NETWORK-ASSISTED STRATEGY FOR DASH OVER CCN

Rihab Jmal, Gwendal Simon** and Lamia Chaari**

*University of Sfax/ CRNS Tunisia **IMT Atlantique, France

ABSTRACT

MPEG Dynamic Adaptive Streaming over HTTP (DASH) has become the most used technology of video delivery nowadays. Considering the video segment more important than its location, new internet architecture such as Content Centric Network (CCN) is proposed to enhance DASH streaming. This architecture with its in-network caching salient feature improves Quality of Experience (QoE) from consumer side. It reduces delays and increases throughput by providing the requested video segment from a near point to the end user. However, there are oscillations issues induced by caching with DASH. In this paper, we propose a new Network-Assisted Strategy (NAS) based-on traffic shaping and request prediction with the aim of improving DASH flows investigating new internet architecture CCN.

***Index Terms*—** MPEG-DASH, Content-Centric Network, video streaming, Caching.

1. INTRODUCTION

Most of the popular online video providers implement *adaptive streaming* technologies for video delivery. In particular, the MPEG Dynamic Adaptive Streaming over HTTP (DASH) standard receives a growing attention [17]. The video is transcoded into several *representations*, which differ by their bit-rate and resolution [2], each representation being then divided into smaller segments with equal duration (typically 5 seconds [9]). The main principle of adaptive streaming is that, for each segment, the client selects the most suitable representation based on multiple metrics it collects during the session. Many selection algorithms have been proposed to ensure that the bit-rate of the selected video representation is close to the actual client bandwidth [3, 18]. More recently, the MPEG-DASH standard group has introduced the Server and network assisted DASH (SAND) enhancement, which aims at enabling the collaboration between the content provider, the network operator, and the client to improve the performance of the selection algorithm at the client side [15]. To the best of our knowledge, the SAND architecture has not been explored in the scientific community.

We study in this paper how the SAND architecture can address the problems met by the delivery systems that implement cache proxies (referred to as *caches* in the following). Caches are typically implemented by Internet Service Providers (ISPs) to cope with the massive growth of video streaming usage. Caches are also essential in the design of Content Centric Network (CCN) [5], a novel Internet architecture, which is characterized by intelligent routers, in-network caching, and multipath delivery. Although CCN aims at improving content delivery, the implementation of caches introduces some troubles in DASH [10]. It is especially the case in network configurations where the bottleneck of the client-server connection is between the cache and the server. As epitomized in Figure 1, caches generate a misunderstanding of the available bandwidth at the client side, which ultimately result in oscillations between representation due to the misinformed selection algorithm. Here the client believes its bandwidth is 3 Mbps when it is served from the cache, but only 1.5 Mbps when it is served from the server.

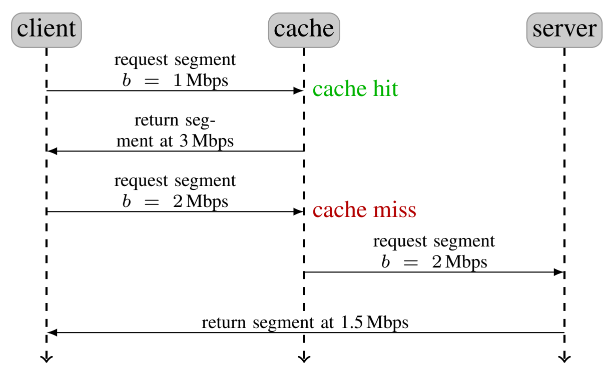

Fig. 1. Oscillations and rate adaptation misunderstanding

The contribution of this paper is the design and the evaluation of a delivery architecture, which combines SAND and CCN. Our proposal overcomes the oscillations that result from standard caching techniques. The CCN node is augmented with the feature of a DASH-aware network element (DANE), the main entity of SAND architecture. The DANE-CCN node executes three main actions: (i) bandwidth measurements, (ii) prediction, and (iii) traffic shaping. Despite many proposals to reconcile video delivery and CCN in the literature (see Section 2), our work is pioneer for the exploration of SAND with the general precepts of Information-Centric Network (ICN) and CCN.

The rest of this article is organized as follows. We detail the literature in Section 2. We introduce SAND in MPEG DASH and CCN as well as the architecture of a SAND-CCN delivery system in Section 3. We present in Section 4 our proposal while its performance evaluation is in Section 5. Finally, conclusions are drawn in Section 6.

2. RELATED WORK

Recent work evaluate caching performance with DASH and investigate the integration in CCN. Liu et al. [11] study the caching performance and the overhead caused by DASH over CCN. This early proposal illustrates the overhead increase when using CCN and reveal possible improvements. Rainer et al. [14] examine the performance of DASH in ICN based on different forwarding strategies and different client-side rate adaptation. They show that ICN multi-path and caching capabilities can improve MPEG DASH performance.

While DASH takes advantage from CCN features, new challenges are also introduced. We found in particular that two directions can be exploited to resolve the caching misunderstanding issue. First, the client-side can be modified to be informed about the requested content and the rate switching. A series of recent studies [7, 12, 16] focuses on such approach; the authors proposed new Rate Adaptation mechanisms for DASH client. Secod, network elements can be improved [10, 13]. A summary of these related works is provided in Table 1 and in the following.

Tian and Liu [16] use the buffered video time as a reference signal to guide video rate adaptation. Their algorithm balance the needs for video rate smoothness and bandwidth utilization. They used machine learning based TCP throughput estimation algorithms to effectively guide DASH server switching. In QDASH [12], a measurement proxy is used to measure the available network bandwidth, round-trip time (RTT), and the loss rate. Using these measurements, Mok et al. [12] estimate the highest acceptable video rate. When the preferred bitrate was estimated to be lower than the current bitrate, they proposed to switch to an intermediate level instead of immediately switching to the preferred rate. Finally, Segment-Aware Rate Adaptation (SARA) algorithm [7], which is a recent proposal, considers the segment size variation in addition to the estimated path bandwidth and the current buffer occupancy to accurately predict the time required to download the next segment.

However, only improving the client-side rate-adaptation and buffer-adaptation limits this gain specifically to the player. Its configuration by the user or a network administrator is also difficult. The scalability of the system can be limited through a measurement proxy as proposed by Mok et al. [12] but it requires additional hardware.

Alternatively, network devices have the potential to assist client system in selecting the representations effectively. Due to the broader view on the network and its capabilities provided by the network devices, an exchange of knowledge between devices and client system can be leveraged to enhance DASH (i.e. better estimation of the available bandwidth). Lee et al. [10] discuss the cause of oscillations and proposed a cache-based shaping solution VISIC (Video Shaping Intelligent Cache) based on estimating the available bandwidth and traffic shaping from the cache to provide stability and buffer fullness. Posch et al. [13] propose in-network scalable content encodings, where ICN routers are adapting entities that manage requests.

Nevertheless, these approaches do not guarantee the best QoE and, unfortunately, video stalling persists.

Our contribution aims to overcome standard caching issues in DASH by integrating intelligent CCN operating within SAND architecture based on communication between client system and DANEs.

3. BACKGROUND AND ARCHITECTURE

3.1. Background on SAND in MPEG-DASH

In order to enhance the delivery of DASH content, SAND creates communications between different network elements. With the aim of improving the efficiency of streaming sessions, exchanged messages provide information about real-time operational characteristics of networks.

SAND architecture is made of four broad components as illustrated in Figure 2: DASH clients, regular network elements, DANE and Metrics server. Regular network elements are DASH unaware and treat DASH delivery objects as any other object. Metrics servers are DASH aware and are responsible of carrying metrics from DASH clients. Finally, DANEs, which are DASH-aware components, are the elements with a certain knowledge about DASH.

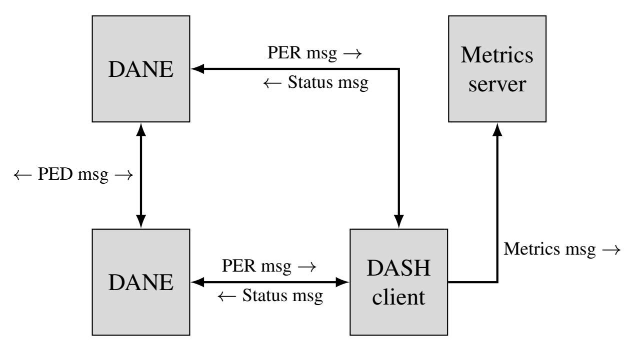

Fig. 2. SAND architecture

DANEs can prioritize, parse or modify Media Presentation Description (MPD) and DASH segments. Examples of DANE may be a DASH server which is the media origin, content delivery network (CDN), and caches.

With the purpose of establish communications between different SAND elements, exchanged messages are categorised on four types: Parameters Enhancing Delivery (PED),

Table 1. Summary of Related Work

Paper and Date	CCN/NDN	Client-Side	In-Network-Side	Solve Misunderstanding issue	Shaping Traffic
DASC [11] 2013	✓	✗	✗	✗	✗
DAS-NDN [14] 2016	✓	✓	✗	✗	✗
ViSIC [10] 2014	✗	✗	✓	✓	✓
INA-ICN [13] 2014	✓	✗	✓	✗	✗
Smooth-Agile [16] 2016	✗	✓	✗	✗	✗
QDASH [12] 2012	✗	✓	✗	✗	✗
SARA [7] 2016	✗	✓	✗	✗	✗
Our proposal	✓	✗	✓	✓	✓

Parameters Enhancing Reception (PER), Status and Metrics messages. Different DANEs exchange information through PED messages. PER messages are sent from DANEs to DASH clients. In reverse, from DASH clients to DANEs, Status messages are sent. DASH clients send Metrics messages to Metrics servers. These messages can be about TCP connections, HTTP Request/Response, Representation Switch events, Buffer level at a given point in time or playlist collected informations. The DASH client inform DANEs through Status messages about Anticipated Requests, Shared Resource Allocation (i.e. access link Bandwidth), Accepted alternatives which allow knowing the media delivery path and such message is sent simultaneously with DASH segment request, absolute deadline and max Round-Trip Time (RTT). In reverse, PER messages include DaneResourceStatus which allow informing DANE in advance about knowledge of segment availability and caching status of the segment in the DANE as well as available and possibly anticipated to be available data structures to the DASH client and also signal which data structures are unavailable. Other PER messages concern MPD validity end time, AvailabilityTimeOffset, QoSInformation, DeliveredAlternative and SharedResourceAssignment informing about available Bandwidth to use. On the other hand, PED messages carry Bandwidth information which can be extracted from the MPD and shared with the service provider or operator to help facilitate the derivation of network Quality of Service (QoS) parameters at the DANE or another network element.

3.2. Background on CCN

The generic concept of CCN consists in replacing host-to-host based communication by a content-based communication. A content is requested by sending an Interest Packet to a CCN node which is a specific router characterized by three data structures: Content Store (CS), Forwarding Information Base (FIB) and Pending Interest Table (PIT). The CS is a local cache. The FIB transmits interest packets to potential data sources. The PIT registers the incoming interest interfaces.

Once a CCN node receives an Interest, it checks its CS to see if there is a copy of the requested content. If so, a response is sent directly through a Data Packet. Otherwise, the Interest Packet is forwarded to other CCN nodes until a Data Packet is replied from an intermediate node or from the origin server. The Data packet follows the symmetric path from source based on the information of the PIT to find the route back to the requester(s). Then, each node traversed stores a copy of the content for future requests [6].

3.3. Architecture of a SAND-CCN delivery system

In our architecture, we propose that a CCN node may be considered as a DANE as depicted in Figure 3. It performs caching and content distribution following CCN principle.

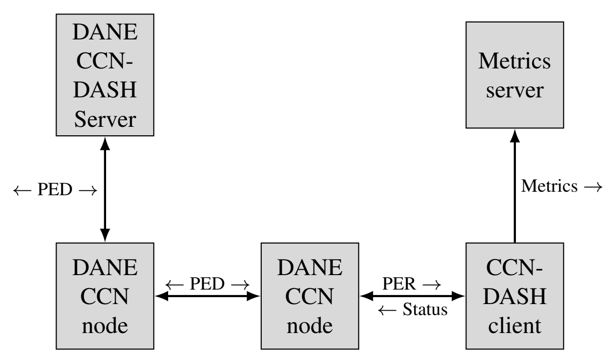

Fig. 3. SAND-CCN architecture

The described architecture takes advantage of CCN caching: paths and delays are reduced because the requested segment is cached closer to the client. However the caching operation can lead to frequent alternation between cache and server when serving back client requests for successive video segments. These oscillations cause a misunderstanding of the rate adaptation. In the next section, we explain our new Network-Assisted Strategy (NAS) strategy applied on the SAND-CCN architecture.

4. NAS PROPOSAL FOR DASH OVER CCN

We describe as follow our proposal in the case of a simple configuration, which consists of one CCN node, CCN-DASH client and CCN-DASH server. The CCN node operating as DANE performs the following tasks:

1. Bandwidth measurements calculated from the CCN node to both client and server.

2. Prediction process allowing the improvement of Quality of Experience (QoE). This process consists of requests anticipation and buffering based on preliminary measurements values on the bases of the MPD information, prefetching segments with estimated representations.

3. Traffic Shaping to handle rate adaptation choices in order to be suitable for the already buffered segments representations. This process can be more benefic to Internet providers and operators

4.1. Bandwidth Measurement

Since the bitrate of the DASH segments are variable, the load on a link dynamically changes, Instant load values are averaged by using the smoothing formula:

$$L_{ij} = \alpha \times L_{ij}^{t} + (1 - \alpha) \times L_{ij}$$

where L_{ij}^{t} is the current load, α is the smoothing factor, and L_{ij} is the averaged load of the link i on the jth path. The available bandwidths of all paths are calculated by measuring the bottleneck link bandwidth:

$$abw_j = \min\left(C_{ij} - L_{ij}\right), \forall 1 \leq i \leq M_j$$

where M_j is the number of links and C_{ij} is the capacity of the link i.

We use available bandwidth measurement instead of simple throughput measurement to help clients in the selection of the most suitable video quality levels [4]. In our simulation, we set α to 0.5 to reduce the impact of temporary fluctuations without neglecting the current load.

4.2. Shaping Process

The client gets the information about the bit-rate of the different representations by downloading the MPD file. An adaptation algorithm in the client-side selects the segment whose bit-rate is the most suitable. This rate adaptation algorithm evaluates the throughput and based on its observations it increases or decreases the bit rate of the next video segments. In order to avoid the misunderstanding issue resulted from oscillations, we propose a shaping method to modify the response throughput with the aim of leveraging client observations and decisions.

The bandwidth between the CCN node and the client is noted bw_c while bw_s is the bandwidth between the CCN node and the server. The resulting throughput to respond client from CCN node T_{res} should be higher or equal to the bit rate of the requested segment:

$$T_{res} \geq S_{req} \qquad bw_s \leq T_{res} \leq bw_c$$

If S_{req}+1 exist in cache, the throughput T_{res} should guide the client (rate adaptation algorithm) in its choice for the next segment. Hence, the shaping algorithm has as input the bitrate of requested segment, the available bandwidth and the cached next segment bit rate if it exists. The output of this algorithm is the throughput of the response T_{res}.

The main idea of our algorithm is to shape the throughput to obtain a transfer that is closer to the bit-rate of the next cached segment. The algorithm is based on the information of the available bandwidth between the client and the CCN node (abw_c), the representations of the next segment which are cached, and the bit rate r_{curr} of the requested segment. Here, we denote by r_i the available representation rate of the next segment $S_{req} + 1, 1 \leq i \leq n$ for n representations. Moreover, the resulting shaped throughput must be close to the predicted representation bit rate of the next requested segment. Meanwhile, β is a configuration parameter less than 1, which we set in our simulation to 0.9 to slightly reduce the rate for shaping. Our goal is to serve segments at the highest possible throughput without causing rate up-shift on the client.

Algorithm 1 Traffic Shaping

```
T_res = β × r_curr                          ▷ initiatlization
if cache=true then          ▷ at least repr. S_req + 1 in cache
    for i = 1 ... n do
        if r_i in cache and abw_c > r_i then
            T_res = β × r_i
        end if
    end for
else
    prediction_process()
    T_res = abw_s
end if
return T_res
```

4.3. Prediction Process

Through exchanged SAND messages, we propose to add some information to the CCN node. A request is sent from CCN node to the server for the segment $S_{req} + 1$ with bit rate r_{pred}. Two constraints must however be addressed:

- Buffer occupancy (cache capacity).
- Available bandwidth (abw_c/abw_s): the segment representation to be anticipated and cached in the CCN node

can be the representation with the highest bit rate supported by the available bandwidth from the server to the CCN node

We obtain:

$$r_{pred} = \underset{r_i, 1 \leq i \leq n}{\arg\min} |r_i - abw_s|$$

5. PERFORMANCE EVALUATION OF NAS

We have implemented our proposal in an extended ndnSIM simulator supporting DASH [8]. We evaluated its performance across a number of different scenarios. We tested the content transfer with the open-source movie Big Buck Bunny encoded by SVC [1] as multimedia streaming content. The dataset contains 300 video segments in several representations with a segment length of two seconds. The CCN-DASH client downloads the MPD, then Interests for successive segments are sent. Next, a Data packet is received with the corresponding representation of the requested segment. We used a simple topology: a CCN-DASH server, two CCN-DASH clients and a CCN router with a maximum cache (CS) size 1000 contents and Least recently used (LRU) cache management policy. Two consumers are used to allow caching of segments in the CCN router. We used a Start/Stop application on consumers to provide for each one the total available bandwidth. Each consumer takes 600 seconds of simulation to download all video segments. The first client requests for segments with classic DASH from the original server. All segments are in the cache. Then, the second client requests for segments from the CCN node. Our evaluation is based on results of the second consumer. The Shaping and Prediction are enabled on the CCN router. The server node provides the MPD as well as the video segments with four representations. R0 with bitrate= 0.6 Mbps, R1 with bitrate=2 Mbps, R2 with bitrate=4.87 Mbps and R3 with bitrate=9.43 Mbps.

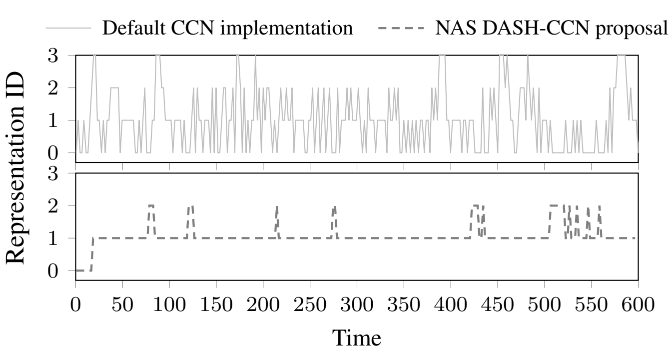

Fig. 4. Bitrate oscillations with 5 Mbps

Firstly, we verified the bit-rate oscillations for the default CCN implementation and how NAS DASH-CCN proposal alleviates the frequency of these oscillations. The simulation with 5 Mbps of bandwidth and 10 Mbps are illustrated by Figure 4 and Figure 5 respectively. We presented the segment

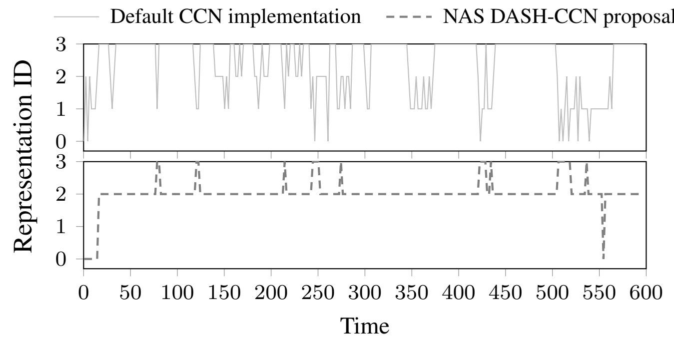

Fig. 5. Bitrate oscillations with 10 Mbps

representation ID downloaded which is impacted effectively by the use of NAS. We can observe the stability of the video transfer with NAS compared to the default CCN implementation, which effect robustly on the QoE.

Afterwards, we compared the performance of NAS to another implementation based on the principal of proposals cited in Section 2 [7, 12, 16] . We considered in our comparison test a DASH-CCN implementation with client-side adaptation that take account of the buffer state. Figure 6 shows the high performance of our proposal in term of segment video bit rate as well as stability in case of 10 Mbps bandwidth. In order to highlight the effect of the prediction process of NAS DASH-CCN, we applied a variation of the bandwidth between the server and the CCN node from 10 Mbps to 7 Mbps. Results of bit rate are described in Figure 7. Our proposal offers a stable bit rate which considered the bandwidth variation otherwise than the client-side adaptation which effected frequent oscillations with an important decrease in bit rate.

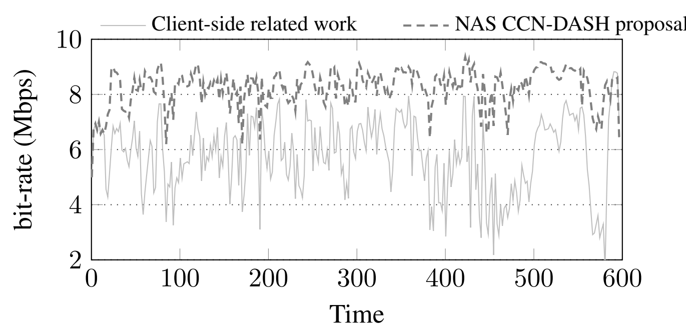

Fig. 6. Bit-rate at the client for the 10 Mbps bandwidth

In summary, CCN allows smart caching strategy, where the the video segments are closer to the consumer and our proposed NAS for DASH over CCN effectively controls video streaming and cached segments, keeps stability of downloading segments video, and achieves near-optimal data throughput with zero packet loss (stalls) across all the simulation cases we have tested. NAS provides a bridge between the traditional DASH-server streaming and DASH-client HTTP streaming.

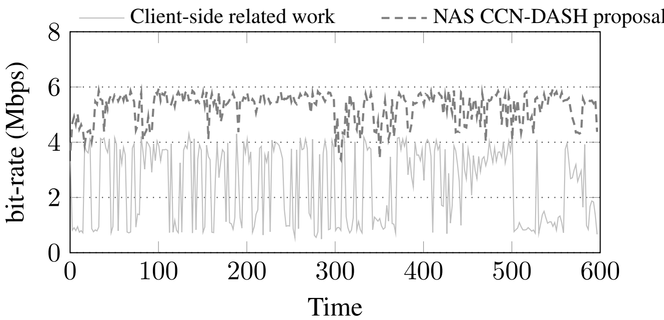

Fig. 7. Bit-rate at the client with bandwidth variation to 7 Mbps

6. CONCLUSION

In this paper, we addressed the problem of bit-rate oscillations in DASH systems caused by simple cache servers. We identified a misunderstanding issue in bandwidth readings at the DASH client side which may cause oscillations in bit rate and stalls. Our proposed solution accounts for the improvement of DASH performance through a new Network-Assisted Strategy inspired from SAND architecture based on CCN nodes as DANEs. We integrated our strategy on CCN nodes to provide a stable bit rate of video segment and always try to cache the following segment that the client may request with the appropriate representation which guarantee the QoE.

Our future work take account of an extended version of our work, which integrate Software Defined Network (SDN) controllers to perform the supervision of the whole network. We plan to test big topologies with CCN nodes, DASH clients and servers and SDN controllers which manage all devices.

References

[1] Scalable video coding bitstream reordering for svc-dash, 2016. http://tinyurl.com/ITEC-SVCMux.

[2] R. Aparicio-Pardo, K. Pires, A. Blanc, and G. Simon. Transcoding live adaptive video streams at a massive scale in the cloud. In *Proc. of ACM MMSys*, 2015.

[3] W. E. Castellanos, J. C. Guerri, and P. Arce. A qos-aware routing protocol with adaptive feedback scheme for video streaming for mobile networks. *Computer Communications*, 77:10–25, 2016.

[4] C. Cetinkaya, E. Karayer, M. Sayit, and C. Hellge. Sdn for segment based flow routing of dash. In *Proc. of IEEE ICCE-Berlin*, 2014.

[5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. Briggs, and R. Braynard. Networking named content. *Commun. ACM*, 55(1):117–124, 2012.

[6] R. Jmal and L. Chaari Fourati. An openflow architecture for managing content-centric-network (OFAM-CCN) based on popularity caching strategy. *Computer Standards & Interfaces*, 2016.

[7] P. Juluri, V. Tamarapalli, and D. Medhi. Qoe management in DASH systems using the segment aware rate adaptation algorithm. In *Proc. of IEEE/IFIP NOMS*, 2016.

[8] C. Kreuzberger, D. Posch, and H. Hellwagner. Amust framework - adaptive multimedia streaming simulation framework for ns-3 and ndnsim, 2016. https://github.com/ChristianKreuzberger/amust-simulator.

[9] C. Kreuzberger, B. Rainer, H. Hellwagner, L. Toni, and P. Frossard. A comparative study of DASH representation sets using real user characteristics. In *Proc. ACM NOSSDAV*, 2016.

[10] D. H. Lee, C. Dovrolis, and A. C. Begen. Caching in HTTP adaptive streaming: Friend or foe? In *Proc of ACM NOSSDAV*, 2014.

[11] Y. Liu, J. Geurts, J. Point, S. Lederer, B. Rainer, C. Müller, C. Timmerer, and H. Hellwagner. Dynamic adaptive streaming over CCN: A caching and overhead analysis. In *Proc. of IEEE ICC*, 2013.

[12] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang. QDASH: a qoe-aware DASH system. In *Proc. of ACM MMSys*, 2012.

[13] D. Posch, C. Kreuzberger, B. Rainer, and H. Hellwagner. Using in-network adaptation to tackle inefficiencies caused by DASH in information-centric networks. In *Proc. of the ACM VideoNext Workshop*, 2014.

[14] B. Rainer, D. Posch, and H. Hellwagner. Investigating the performance of pull-based dynamic adaptive streaming in NDN. *IEEE Journal on Selected Areas in Communications*, 34(8):2130–2140, 2016.

[15] E. Thomas, O. Van Deventer, T. Stockhammer, A. Begen, and J. Famaey. Enhancing MPEG DASH performance via server and network assistance. In *Int. Broadcaster Conf. (IBC)*, 2015.

[16] G. Tian and Y. Liu. Towards agile and smooth video adaptation in HTTP adaptive streaming. *IEEE/ACM Trans. Netw.*, 24(4):2386–2399, 2016.

[17] N. Weil. The State of MPEG-DASH 2016. Technical report, Streaming Media, March 2016. http://tinyurl.com/hs2zbuh.

[18] S. Xiang, M. Xing, L. Cai, and J. Pan. Dynamic rate adaptation for adaptive video streaming in wireless networks. *Sig. Proc.: Image Comm.*, 39:305–315, 2015.