On the Performance of Adaptive Video Caching over Information-Centric Networks

Wenjie Li, Sharief M.A. Oteafy, Hossam S. Hassanein
School of Computing
Queen's University, Kingston, ON, Canada
Email: {liwenjie, oteafy, hossam}@cs.queensu.ca

***Abstract*—The growing demand for video streaming is straining the Internet, and mandating a fundamental change in future networking paradigms. Current advancements in Information-centric Networks (ICN) promise a novel approach to intrinsically handling content dissemination, caching and retrieval. While streaming technologies are converging towards Dynamic Adaptive Streaming (DAS), in-network caching in ICN facilitates serving users with better video qualities, potentially beyond their actual bandwidth-mandated throughput. In this paper, we propose an assessment framework for adaptive video streaming to evaluate the performance of ICN caching schemes, and measure their impact on improving users' streaming experience. We present a thorough study of core performance metrics and adopt those metrics which are designed for video caching. We conduct experiments on a NS-3 based simulator, ndnSIM, and propose insights which will aid the development of future caching schemes that cater to inevitable bitrate variations.**

***Index Terms*—In-network caching; ICN; dynamic adaptive streaming; performance analysis.**

I. Introduction

The growing demand for video content is reshaping our view of the current Internet. According to Cisco's Visual Networking Index (VNI), by 2020, it is projected that global Internet traffic would surpass 162 exabytes per month, where video traffic would amount to a dominating 80% in conservative estimates [1]. This dramatic growth in video demand has motivated service providers (*e.g.,* Netflix and YouTube) to adopt HTTP-based dynamic adaptive streaming (DASH) [2]. This protocol attempts to serve video users with the best possible bitrate under time-varying bandwidth conditions, in order to satisfy their increasing expectations on video quality.

DASH has played a major role in improving video delivery services. However, DASH relies on HTTP connection protocol which is implemented over the current host-centric network. Ultimately, service degradation and inevitable downtime will dominate unless scalable networking primitives are built on contents rather than connections. A recent shift towards Information-Centric Networks (ICN) [3] is promising a novel approach to intrinsically handling large content dissemination, caching and retrieval. Since video delivery services are projected to remain influential as we steer to ICNs, dynamic adaptive streaming must be addressed as an essential component in this future Internet [4].

The architecture of ICN offers the unique premise of ubiquitous in-network caching, which is recognized as an efficient way to reduce access delay. In the context of adaptive video streaming, the effect of in-network caching is even more important since it provides an opportunity to serve users with bitrates higher than the actual link bandwidth between the producer and consumer [5]. This means that high-resolution video content, which may be impossible to deliver without caching, will effectively become accessible by users.

Many research efforts have investigated ICN caching [6], both new and re-engineered ones (*e.g.*, from Content Distribution Networks (CDNs)). However, catering to video caching in ICNs presents notable challenges. For example, since video players inherently request data sequentially, they render most of these schemes inefficient as they disregard data order. Thus, we cannot rely on generic caching strategies that compare content popularity or request rates, regardless of the content itself. Moreover, in-network caching could result in significantly varying throughput of video segments. As adaptive bitrate control depends on measurements of throughput to evaluate the current network condition, and makes the decision on the next segment, in-network caching could, in fact, worsen bitrate oscillation. Thus, it is crucial to design a caching scheme which caters to adaptive video bitrate control.

Our previous work [7] targeted minimizing the access delay per bit for video contents under steady-state video requests for variable bitrates in order to improve the average throughput experienced by users. However, as adaptation control logic is usually built upon heuristics (*e.g.,* [8], [9]), modelling the bitrate switches is already difficult, not to mention designing caching schemes that maximize users' Quality of Experience (QoE) in the real time. In order to fully understand the characteristics of adaptive streaming traffic, we argue that the evaluation of widely accepted caching schemes is an essential prerequisite, which would facilitate pinpointing the pros and cons of these schemes that affect the performance.

The contributions of this work are threefold. First, we develop an assessment framework which evaluates the performance of any caching scheme under the adaptive video streaming traffic in ICN. Secondly, we utilize metrics, which are the *Expected Bitrate*, *Chance of Video Freezing* and *Bitrate Oscillation*, as representatives of QoE, to quantize the performance of ICN in-network caching. These metrics differ from existing ones in the literature in that existing metrics are used to measure the streaming quality for individual users, while our proposed metrics represent the performance of a caching system, which is influenced by requests from all users over

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

time. Thirdly, we conduct extensive simulations, implementing and assessing different video caching schemes and scenarios. Through analyzing the reasons behind significant differences in users' QoE under these scenarios, we identify the core characteristics of adaptive video caching and present our insights, which will aid future scheme design.

II. Background

In dynamic adaptive streaming, video files are encoded in different bitrate levels by video producers. These files are chopped into segments with equal durations. On the user end, each video player would request video segments, under a chosen bitrate, based on its implementation of a rate adaptation control algorithm. This choice can change bitrates over time, in response to varying network conditions in order to view/download the video with the best possible quality [4].

Rate adaptation control logic makes bitrate choices primarily on the measurement of throughput over the most recent video segment. In the context of ICN, in-network caching would lead to oscillated throughput since the cache hit and cache miss of a video request would result in significantly different delays. Such fluctuated throughput might make the requested bitrates switch back and forth which degrades the users' QoE. Recent research tackled this problem by seeking to balance between bitrate switching efficiency and stability. For example, Jiang et al. [8] proposed *Delayed Update*, where the candidate bitrate can win the selection only if the combination of efficiency and stability score is less than the current bitrate.

Liu et al. [5] analyzed the improvement made by in-network caching on the quality of served video contents. However, their experiment setup is problematic which makes another motivation of our work. In their experiments, there is always only one user sending video request at any time. This setting will let each user occupy the entire link bandwidth which contradicts with the real scenario. More importantly, based on the experiment results, we infer the cache size is set too large so that there is no cache replacement occurred. This explains why the requests for highest bitrates dominates as experiment proceeds since the cache holds all previous requested contents and requests from subsequent users are usually satisfied by cache hits. Other research like [10], tested the streaming performance under the mobile scenario.

To the best of our knowledge, there is no systematic and standardized methodology for evaluating caching performance under adaptive streaming traffic in the literature. Our work is also the first attempt which presents the variations on users' QoE among video contents with different system loads and caching schemes.

III. Assessment Framework

Our proposed assessment framework is composed of four modules: *Adaptive Video Consumer*, *ICN*, *User Request Profile* and *Tracer*. Each module in this framework can be specified with an actual implementation or a simulated program.

As depicted in Fig. 1, the main module of this framework is *Adaptive Video Consumer*, which correlates with ICN module.

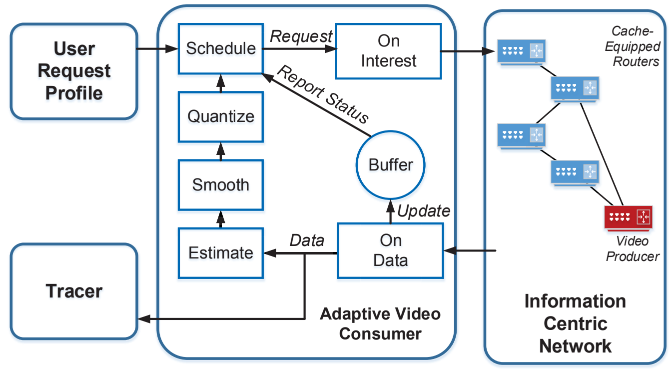


Fig. 1. Components of Adaptive Streaming Assessment Framework

The *OnInterest* component receives video requests from the adaptive control logic and translates them into identifiers which are recognized by the ICN naming system. The *OnData* component receives the video contents delivered from either the video producer or in-network caches and parses the identifiers to recover detailed indicators (meta-data) on video requests (*e.g.,* selected bitrate, filename, segment starting time). These indicators, along with delays in retrieving video segments are passed to the *Tracer* module for further analysis, which would be used to expose the performance of the caching schemes.

The adaptive control logic is triggered when the *OnData* component receives a video segment. Any control logic can be modelled with four phases [9]. The *Estimate* phase assesses the instant bandwidth by referring to the throughput of the last video segment. The *Smooth* phase then creates a filtered version of throughput in order to reduce the impact of throughput fluctuation caused by cache hits and misses. The *Quantize* phase is responsible for mapping this estimation of bandwidth to a discrete video bitrate. After these three phases, the adaptive rate control logic already selects a bitrate for the next video request. Lastly, the *Schedule* phase sets the time interval of passing this request, along with the decision on which video content to watch, to the *OnInterest* component.

The decision of selecting a video content is made by the *User Request Profile* module. If the assessment is performed over a testbed which involves real customers, it will not be necessary to implement this module. Otherwise, in order to mimic the behavior of users when they watch videos, a certain video content is picked based on its popularity.

To model the popularity of video files, the multimedia community largely adopts a *Zipf* distribution, where the probability of requesting the fth popular file $q_f \propto f^{-\alpha}$ [11]. The parameter α controls the skewness of popularity distribution. A larger α value indicates that fewer video files are frequently requested. The requested video file is randomly selected by a Zipf distribution and *User Request Profile* module further determines the playback duration. Since users typically start watching a video from the beginning and stop after a period of time, we propose that the total duration of video playback can be modelled as a geometric distribution, controlled by parameter p_f, where $(1-p_f)^{k-1}p_f$ indicates the probability that a user stops watching at the kth video segment. As a result, the video chunks with small IDs are more popular than chunks with larger IDs in the same file.

IV. Performance Evaluation

In this section, we elaborate on the experiment setup and the design of novel performance metrics introduced specifically to adaptive video caching literature, which aid future comparative analyses and experiments.

Our implementation is built on the Named Data Networking (NDN) architecture [12], a pioneering ICN infrastructure. Due to lack of space, we elaborate on two representative scenarios of these comparative experiments: 1) we evaluate the impact of system load on different cache replacement approaches, comparing the performance of **LFU** and **LRU** with a baseline placement scheme *Cache Everything Everywhere(CE2)*, and highlighting the variations of users' QoE among video files and chunks; 2) we assess a more recent caching placement scheme *ProbCache* [13] and our generic rate-selective approach *StreamCache* [7] across different cache sizes.

A. Simulation Setup

The simulated NDN environment is built over ndnSIM, a ns-3 based simulator, where each NDN router is equipped with a Content Store (CS) of the same size. The size of CS is measured in number of packets as a default setting in ndnSIM. However, to distinguish packets of video segments encoded with different bitrates, we modified this measure to a byte count. The varying capacity of *CS* is controlled by parameter ω, and is calculated as follows

$$CS = \frac{\sum \text{Size of Video Segment}}{\text{\# of NDN Routers}} \times \omega \quad (1)$$

To capitalize on recent advancements in bitrate adaptation algorithms, we implement the *Adaptive Video Consumer* module using the **FESTIVE** rate control logic [8]. FESTIVE is a highly cited approach, which accounts for users making simultaneous adaptive video requests, and aims to balance between bitrate switching efficiency and stability. We configure the observation window of FESTIVE to contain five segments, where video players attempt to maintain a steady-state buffer of 50 seconds.

We adopt four bit rates in our simulations: 250Kbps, 400Kbps, 600Kbps, and 900Kbps. To control variations across files, we set all video files to have the same length of playback time (300 seconds), then each file is chopped into equal-duration 10 second segments. To account for users' interests in video content, these files are assigned a request probability following the *Zipf* popularity distribution, as explained in Section III. Once a user selects a certain video file, the length of watching this video follows the geometric distribution, with quitting probability p, which is kept as a control parameter across all video files.

We consider the one-producer scenario and create a tree topology of 16 routers (not including consumers). We cap the tree height from the video producer to all consumers at 5 levels. In this topology, edge routers connect an equal number of consumers, where each consumer has the same request rate for video files. We set the bandwidth of links between routers as 20 Mbps. The default parameters including our configurations on NDN and FESTIVE are listed in Table I.

TABLE I
Simulation Parameters

NDN	
Number of video files	20
Number of video segments per file	30
Number of NDN routers	16
Video segment playback time	10 sec
Number of video consumers	120
Video request rate per user	0.004 file/sec
Skewness factor (α)	0.8
Video quitting probability (p)	0.02
Cache capacity percentage (ω)	0.3
FESTIVE	
Drop Threshold	1.0
Combine Weight	8

B. Performance Metrics

Existing literature on adaptive control algorithms [8], [9] already adopt performance metrics aimed at measuring streaming quality. However, these metrics are user oriented, addressing QoE for each user, which offers no insight into the impact of in-network caching on adaptive streaming. This impact is contributed to by all users over time, motivating us to present the following new metrics to reflect overall streaming quality in the caching system.

1) Expected Bitrate: The ratio of bitrates selected by adaptive rate control during the entire experiment can be summarized by a bitrate distribution. Under the default simulation settings, the bitrate distributions of LRU and LFU are different, and Fig. 2 shows a snapshot of such distribution. This distribution reflects an aggregated measure of overall video quality that current network resources can support under a given caching replacement approach. We further utilize the weighted average to compute the *Expected Bitrate* ($\mathcal{E}_{br}$) according to the appearance of bitrates in users' requests, which maps the discrete bitrates to a continuous and comparable metric. A higher $\mathcal{E}_{br}$ indicates better cache efficiency in improving adaptive streaming quality.

2) Chance of Video Freezing: Avoiding video freezing is one of the primary objectives in dynamic adaptive streaming. In addition to the effort made by the adaptive rate control logic, a desirable caching scheme should satisfy the majority of requests by cache hits, which thereby decreases the video delay and the chance of having an empty buffer. This metric reflects the probability of buffer underrun ($\mathcal{P}_{vf}$) during the video playback. It is calculated by dividing the number of times

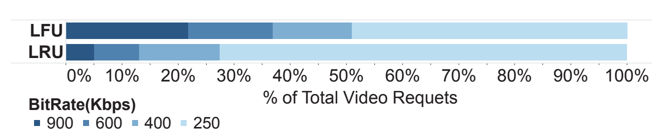


Fig. 2. Bitrate Distribution under Default Simulation Parameters

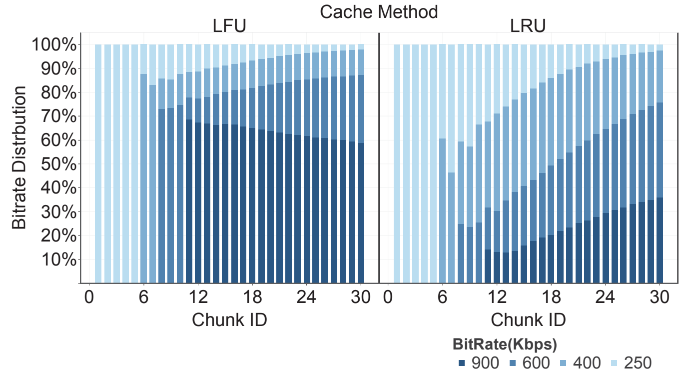
when users' buffers empty, by total video segment requests over all users. Since each adaptive rate control algorithm has its own version of a warmup phase at the beginning of streaming (*e.g.,* first 5 video segments for FESTIVE protocol), we count the total number of video requests beyond this warmup phase.

3) Bitrate Oscillation: This metric gauges the average number of bitrate switches over the viewing of an entire video file. The number of bitrate switches can be influenced by not only the adaptive rate control logic but also the caching scheme. Consecutive cache hits on video segment requests by a certain user will result in somewhat consistent throughput, which avoids frequent switches. We are interested in two specific metrics under this category, 1) *Average Times of Bitrate Switching Up* ($\mathcal{A}_{bo}$); 2) *Average Times of Bitrate Switching Difference* ($\mathcal{A}_{bo}^{\Delta}$), which are calculated as follows

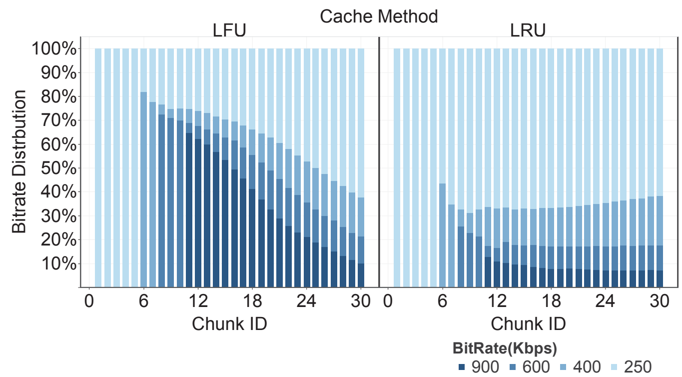
$$\mathcal{A}_{bo} = \sum_{f}^{\text{Files}} q_f \times \overline{SU(f)} \tag{2}$$

$$\mathcal{A}_{bo}^{\Delta} = \sum_{f}^{\text{Files}} q_f \times (\overline{SUp(f)} - \overline{SD(f)}) \tag{3}$$

We use SU to denote the times of bitrate switching up and SD to denote switching down. A higher $\mathcal{A}_{bo}$ means in-network caching allows users to watch a higher quality of videos, while a smaller $\mathcal{A}_{bo}^{\Delta}$ indicates that users cannot stay with high video quality and are influenced by bitrate oscillation.



(a) 80 Users (Moderate)



(b) 120 Users (Heavy)

Fig. 3. Bitrate Distribution Variation among Video Segments

C. Impact of System Load

In dynamic adaptive streaming, users compete for limited link bandwidth. Changing the number of video consumers in the network will influence the available bandwidth allocated to each user which results in different streaming experiences. In this section, We focus on the question of how system load affects the performance of caching. We vary system loads by varying the number of users, experimenting with 80 users (moderate load) and 120 users (heavy load).

We present the bitrate distribution across video chunks, as shown in Fig. 3, to observe the details of this distribution among segments and its variation as users proceed in watching video files. Due to the step-settings of FESTIVE, the first 5 segments in the warmup phase maintain the lowest bitrate (250Kbps) and the highest bitrate (900Kbps) is only allowed to be requested, starting from the 11^{th} chunk.

Fig. 3a presents that the ratio of lowest bitrate decreases for both LRU and LFU as users continue watching the video, under the moderate load. This means the simulated network managed to afford each user sufficient bandwidth to satisfy its lowest video quality requirement. However, the ratio of high bitrates between LRU and LFU are different, and the ratio of the highest bitrate decreases for LFU with increased video chunk ID. This is because LFU caching helps aggressively improving users' video quality and leaves the adaptive rate control to adjust the bitrate later in order to match the link bandwidth. However, LRU increases the ratio of high bitrates progressively. For example, at the 11^{th} chunk, the ratio of 900Kbps is 68.7% for LFU and only 14.2% for LRU. LFU exhibits a desirable caching performance and allows users to watch a higher quality of video as soon as possible.

Fig. 3b shows the bitrate distribution under the heavy load. Compared with Fig. 3a, the overall ratio of high bitrates decreases for both LRU and LFU. This result is straightforward since each user is allocated with less bandwidth. As LFU is a popularity-based caching scheme, it caters to the most frequent video contents. Thus, requests for segments with smaller chunk IDs usually get cache hits, which leads to higher throughput. When the content popularity decreases for chunks with larger IDs, those requests may have to retrieve data directly from the video producer. This explains why the ratio of high bitrates decreases for LFU with the increased segment ID since the available bandwidth cannot support traffic of high bitrates from the producer under this heavy system load. This is further verified when we discuss the bitrate oscillation later in this section.

In contrast, the video bitrate distribution for LRU is relatively stable. LRU can only achieve low cache hits ratio such that most of requests have to reach the video producer, which leads to less throughput fluctuation instead. Nevertheless, we still consider LFU is a better scheme than LRU under this heavy system load since it increases the chance for users to watch high resolutions of video contents. However, by improving the utilization of caching storage, a more desirable

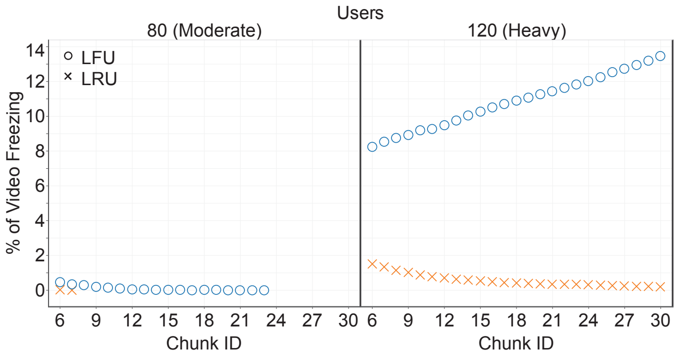
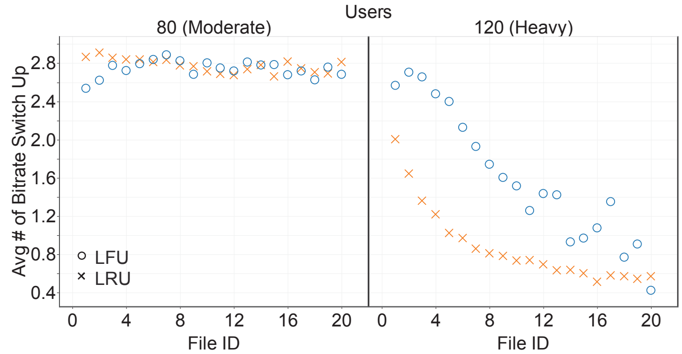
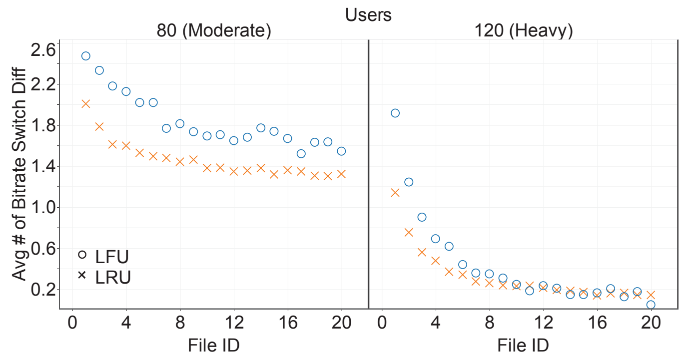


Fig. 4. Chance of Video Freezing among Segments



(a) Bitrate Switching Up



(b) Bitrate Switching Difference

Fig. 5. Bitrate Oscillation among Video Files

adaptive caching scheme should reduce the bitrate drops and maintain the high bitrates for as long as possible.

Fig. 4 shows the percentage of empty buffer under different system loads. For moderate load, the chance of video freezing is close to 0 for both LRU and LFU. It is because the network can provide sufficient bandwidth to each user, which even supports retrieving high quality videos from the producer directly. However, for heavy load, LRU and LFU result in significantly different performance. Whenever a cache miss for high bitrates occurs, the requested video segment may not be delivered to users in time. As FESTIVE requires bitrate switch smoothness, even though the current bitrate cannot be served, the rate control logic may still enforce the video player to keep requesting the current video quality for the next few chunks. This requirement increases the chance of video freezing for the subsequent segments because the video player has to consume the accumulated playback time in buffer. Referring to Fig. 3b, LFU caching allows users to make requests for videos with higher quality than LRU. Thus, the chance of video freezing for LFU increases correspondingly. There exists a tradeoff between this metric and the expected bitrate. When in-network caching facilitates improving the streaming quality, users may take the risk of video freezing at the same time. We aim for an adaptive video caching scheme which can achieve high bitrate distribution score but keep the chance of video freezing low.

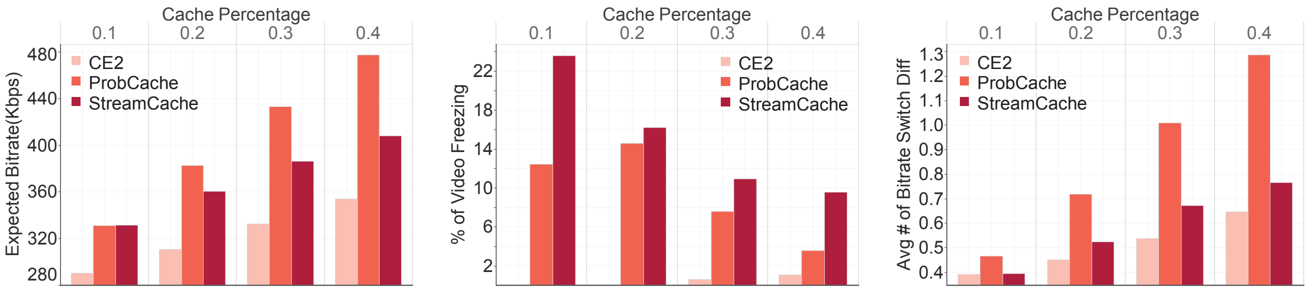
Fig. 5 shows the bitrate oscillation among video files in order to detail the variation across files of different popularity. For moderate load, the average number of bitrate switching up is almost the same between LRU and LFU since there is sufficient bandwidth to support video requests for high bitrates. However, the performance of LFU caching surpasses LRU with lower average times of bitrate switching down. Hence users' selection of bitrates is more stable under LFU, instead of switching back and forth under LRU. For heavy load, video quality of popular files (smaller index) are switched up more times than unpopular files for both LRU and LFU; this is because the in-network caching caters to frequent video requests. The bitrate switching difference of LRU and LFU is close and decreases to almost 0 for unpopular files. Since the limited cache storage can only satisfy the frequent requests, popular contents thus may sustain the same quality level while unpopular files have to switch the bitrate down in order to match the available bandwidth.

D. Impact of Cache Placement Schemes

We evaluate the performance of cache placement schemes under heavy system load with LRU as the default replacement approach. As shown in Fig. 6, and as expected, a larger caching storage can facilitate improving the overall streaming experience of users.

The specific improvement includes increased expected bitrate $\mathcal{E}_{br}$ and less bitrate oscillation $\mathcal{A}_{bo}^{\Delta}$. It is worth noting that the chance of video freezing $\mathcal{P}_{vf}$ decreases for ProbCache and StreamCache but increases for CE2 as ω changes from 0.1 to 0.2. Since the chance of cache hit of consecutive video requests for high bitrates increases with larger caching storage, the risk of buffer underrun is thus reduced, which explains the trend of ProbCache and StreamCache. The reason for different trend of CE2 is that worse cache utilization fails to effectively increase the cache hit ratio, which verifies that $\mathcal{E}_{br}$ and $\mathcal{P}_{vf}$ are correlated but their relationship is also influenced by the applied caching scheme.

Fig. 6 also shows that ProbCache outperforms CE2 across all tested cache sizes and thus provides a better streaming quality to users. Even though ProbCache is not specifically designed for video streaming, it enhances users' QoE by carefully assigning caching probability based on the length of forwarding path, which improves the cache utilization under multi-tenancy scenario. The performance improvement of ProbCache over CE2 also shows that popularity-based



(a) Expected Bitrate ($\mathcal{E}_{br}$) (b) Chance of Video Freezing ($\mathcal{P}_{vf}$) (c) Average Bitrate Switching Difference ($\mathcal{A}_{bo}^{\triangle}$)

Fig. 6. QoE across Different Cache Sizes

caching schemes receive advantages when catering to adaptive streaming traffic.

StreamCache is also designed as a popularity-based scheme. However, it does not achieve better performance than ProbCache. This result contradicts to our previous evaluation shown in [7], where we used the access time per bit as the performance metric and StreamCache outperformed ProbCache. The reason is twofold. First, StreamCache considers a generic rate adaptation scenario where the average throughput is the only metric to determine the bitrate for next request. However, in this simulation, we apply a specific FESTIVE algorithm. As FESTIVE avoids bitrate oscillation, it will force the users to stay low bitrates (*e.g.*, the first 5 segments), which worsens the cache utilization of StreamCache since StreamCache may cache high bitrates for better throughput. Second, the bitrate adaptation in StreamCache is modelled as a Discrete-time Markov chain, which is not the case used in simulations. As we apply a tree topology and link bandwidth between users and edge routers is constant, the variation on throughput among video segments mainly depends on the cache hit or miss. Thus, a same specific bitrate will always be requested for a segment. Since StreamCache selectively caches multiple bitrates, it can become inefficient in this scenario.

V. Conclusions

An emerging problem of adaptive video streaming over ICN is how to efficiently utilize in-network caching to satisfy users with higher QoE. In this work, we develop an assessment framework for adaptive video caching over ICN and adopt metrics particularly designed to measure the performance of a caching system for video contents. We conclude our work with the following two points. First, a content-popularity-based placement/replacement caching policy can effectively improve users' QoE and should be adopted by any caching system which caters to real-time adaptive streaming traffic. When considering content popularity, caching schemes can increase expected throughput and decrease bitrate oscillation under various system loads and cache sizes. However, these schemes still face the challenge of frequent bitrate change, which makes capturing users' access pattern more difficult. Second, the chance of video freezing and the expected bitrate are correlated. However, caching scheme should not just optimize the cache hit ratio only, as continuous cache hits will encourage the bitrate switching up, achieving a better expected bitrate, but leads to a higher chance of video freezing. Future design needs to strike a balance between these two metrics.

Acknowledgement

This research is funded by a grant from the Ontario Ministry of Economic Development and Innovation under the Ontario Research Fund-Research Excellence (ORF-RE) program.

References

[1] Cisco, "Cisco visual networking index: Forecast and methodology, 2015–2020," 2016.

[2] T. Stockhammer, "Dynamic adaptive streaming over http–: standards and design principles," in *Proceedings of the 2nd annual ACM conference on Multimedia systems*. ACM, 2011, pp. 133–144.

[3] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[4] S. Lederer, C. Mueller, C. Timmerer, and H. Hellwagner, "Adaptive multimedia streaming in information-centric networks," *IEEE Network*, vol. 28, no. 6, pp. 91–96, 2014.

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

[6] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1473–1499, 2015.

[7] W. Li, S. M. Oteafy, and H. S. Hassanein, "Streamcache: Popularity-based caching for adaptive streaming over information-centric networks," in *IEEE International Conference on Communications (ICC)*, 2016.

[8] 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 (TON)*, vol. 1, no. 22, pp. 326–340, 2014.

[9] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. 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, 2014.

[10] D. O. Mau, T. Taleb, and M. Chen, "Mm3c: Multi-source mobile streaming in cache-enabled content-centric networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2015.

[11] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007, pp. 15–28.

[12] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos *et al.*, "Named data networking (ndn) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[13] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*. ACM, 2012.