centric TCP/IP network prevents it from fully realizing the potential of DASH. In contrast, CCN/NDN naturally supports the idea of DASH. For this reason, applying of the DASH idea into CCN/NDN receives attention. It was demonstrated for the first time that DASH over CCN can improve system throughput, but with large overhead [19]. It suggested that the traditional end-to-end adaptive logic should be adjusted to fit the CCN/NDN's hop-by-hop transmission. INA(integration of innetwork adaptation) was proposed [20] to tackle inefficiencies caused by DASH in ICN using innetwork adaptation. IRTF's ICN research group made a comprehensive survey on the feasibility and possible problems of applying DASH over ICN [21]. It concluded that by replacing MPED-DASH's transport from HTTP to CCN/NDN, utilizing CCN/NDN's natural support for mobility and multiple interfaces, it is possible for clients to achieve seamless switch among multiple networks and interfaces. This work further discussed the potential of applying SVC on NDN to reduce the burden on caches at NDN routers. However, CCN/NDN's specific features may also affect the effectiveness of DASH's adaptive logic, especially the accuracy of bandwidth predication. Pervasive caching may easily lead to bandwidth overestimate. CCN/NDN's multi-source may cause the oscillation of TTL, indirectly affecting the bandwidth estimate [22]. And because of PIT, it is possible that the Data is already on its way back when an Interest is sent, which leads to underestimate of TTL [23]. Purely bandwidth prediction based adaption policy can lead to bitrate oscillation, and break cache friendliness. SVC is believed to mitigate this issue. Ref. [24], [25] proposed to apply adaptive forwarding inside the network based on SVC, which alleviated network congestion and improved clients' quality of experience.

III. OUR APPROACH

Our approach couples DAS, SVC and NDN. We focus on the routers'behavior, and use a simple client-side adaption logic to evaluate our inside network policies. Particularly, our motivation comes from the special feature of SVC encoding. Since the base layer is shared by all users and is more important than enhanced layers in this sense, we can customize the content router to prioritize the caching of base layers. Furthermore, since chunks are not equitable, we changed the FIFO behavior to prioritizing the forwarding of base layer chunks.

A. *Probabilistic caching*

Cache Everything (CE) is a common caching policy in CCN/NDN. However, since in SVC, enhancement layers depend on base layers for decoding, we give base layers more opportunity to be cached. Specifically, we applied the probabilistic caching (PC). Assume there are k layers L_0, L_1, , L_k, (L_0 is the base layer) , we assign each layer a cache probability P_0, P_1, . . . , P_k *with* $P_0 > P_1 > \ldots > P_k$. When the cache is full, we use *Least Recently Used* (LRU) to replace the chunks, regardless of the layer of the chunks. Hopefully, giving lower layer chunks higher probability to be cached, it is more easy for users to fetch base layers inside the network, hence reducing possible playback stalls at clients.

B. *Buffer and reorder based forwarding*

FIFO is the default policy to forward packets at an interface. However, with SVC encoding, equally treating each chunk is not appropriate. For example, if a base layer chunk and an enhancement layer chunk are both to be forwarded with similar playback deadline, it is more reasonable to forward the base layer chunk before the enhancement layer chunk.

We still use one queue to queue the packets to be forwarded. However, we made small changes to the FIFO behavior. We buffer and reorder the packets to be forwarded in a batch. Initially, we buffer the first packet arrived and set $n=1$, while n indicates the number of packets in the buffer. Let κ denotes the threshold of packets to be reordered. Then, we reorder the first $u=min\{\kappa,n\}$ packets and put these u packets into the forwarding queue. After these u packets have been forwarded, we repeat the above step, i.e, choose another $min\{\kappa,n\}$ packets, reorder them and put them into the forwarding queue. With this policy, packets within a batch may be reordered due to its layer property. But packets falling into later batches will not preempt the forwarding of previous batches. Within a packet batch, the reorder rules are as follows: Interest packets are forwarded

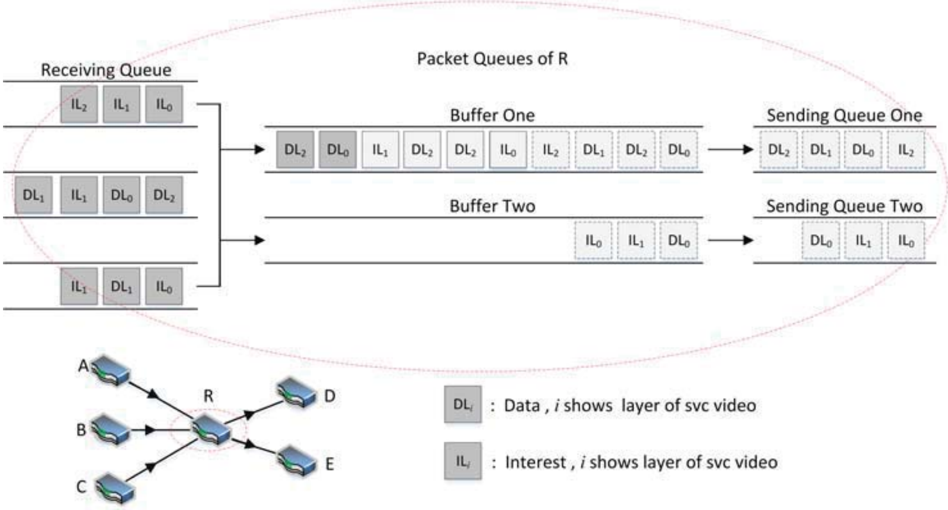


Fig. 1: Example queues of router R with BRF policy

before Data packets, and lower layer Interest(or Data) packets are forwarded before higher layer Interest(or Data) packets. We call this *buffer and reorder based forwarding* as BRF.

Figure 1 illustrates how the BRF policy works. First of all, router R receives packets in sequence and puts them into Buffer one and Buffer two respectively according to the FIB lookup results. Then, the packets are grouped into batches. We assume $\kappa=4$ in our example. For buffer one, $u=min\{4,10\}=4$. Hence, DL_0, DL_2, DL_1 and IL_2 are in the same batch and will be reordered to IL_2, DL_0, DL_1, DL_2 before being put into the sending queue. After which, IL_1, DL_2, DL_1 and IL_0 are in the following batch and be reordered to IL_0, IL_1, DL_1, DL_2 before being put into the sending queue. For buffer two, $u=min\{4,3\}=3$, so all the three packets are in the same batch.

IV. EVALUATION

In order to evaluate the performance of probabilistic caching and BRF proposed in Section II-A and II-B, we use *ndnSIM 2.3* which is based on ns-3 to run the simulations. Firstly, we outline the evaluation set-up. Secondly, we show the results and compare them to the results of a commonly used strategy CE+FIFO, i.e., caching the chunk everywhere on path and forwarding packets according to *first-in-first-out* (FIFO).

A. Evaluation Setup

For the content of test, we use MPEG-DASH-compliant SVC-encoded video with a segment size of two seconds. The video content is taken from the SVC dataset [26] with duration of about 10 minutes. We extend it to 20 minutes by copying itself. The chosen video is encoded into a base layer and two enhancement layers. The base layer (L_0) has an average bitrate of approx. 798 kbps. The first enhancement layer (L_1) has a bitrate of approx. 438 kbps. One has to fetch the same segment of L_0 and L_1(L_0 + L_1≈1236 kbps) so that it can playback a segment at the quality of L_1. The second enhancement layer (L_2) has an average bitrate of approx. 526 kbps. Obviously, the highest-quality video has an average bitrate of L_0 + L_1 + L_2≈1862 kbps). For PC policy, we set the probabilities for caching each layer as $P_0=1$, $P_1=0.9$ and $P_2=0.8$, which can highly utilize router's cache ability. Additionally, we use LRU replacement strategy for both PC and CE. For BRF forwarding policy in Section II-B, we set κ(the threshold of packets to be reordered) equal to 8 because it is proved by experiments that Interest will expire before corresponding Data returns if κ is too big.

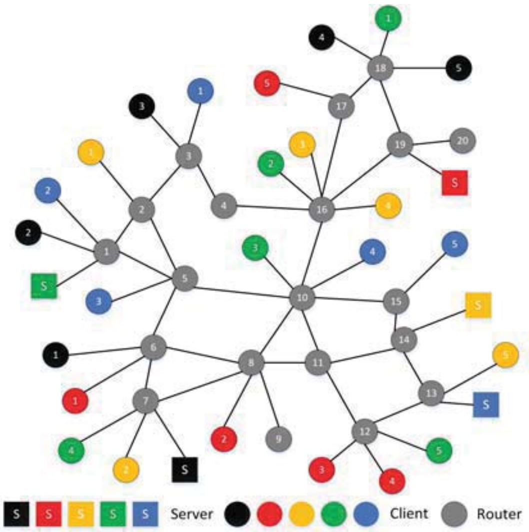
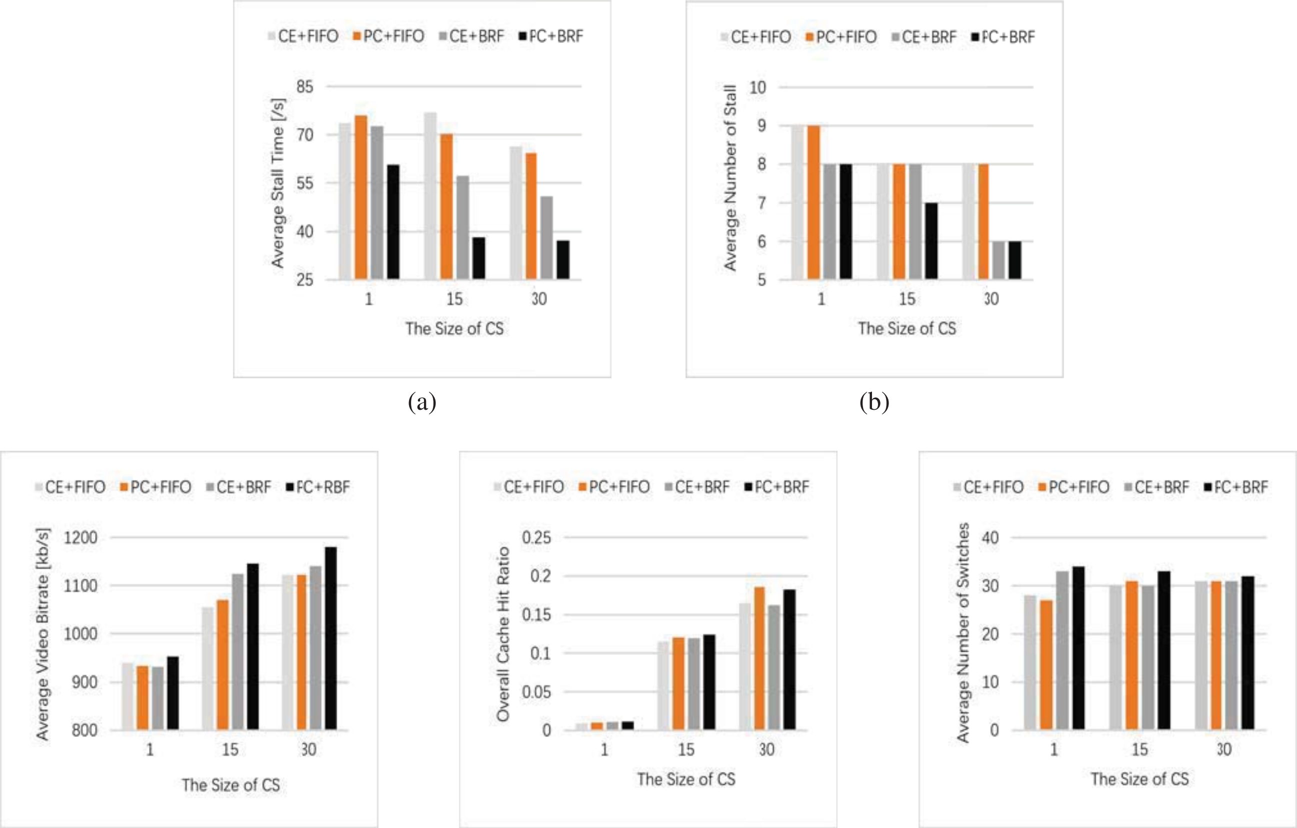


Fig. 2: Topology

The network topology is shown in Figure 2 for the evaluation in order to evaluate the performance of different caching policies and forwarding policies. There are 25 clients and 5 servers with colors in the network. Clients with the same color request the same video from the corresponding server. For example, yellow clients all request video content from yellow server. Therefore, clients and servers are divided into five groups, denoted by the colors green, red, blue, black and orange(cf. Figure 2). Packet



(a) (b)

(c) (d) (e)

Fig. 3: Main indexes of DAS in NDN with different CS storage space.

name consists of name segments. Each name segment carries a label representing something, and a value[27]. For the base layer of the first red-video block, we name it ccnx:/TYPE=video/ENCODING=svc/NAME=red/LAYER=0/ seq=%FE%00. We can easily detect SVC traffic and SVC-packet's layer from the content name. ENCODING=svc represents SVC traffic, while LAYER=0 represents base layer of SVC-video.

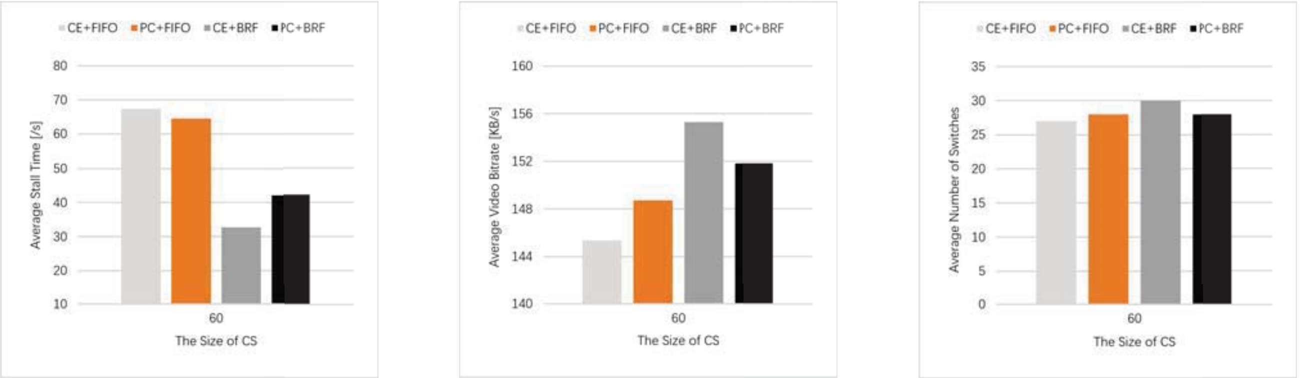
We set the start time of clients from an exponential distribution(*mean=5s*), because it is similar to scenarios in real life. The links between routers and routers have a bandwidth of 6 Mbps (bidirectional). The links connecting the servers to routers have a bandwidth of 8 Mbps (bidirectional). The network links connecting the clients to routers have a bandwidth of 3 Mbps (bidirectional). Network congestion will occur after clients start in the presented topology with the given settings. We only consider real-time playing and all clients play once with no forward and backward functions. Thus, clients possess a small playback buffer which is capable of storing eight seconds of video content. Requests from client applications are sent based on a constant frequency model. We repeat the following two tests separately ten times and average the results.

B. Results

In the first test, the CS size is set 1, 15, 30 respectively, meaning the number of Data packets can be cached. Figure 3(a) and 3(b) depict the average stall time and average number of stalls at clients with different policies: CE+FIFO, PC+FIFO, CE+BRF and PC+BRF. Compared with CE+FIFO, approx. 36% of stall time and approx. 16% of number of stall are reduced using PC+BRF because routers can differentiate lower-layer and higher-layer packets of SVC video and prioritize the caching and forwarding accordingly. For PC+FIFO and CE+BRF, they perform worse than PC+BRF either.

Fig. 3(c) shows the average video playback bitrate achieved at the clients. It can be observed that our approach PC+BRF achieves 5% average bitrate improvement over CE+FIFO, 3% over PC+FIFO and CE+BRF. The reason is that all clients with low bandwidth should get base layer packets of SVC video. Since all clients share the base layer, prioritizing caching and forwarding of the base layer chunks benefits the clients as a whole. Fig. 3(d) shows that PC+FIFO and PC+BRF perform better than CE+FIFO and CE+BRF in the overall cache ratio, because more low-layer Data packets are likely to be hit in CS while using PC strategy.

However, Fig. 3(e) shows that using PC+BRF has the biggest number of switches in the four policies, which increases approx. 11% compared to CE+FIFO, and 11% to PC+FIFO, and 8% to CE+BRF. The number of switches are used to measure the bitrate variances during the video playback. A small number of quality switches will increase the *Quality of Experience* (QoE). The reason of higher number of switches may be due to our simple client-side adaption logic. Since



(a) (b) (c)

Fig. 4: Main indexes of DAS in NDN and the size of CS is 60.

we focus our study on the in-network caching and forwarding behavior, we take a very simple rate adaption logic at the client as follows. The client always measures the available bandwidth while it downloads a segment. Based on the estimated bandwidth, clients will choose to download layer-packets of the subsequent segment as high as possible. We didn't take playback stability into account. This simple logic at the client side may cause high bitrate switches. Requests can be served either at servers, at intermediate caches or router reorder queue. When requests are served at caches and router reorder queue, it introduces highly dynamic RTTs and thus high variation in the bandwidth estimate. Because PC+BRF has higher cache hit ratio and reorder queue of BRF makes RTTs fluctuate violently, our simple adaption logic could result in high bitrate switches. There is another reason. PC+BRF has the least stall time which negatively correlates with bitrate switches. The number of switches decreases slightly when the CS size gets larger, as is shown in Fig. 3(e). These results show that to fully take the power of in-network SVC awareness, the client-side adaption logic should be adjusted as well, which will be part of our future work.

In the second test, we set the CS size to 60. Fig. 4(a) and 4(b) show that regarding average stall time and average playback bitrate, CE+FIFO performs worst, while CE+BRF performs best. That is to say, CE+BRF will make clients obtain higher playback bitrate and playback more smoothly than other policies when routers are not short of CS storage space. This is reasonable because when the CS size is big, caching everywhere can reduce the RTTs and reduce the bandwidth predication inaccuracies. As shown in Fig. 4(c), while using CE+BRF, the average number of switches is bigger than using other policies, however, it is worthy to trade the small increase of switch times to higher playback bitrate and less stall time.

V. Conclusion

The goal of this paper is to improve the SVC video performance in NDN relying on the principles of DAS and the benefits of SVC. We proposed *probabilistic caching policy*(PC) and *BRF forwarding policy*, which prioritizes the caching and forwarding of base layer chunks inside the network. Simulation results based on *ndnSIM2.3* show that our approaches can increase the average playback bitrate and reduce the video stall frequency as well as duration regardless of the cache size. However, the number of video switches has a small increase compared to CE+FIFO. Improved client-side adaption logic should be studied in the future to take full power of in-network SVC awareness.

VI. Acknowledgement

This work was partly supported by the National Natural Science Foundation of China under Grant 61772279, 61602114, and 61370227.

References

[1] "Cisco visual networking index: Forecast and methodology, 2016-2021, cisco white paper," p. 3, 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf

[2] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, April 2011.

[3] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h.264/avc standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sept 2007.

[4] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2656877.2656887

[5] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http," *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 271–287, 2012.

[6] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive http streaming," in *ACM Conference on Multimedia Systems*, 2011, pp. 169–174.

[7] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "Qoe-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over http," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 451–465, March 2015.

[8] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM '15. New York, NY, USA: ACM, 2015, pp. 325–338.

[9] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, "Probe and adapt: Rate adaptation for http video streaming at scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719–733, April 2014.

[10] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 326–340, Feb 2014.

[11] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, ser. SIGCOMM '14. New York, NY, USA: ACM, 2014, pp. 187–198.

[12] C. Sieber, T. Hoβfeld, T. Zinner, P. Tran-Gia, and C. Timmerer, "Implementation and user-centric comparison of a novel adaptation logic for dash with svc," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, May 2013, pp. 1318–1323.

[13] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in http adaptive streaming," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2386–2399, Aug 2016.

[14] D. H. Lee, C. Dovrolis, and A. C. Begen, "Caching in http adaptive streaming: Friend or foe?" pp. 31:31–31:36, 2014. [Online]. Available: http://doi.acm.org/10.1145/2578260.2578270

[15] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: near-optimal bitrate adaptation for online videos," *In INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9 2016.

[16] R. Huysegems, B. De Vleeschauwer, T. Wu, and W. Van Leekwijck, "Svc-based http adaptive streaming," *Bell Lab. Tech. J.*, vol. 16, no. 4, pp. 25–41, Mar. 2012.

[17] T. Andelin, V. Chetty, D. Harbaugh, S. Warnick, and D. Zappala, "Quality selection for dynamic adaptive streaming over http with scalable video coding," in *ACM Sigmm Conference on Multimedia Systems, Mmsys 2012, Chapel Hill, Nc, Usa, February*, 2012, pp. 149–154.

[18] J. Samain, G. Carofiglio, L. Muscariello, M. Papalini, M. Sardara, M. Tortelli, and D. Rossi, "Dynamic adaptive video streaming: Towards a systematic comparison of icn and tcp/ip," *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2166–2181, Oct 2017.

[19] Y. Liu, J. Geurts, J. C. Point, S. Lederer, B. Rainer, C. Mller, C. Timmerer, and H. Hellwagner, "Dynamic adaptive streaming over ccn: A caching and overhead analysis," in *2013 IEEE International Conference on Communications (ICC)*, June 2013, pp. 3629–3633.

[20] D. Posch, C. Kreuzberger, B. Rainer, and H. Hellwagner, "Using in-network adaptation to tackle inefficiencies caused by dash in information-centric networks," pp. 25–30, 2014. [Online]. Available: http://doi.acm.org/10.1145/2676652.2676653

[21] E. C. Westphal, S. Lederer, D. Posch, C. Timmerer, A. Azgin, W. Liu, C. Mueller, A. Detti, D. Corujo, and J. Wang, "Adaptive video streaming over information-centric networking (icn)," 2016.

[22] H. B. Abraham and P. Crowley, "Forwarding strategies for applications in named data networking," in *2016 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, March 2016, pp. 111–112.

[23] C. Yi, C. Yi, B. Zhang, and L. Zhang, "A practical congestion control scheme for named data networking," in *ACM Conference on Information-Centric NETWORKING*, 2016, pp. 21–30.

[24] Z. Liu and Y. Wei, "Hop-by-hop adaptive video streaming in content centric network," in *IEEE International Conference on Communications*, 2016, pp. 1–7.

[25] S. Lederer, C. Mueller, B. Rainer, C. Timmerer, and H. Hellwagner, "Adaptive streaming over content centric networks in mobile networks using multiple links," in *2013 IEEE International Conference on Communications Workshops (ICC)*, June 2013, pp. 677–681.

[26] C. Kreuzberger, D. Posch, and H. Hellwagner, "A scalable video coding dataset and toolchain for dynamic adaptive streaming over http," in *Proceedings of the 6th ACM Multimedia Systems Conference*, ser. MMSys '15. New York, NY, USA: ACM, 2015, pp. 213–218. [Online]. Video file vailable: http://concert.itec.aau.at/SVCDataset/dataset/BBB/II/segs/720p

[27] " CCNx Semantics draft-irtf-icnrg-ccnxsemantics-07," https://www.ietf.org/mail-archive/web/icnrg/current/msg02421.html