Investigating the Performance of Pull-based Dynamic Adaptive Streaming in NDN

Benjamin Rainer, Daniel Posch and Hermann Hellwagner
Institute of Information Technology
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria
Email: first.lastname@itec.aau.at

***Abstract*—Adaptive content delivery is the state-of-the-art in real-time multimedia streaming. Leading streaming approaches, e.g., MPEG-DASH and Apple HLS, have been developed for classical IP-based networks, providing effective streaming by means of pure client-based control and adaptation. However, the research activities of the Future Internet community adopt a new course that is different from today's host-based communication model. So-called Information-Centric Networks are of considerable interest and are advertised as enablers for intelligent networks, where effective content delivery is to be provided as an inherent network feature. This paper investigates the performance gap between pure client-driven adaptation and the theoretical optimum in the promising Future Internet architecture Named Data Networking (NDN). The theoretical optimum is derived by modeling multimedia streaming in NDN as a fractional Multi-Commodity Flow Problem and by extending it taking caching into account. We investigate the multimedia streaming performance under different forwarding strategies, exposing the interplay of forwarding strategies and adaptation mechanisms. Furthermore, we examine the influence of network inherent caching on the streaming performance by varying the caching polices and the cache sizes.**

***Index Terms*—Information-Centric Networking; Named Data Networking; Multimedia; Dynamic Adaptive Streaming.**

I. Introduction

Dynamic Adaptive Streaming (DAS) has become the state-of-the-art for on-demand and real-time multimedia streaming. The majority of streaming platforms provided by large business players like Netflix, Amazon, and Sky rely on DAS. Besides Apple HTTP Live Streaming, Microsoft Smooth Streaming, and Adobe's HTTP Dynamic Streaming solutions, a new standard has emerged and become wide-spread, MPEG-DASH (Dynamic Adaptive Streaming over HTTP), ratified by ISO/IEC. MPEG-DASH specifies the description of the available multimedia content and how it shall be segmented [1]. Segments are defined as equally-long temporal units of the multimedia content. An adaptation logic/mechanism at the client side is responsible for deciding which representation (scalability in the spatial, temporal, and quality domain) a client selects for downloading a specific segment. MPEG-DASH does not specify the adaptation mechanism, enabling competition in industry and research. The mentioned technologies/solutions have been developed for over-the-top multimedia streaming in classical IP-based networks using pure client-driven adaptation among the available representations.

However, given the research activities of the Future Internet community, it is evident that there is tremendous interest in a novel information-centric communication model. Instead of IP's host-based communication, the concept of Information-Centric Networking (ICN) [2], [3] proposes a content-centric communication paradigm, where content only is addressed, not nodes (hosts). Introducing a content-based security model, the concept of ICN provides advantages over classical IP-based networks, including network-inherent caching of content and multi-path forwarding capabilities. Various approaches to implement an ICN architecture have been proposed [4], [5], [6], [7], and [8]. We use the concepts of Named Data Networking (NDN) [8]. As already mentioned, NDN establishes that data is addressed by its name. In order to request content, a client sends an *Interest* message that asks the network for the desired content. This Interest message is forwarded by the NDN nodes until a matching content replica is located, which is then returned in a *Data* packet. *Data* packets are always returned on the reverse path of the requesting Interests. Since the NDN proposal also includes that nodes maintain a cache (content store) to store a limited amount of *Data* replicas, content may not have to be retrieved from the content origin.

The objective of this paper is to *investigate the performance of pull-based DAS in NDN* using different (Interest) *forwarding strategies* at the network level and different client-side adaptation mechanisms at the application level, especially under non-optimal conditions (e.g., network congestion). We determine the performance gap between the theoretically possible and realized streaming performance by NDN considering concurrently streaming consumers. We further compare these performance evaluations to classical MPEG-DASH streaming in IP-based networks.

In order to derive upper bounds for the multimedia streaming performance in NDN without and with caching, we model the concurrent streaming activities by a given number of clients in a network as a Multi-Commodity Flow Problem (MCFP) [9]. The solution to the MCFP provides us with upper bounds taking multi-path transport into account. Both the theoretical investigations and the practical evaluations clearly state that NDN, as a Future Internet architecture, is able to compete with nowaday's IP-based networks in the case of multimedia streaming. Please note that we do *not focus on finding a near-optimal adaptation heuristic or forwarding strategy* for DAS in NDN. We rather carefully select representatives for each of the algorithms and compare every possible combination with respect to their general performance.

The remainder of the paper is organized as follows. Section II provides an overview of multimedia streaming in NDN and introduces the forwarding and adaptation algorithms used later on. The fractional Multi-Commodity Flow Problem is introduced in Section III. The evaluation using an NDN-based simulator is presented in Section IV. Section V discusses the results and concludes the article.

II. Related Work and Preliminaries

Lederer et al. [10], [11] investigated the performance of pull-based DAS using multiple links and the benefits of SVC in CCN/NDN (CCN is a concrete implementation of ICN). To this end, experiments on a small scale were conducted using real network traces from mobile networks. The major result is that, leveraging the inherent multi-path principles of NDN, it is possible to achieve a good performance when streaming multimedia content. Liu et al. [12] investigated the caching performance and the overhead caused by pull-based DAS over CCN/NDN. The paper shows that the overhead caused by CCN is large and that there is room for improvement.

An implementation of Voice over IP in CCN/NDN is presented in [13]. This implementation demonstrates the real-time capabilities of CCN/NDN. Detti et al. [14] show that ICN can be used for offloading the cellular radio interface in the case of multimedia streaming using Apple's HTTP Live Streaming. Recently, the Internet Research Task Force (IRTF) took up the topic of pull-based multimedia streaming in ICN/NDN and initiated a first Internet draft [15].

However, related work has not yet investigated how well NDN is suited for pull-based multimedia streaming by taking into account different forwarding strategies and, in the case of MPEG-DASH-compliant scalable multimedia content, different types of adaptation logics.

A. Adaptation Mechanisms

Previous work has investigated how multimedia streaming in NDN can be accomplished. NDN's architecture and principle affirms a pull-based multimedia streaming approach where the client *requests* multimedia content. MPEG-DASH fits this purpose very well [1], [10], [12] because it adopts a pull-based approach. MPEG-DASH builds on top of HTTP and provides a so called Media Presentation Description (MPD) that contains information about the segmented multimedia content. The MPD contains information about different representations of the multimedia content (e.g., differing in the spatial and/or temporal and/or in the quality domain). Once a client has received the MPD it knows how many representations are available and which bit-rates these representations provide/require.

For our experiments we use MPEG-DASH-compliant multimedia content that is encoded using Scalable Video Coding (SVC) [16]. SVC offers the possibility to encode video content into a base layer and several enhancement layers. The enhancement layers build upon the base layer and provide scalability in the spatial and/or temporal and/or quality domain(s). Instead to non-scalable encodings (e.g. H.264/MPEG-4 AVC), this principle fits very well with NDN's inherent caching, as clients requesting different content representations, at least have the base layer in common. This increases the overall cache hit ratio in the network and, thus, increases the delivered quality of the multimedia content [17].

In order to investigate the interplay of the forwarding strategies discussed in Section II-B and the adaptation mechanisms at the clients, we select for each type of client-side adaptation mechanism (no adaptation, rate-based adaptation, and buffer-based adaptation) one representative as follows:

No Adaptation: Here, a client tries always to request each segment from each layer. Thus, it simply tries to retrieve the best multimedia representation.

Rate-based Adaptation: Here, a client measures the currently available bandwidth while downloading a segment. Then the client estimates the future available bandwidth using an exponential moving average given by $b_{k+1} = (1 - \alpha) \cdot b_k + \alpha \cdot b$, where b_{k+1} denotes the new estimate, b_k denotes the previous estimate, and b denotes the currently measured bandwidth [18]. For our experiments, we select $\alpha = 0.3$, which reduces the impact of recent measures on the estimate. The lower alpha, the more influence the historic measurements have on the estimated bitrate. Based on the estimated bitrate, the client selects a suitable representation from which it tries to download segments.

Buffer-based Adaptation: Here, the decision which representation is selected to download a segment is based only on the client's playback buffer. We adopt the adaptation logic described in [19] which uses a deadline-based approach for selecting the appropriate representation and is optimized for SVC content. It first downloads k segments of the lowest representation (layer). If the playback did not yet approach the playback timestamps of these k segments, the adaptation logic tries to download the first (later on, the next) enhancement layer for these segments. The quality (layer) that is downloaded for the available segments in the buffer follows a pattern of sloping stairs favoring segments that are closer to the playback time stamp. The adaptation logic tries always to maintain k segments of the lowest representation in the buffer before fetching any enhancement layers [19].

B. Interest Forwarding Strategies in NDN

In classical IP-based networks forwarding decisions are determined by routing. This is necessary to avoid loops, inhibiting opportunities to realize an adaptive and flexible forwarding plane. Therefore, inherent multi-path transmission is usually not available, reducing content distribution efficiency. In NDN instead, routing shall hold a supporting role to forwarding, providing sufficient potential to enhance content dissemination at the forwarding plane [20]. It is stated that routing shall take a bootstrapping role. Forwarding is responsible to realize effective content delivery on the propagated routes, taking failures into account, and recovering from them independently from routing.

For the experiments presented in this paper, we use *ndnSIM 2.0* [21] which builds on top of *ns-3*. Since forwarding strategies have a significant influence on the performance of content delivery in NDN, we consider a variety of strategies for our investigations. To this end, we selected forwarding strategies with different objectives, including maximal coverage (Broadcast) and throughput (SAF), minimal hop count (BestRoute) and delivery time (NCC), and effective cache utilization (iNRR). In the following we outline the principles of each strategy:

BestRoute: This strategy relies on routing information and forwards Interests on the path with the lowest costs considering a specific metric. We use the hop count as the metric.

Broadcast: This scheme forwards received Interests to all available interfaces (according to the interfaces that match the content name prefixes in the Forwarding Information Base (FIB), determined initially by the routing protocol), except the incoming interface. Note that multiple copies of an Interest may be created if multiple interfaces are registered in the FIB.

NCC: Each node monitors the delays of its interfaces. The delay is defined as the time period that elapses until a forwarded Interest is satisfied by a Data packet. Interests are forwarded to the interface, that provides content with the lowest delay. This forwarding strategy is similar to the forwarding strategy used in CCNx 0.7.2 (www.ccnx.org) and its name was derived by flipping the initials of the term *Content-Centric Networking* (CCN) [6].

SAF: This strategy, called *Stochastic Adaptive Forwarding* (SAF) [22], mimics the behavior of a water pipe system where each network node represents a crossing and distribution node with a pressure control valve. SAF is not content/prefix agnostic and, therefore, maintains a certain state for each content/prefix observed at the network nodes. The Interests are forwarded based on a probability distribution on the interfaces for the corresponding content. The probability distribution is learned by maximizing a given measure (e.g., throughput). Here, we use a purely throughput-based measure that counts how many Interests are satisfied during a given time period.

iNRR: Ideal Nearest Replica Routing [23] couples caching and forwarding. The approach makes use of an *oracle* that provides information on the availability of content in all caches in the network. The algorithm determines the nearest content replica (in terms of hop count) and forwards the Interest to the corresponding interface to obtain the replica.

III. Multimedia Streaming as a Multi-Commodity Flow Problem in NDN

1) Modeling the Upper Bound without Caching

We model the *fractional* MCFP for a given network, clients and their corresponding servers using the paths from each client to its servers. One may also see the network as constrained to the maximization of the possible multimedia bitrate that each consumer may retrieve. As a preprocessing step, we compute every possible path from the clients to their servers. Let the three-tuple $G := (V, E, c)$ be a weighted graph that represents the underlying network topology, where V denotes the set of vertices, $E \subseteq V \times V$ denotes the set of edges, and $c : E \rightarrow \mathbb{R}$ assigns a bandwidth capacity to each edge. C denotes the set of clients. Then the paths from a client to a server can be enumerated by a slightly modified version of the classical breadth-first or depth-first search. We denote P as the set of paths for all (s, t) pairs, where s denotes the client and t denotes the corresponding server for a given client s. We denote S as the set of all client-server pairs (s, t). Note that for a single client s multiple (s, t) pairs exists if a client's multimedia stream can be served by multiple servers. Further each (s, t) pair may have multiple delivery paths, hence, multiple sub-flows. We further denote P_i as the set of paths for client i to all of its servers. For each path $p \in P$ we have a variable $x_p \in \mathbb{R}_+$ representing the bandwidth consumed on path p. This allows us to set up the LP 1 using vector $\boldsymbol{y} \in \mathbb{R}^{|C|}$ as auxiliary variable (henceforth vectors are denoted using bold math symbols) as follows:

$$\text{minimize} \quad -||\boldsymbol{y}||_1 \tag{1a}$$

subject to

$$y_i \cdot \text{minBitrate}_i - \sum_{p \in P_i} x_p \leq 0 \quad, \forall i = 1, \ldots, |C| \tag{1b}$$

$$\sum_{(u,v) \in p} x_p \leq c((u, v)) \quad, \forall (u, v) \in E, p \in \bigcup_{i=1}^{|C|} P_i \tag{1c}$$

$$\sum_{p \in P_i} x_p \leq \text{maxBitrate}_i \quad, \forall i = 1, \ldots, |C| \tag{1d}$$

Equation 1a provides the objective function for the optimization problem. $|| \cdot ||_1$ denotes 1-norm, which is defined as $||\boldsymbol{x}||_1 := \sum_{i=1}^{n} |x_i|$. Here, x_i denotes the i-th element of vector $\boldsymbol{x}$ and n denotes the number of elements in $\boldsymbol{x}$. In LP 1, y_i represents the auxiliary variable for the i-th client, and minBitrate_i denotes the minimum bitrate the i-th multimedia stream. Equation 1b denotes the constraint that each multimedia stream shall at least get the lowest possible media bitrate available. The LP becomes infeasible if this lower bound cannot be achieved by at least one of the clients (this is the case if the $y_i \geq 1$, where $i = 1, \cdots, |C|$). This is a very strict criterion ensuring a smooth media playback for all clients. This constraint may be relaxed by choosing minBitrate_i lower than the lowest available representation bitrate or by allowing $y_i < 1$. However, this might lead to clients receiving too few resources, even for streaming the base layer resulting in so-called media playback stalls (playback disruptions due to buffer drains of the video/audio buffer in the playback software). Equation 1c takes the edge capacities into account such that all paths that have an edge (u, v) in common do not consume more than the available capacity. Equation 1d denotes the constraint for restricting the maximum used media bitrate. A client cannot retrieve a higher representation bitrate than the highest available one (maxBitrate_i); this is again a very strict constraint. Allowing higher values than the highest available representation bitrate (i.e., arbitrarily high) would yield the highest possible streaming bitrate for each client.

LP 1 provides us with an upper bound for the case where we do not assume that any content is cached by the intermediate nodes on the paths. It further assumes that all clients start streaming at the very same time. An optimal solution to the

Algorithm 1 Determine Upper Bound of the Average Multimedia Bitrates with *Idealized* Caching

1: $L \leftarrow disjointStreams(M, C)$
2: **while** $comb \leftarrow getNextCombination(L)$ **do**
3: $\quad \{rG, result_n\} \leftarrow solveMCFP(comb, G)$
4: $\quad S' \leftarrow createClientServerPairs(S, rG)$
5: $\quad \{sG, result_{|C|-n}\} \leftarrow solveMCFP(S', rG)$
6: $\quad R[comb] \leftarrow \{result_{|C|-n}\}$
7: **end while**
8: **return** $\max\{R\}$

introduced LP provides therefore also an upper bound for the streaming scenario in TCP/IP networks with the TCP extension of allowing multiple paths (MPTCP) [24] disregarding any overhead considerations.

A. Calculating the Upper Bound with Idealized Caching

In order to experimentally compare client adaptation strategies and forwarding strategies to their theoretical upper bounds in NDN we extend the LP given in Equations 1a to 1d such that it takes caching into account. Therefore, we assume *idealized* caching along each path a data packet has been sent and that intermediate nodes have unlimited cache size. First, we determine how many different multimedia contents (M) are retrieved by the clients. Second, we add each client that streams the same multimedia content $m \in M$ with $k \in C$ to a set M_k (cf. Algorithm 1 line 1, denoted by $disjointStreams$, $L \subseteq M \times C$). Third, we pick a possible combination of $n = |M|$ clients such that the clients request pairwise disjoint ($m_i \neq m_j, i \neq j$) multimedia content (cf. Algorithm 1 line 2, denoted by $getNextCombination$). Thus, we have $\prod_{j=1}^{|L|} |L_j|$ possible combinations. Fourth, their paths are computed and the LP given in Equations 1a to 1d is solved for these n clients (cf. Algorithm 1 line 3, denoted by $solveMCFP$). This yields the optimum for these n clients. Fifth, we use the residual graph rG as network graph for the remaining $|C| - n$ clients and we set all the vertices from all paths for each of the n clients as *servers* for the other clients that are about to stream the same multimedia content (cf. Algorithm 1 line 4, denoted by $createClientServerPairs$). Therefore, we assume that all the nodes on the corresponding paths have *cached all the data* from their corresponding multimedia streams (if and only if the LP given in Equations 1a to 1d is feasible for the selected combination). For instance, if a client uses two paths to retrieve the desired data, not all nodes on the two paths will cache the same data (because the Interests may arbitrarily be forwarded on these two paths). Sixth, we solve the MCFP using the modified set of client-server pairs S' and the residual graph/network (cf. Algorithm 1 line 5, denoted by $solveMCFP$). This procedure is repeated for all possible combinations and the result provides the highest average multimedia streaming bit rate assuming *idealized* caching and unlimited cache size on each intermediate node.

B. Example

Figure 1 depicts an example network with two clients (C_1 and C_2) interested in the same content available at a single server (S). For the sake of simplicity we assume that the capacity of the links between the vertices is 1500 kbps and that the links are bidirectional. We further assume that the server provides SVC-encoded multimedia content with the following three layers/bitrates: $\{L0 = 640$ kbps, $L0 + L1 = 995$ kbps, $L0 + L1 + L2 = 1400$ kbps$\}$. In order to solve the LP given in Equations 1a to 1d, which assumes that none of the nodes in the network cache content, we compute all paths for the client-server pairs. The paths for the client-server pairs (C_1, S) and (C_2, S) are as follows: $P = \{\{(C_1,1),(1,2),(2,3),(3,4),(4,S)\},\{(C_1,1),(1,5),(5,6),(6,4),(4,S)\},\{(C_1,1),(1,5),(5,7),(7,4),(4,S)\},\{(C_2,7),(7,4),(4,S)\},\{(C_2,7),(7,5),(5,6),(6,4),(4,S)\},\{(C_2,7),(7,5),(5,1),(1,2),(2,3),(3,4),(4,S)\}\}$. So, each client has three possible paths to the server. Although the LP does not consider caching, it considers multi-path transmission as foreseen in NDN and MPTCP. The solution of the LP indicates that in the given network an average download bitrate of 750 kbps can be retrieved by the clients. This takes into account that the minimum bitrate of 640 kbps (the lowest representation/layer) shall be achieved by all clients, so that no stalls of the playback occur. If we take caching into account, we have to use Algorithm 1. In this case the achieved average download bitrate per client would be 1400 kbps. A more detailed investigation of the solution shows that C_1 should request the multimedia content from the content origin S using the three available paths such that C_1 is able to achieve a media bitrate of 1400 kbps. Client C_2 then has 23 paths to all intermediate network nodes that are on the three paths from C_1 to S. Since we assume *idealized* caching, these nodes have the desired multimedia content in their local cache. Thus, C_2 is also able to maintain a download bitrate of 1400 kbps. This is for sure a very artificial result since it assumes that, even though not all the Interests passed through a network node, that node still has every data packet in its cache.

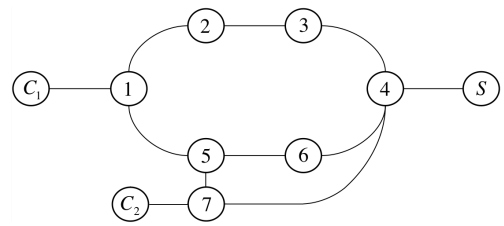


Fig. 1: Example network with two clients (C_1 and C_2) interested in the same content available at the server (S).

We provide a MATLAB implementation for solving the given LP at [25] licensed under the LGPL. We further provide the source code used for the evaluation (cf. Section IV-A) as open source contribution at [25] under the GPL.

IV. Performance of DAS in NDN

To evaluate and investigate the performance of pull-based multimedia streaming in NDN using the adaptation algorithms described in Section II-A, and forwarding strategies described in Section II-B, we use *ndnSIM 2.0* [21], a simulation software based on *ns-3*. First, we outline and justify the evaluation set-up. Then, we present the results comparing them to the theoretical upper bounds determined using the MCFP from

Section III without and with *idealized* caching assuming unlimited cache sizes.

A. Evaluation Setup

As test content we use MPEG-DASH-compliant SVC-encoded multimedia content with a segment size of two seconds. The multimedia content is taken from the SVC dataset [26]. The dataset provides four movies with an average duration of about 12 minutes. We concatenated the movies to obtain content with a duration of about 48 minutes, which is roughly the length of a typical TV episode. In [26] the multimedia content is encoded in various variants. A variant defines the encoding parameters as well as the scalability domains (temporal, spatial, quality). For this evaluation we have chosen a variant providing SNR scalability only since the scalability domain(s) do not influence the objective streaming performance, but only the subjective one (QoE). The chosen content is provided using a base layer and two enhancement layers. The base layer (henceforth denoted as $L0$) has an average bitrate of approx. 640 kbps. The first enhancement layer ($L1$) has a bitrate of approx. 355 kbps. In order to play back a segment at the quality of $L1$, one has to fetch the same segment of $L0$ yielding a combined multimedia bitrate of $L0 + L1 \approx 995$ kbps. The second enhancement layer ($L2$) has an average bitrate of approx. 407 kbps (resulting in a cumulative bitrate of $L0 + L1 + L2 \approx 1400$ kbps).

Figure 2 depicts the *fixed* network topology for the evaluation in order to investigate the pull-based streaming performance of forwarding strategies coupled with different client-based adaptation mechanisms. The network topology is *fixed* to ensure comparability among the simulations and the theoretical upper bounds provided by MCFP. We are aware that the fixed topology is a limitation, however, otherwise the results could not be compared to the theoretical work from Section III. In total 25 clients are placed in the network. Every five clients request the same multimedia content from the corresponding server. Thus, we have five groups of clients denoted by the colors (or numbers) red (1), green (2), blue (3), orange (4) and black (5) (cf. Figure 2). The servers are illustrated as rectangles labeled with S using the corresponding group color. The network nodes are equipped with a cache. The size of the cache is varied from 25 MB, 50 MB up to 100 MB per node, which corresponds to a cache size of approximately 1%, 2% and 4% of the total content catalogue, respectively. As suggested in [27], we consider two caching approaches: first, *Cache Everything Everywhere* (CEE), and second, *Probabilistic Caching* (PC) with a probability $p \in \{0.1, 0.3, 0.6\}$ of caching seen content using a Least Recently Used (LRU) replacement strategy for both approaches.

We use two settings for the start times of the clients. The first setting exactly follows the problem description of the MCFP, which requires that all clients are configured to start simultaneously. This is again a limitation, yet required to ensure comparability to the theoretical results from Section III. As the clients start at the same time, this may be beneficial for the overall caching performance, since requests for the same content are issued in a small time window and can be aggregated by the forwarding nodes. For the second setting, we

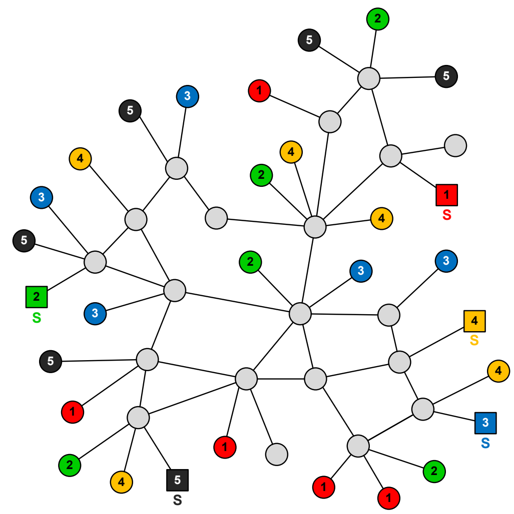
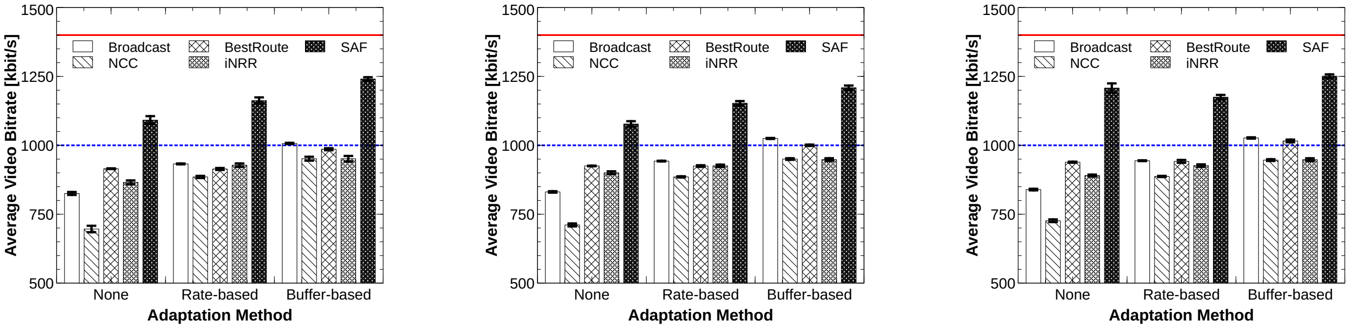


Fig. 2: Topology used for evaluating the multimedia streaming performance in NDN.

draw the start time of a client from an exponential distribution (*mean=60s*, *max=180s*). This shall mimic the behavior of users joining streaming sessions, especially during prime time when a new movie or event is shown. We expect that the caching performance, and therefore the overall performance, will be worse than in the first setting, because the requests for the same content are issued in a larger time window and fewer requests can be aggregated or result in cache hits.

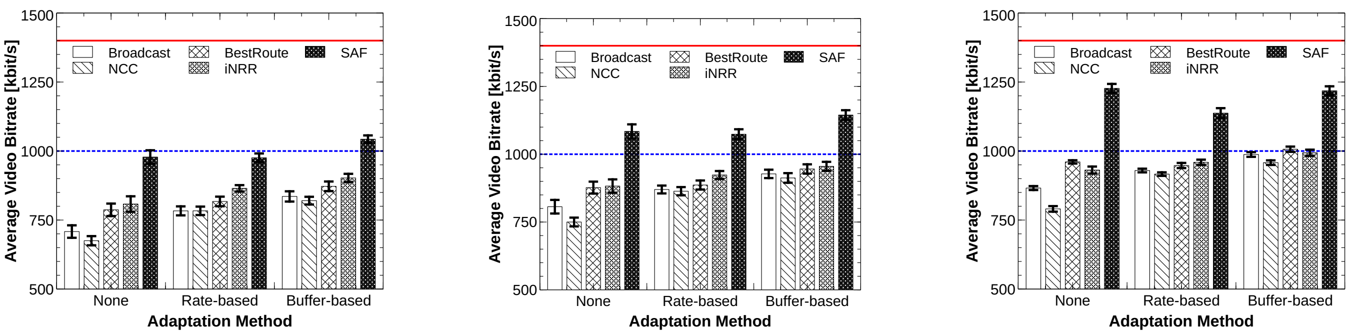
The links between the network nodes are bidirectional and have a bandwidth of 4 Mbps (in each direction). The links connecting the servers to their ingress/egress nodes have a bandwidth of 5 Mbps (bidirectional). The network links connecting the clients to their ingress/egress nodes have 2 Mbps (bidirectional). The presented topology with the given settings has been selected because it is likely that congestion will occur if all clients want to stream the multimedia content. For every configuration (forwarding strategy + adaptation algorithm) we conducted 25 simulation runs in order to reduce the influence of random variables on the sample means in the results. For the forwarding algorithms we pre-computed all possible routes to evaluate all forwarding strategies under the same conditions. The algorithms, Broadcast, Bestroute and NCC do consider the propagated routes for forwarding only, while SAF and iNNR use the routing information merely as starting point. The selected topology does not favor any of the forwarding strategies. All clients maintain a playback buffer that is capable of storing 50 seconds of multimedia content. Request from client applications are issued based on a constant bitrate model as congestion in NDN shall be handled by the forwarding plane [28].

In addition to the NDN specific simulations using ndnSIM 2.0, we provide a baseline evaluation of DASH using OMNeT++ as simulation environment utilizing the INET framework. We use the presented topology (cf. Figure 2) and assume that the intermediate nodes are routers/switches. In order to obtain baseline results for HTTP adaptive streaming we use the rate-based adaptation logic introduced in Section II-A and we set the playback buffer size to 50 seconds. As for the NDN scenario, we vary the starting times of the clients as



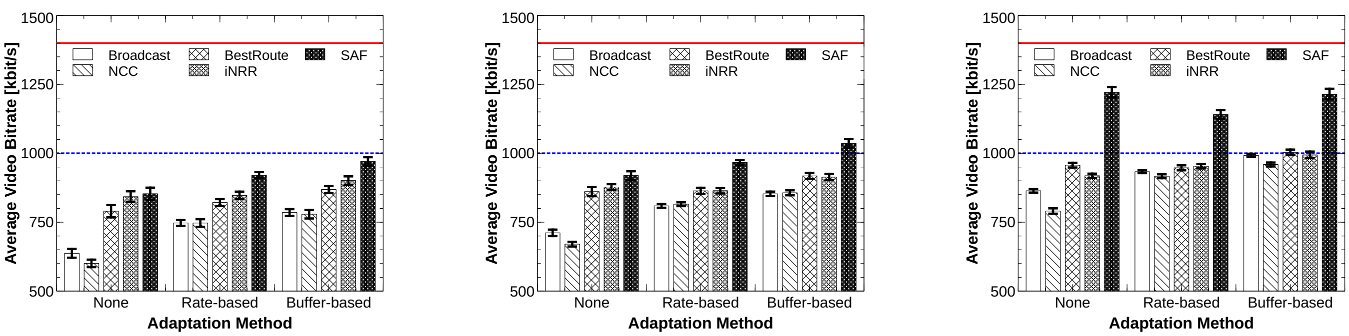
(a) **25 MB** cache size per node (b) **50 MB** cache size per node (c) **100 MB** cache size per node

Fig. 3: Average achieved bitrates by **simultaneously starting** clients using a **CEE caching** policy.



(a) **25 MB** cache size per node (b) **50 MB** cache size per node (c) **100 MB** cache size per node

Fig. 4: Average achieved bitrates by clients with **exponentially distributed start times** using a **CEE caching** policy.



(a) **25 MB** cache size per node (b) **50 MB** cache size per node (c) **100 MB** cache size per node

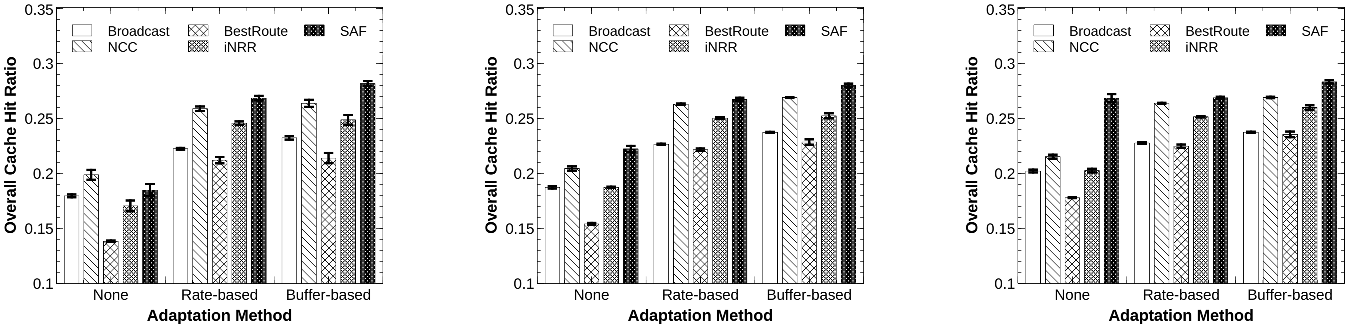
Fig. 5: Average achieved bitrates by clients with **exponentially distributed start times** using a **PC caching** policy ($p = 0.6$).

described before. Instead of SVC-encoded multimedia content, we use AVC-encoded multimedia content since it is the most used video coding standard in conjunction with DASH. We select *Big Buck Bunny* from the dataset [29] with a segment size of two seconds. In order to obtain the same duration as the content used for the NDN simulations, we extended the length of Big Buck Bunny with itself.

B. Results

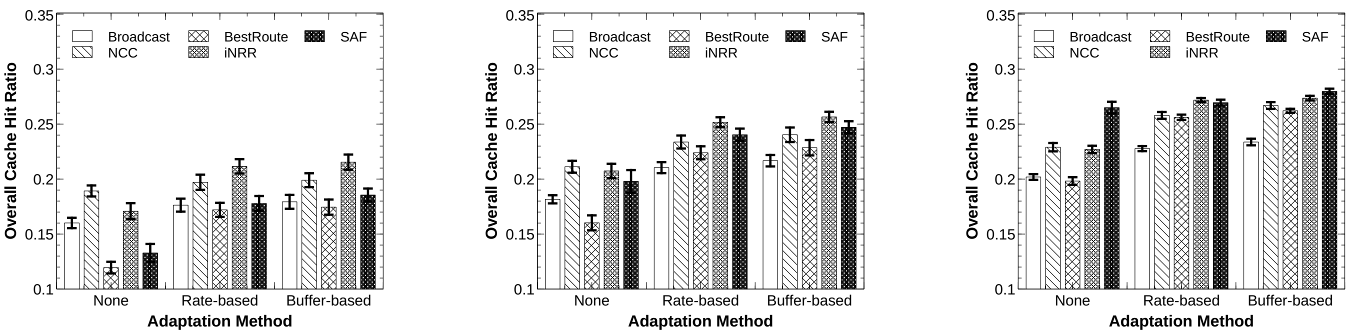
For the baseline evaluation of MPEG-DASH in a TCP/IP scenario, we obtained the following results for the average streaming bitrates. Considering simultaneous starting times of clients, we obtained an average streaming bitrate of 416 kbps and ± 0.382 kbps for the 95% confidence interval (CI). In the case of exponentially distributed starting times we obtained an average streaming bitrate of 423.441 kbps ± 0.736 kbps (95% CI). As expected the performance is low. This is due to the fact that only single paths can be used by DASH and no caching of content takes place at network nodes.

Figures 3, 4, and 5 depict the 95% CI of the average video bitrates achieved by clients considering the combination of the different forwarding strategies, adaptation logics, caching strategies and starting times of the clients in NDN. The dashed (blue) line in the figures indicates the theoretical upper bound for the average video bitrate when solving the MCFP without consideration of caching as introduced in Section III. The solid (red) line indicates the theoretical upper bound for the average video bitrate that is obtained by solving the MCFP assuming idealized caching as introduced in Section III-A. In order to account for segments that are not retrieved in time causing stalls (the segment is not available until its associated playback timestamp), we penalize the average video bitrate by counting



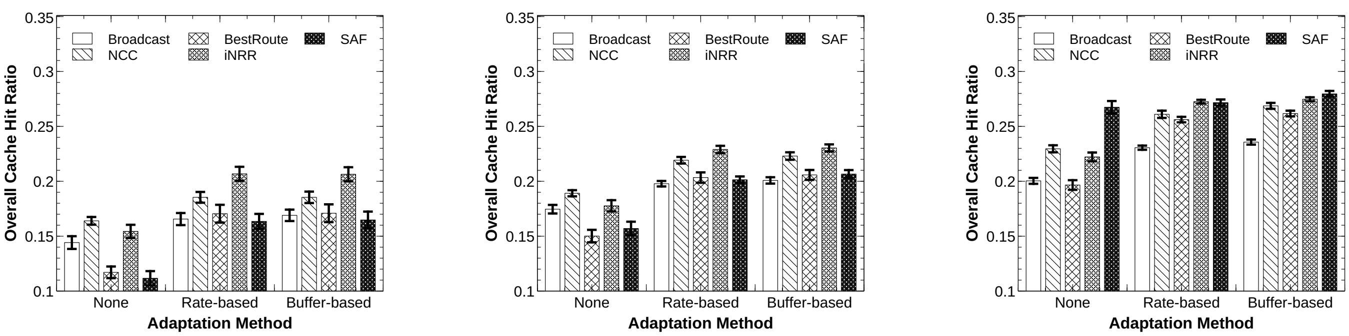
(a) **25 MB** cache size per node (b) **50 MB** cache size per node (c) **100 MB** cache size per node

Fig. 6: Average achieved cache hit ratio per node with a **simultaneous start time** using a **CEE caching** policy.



(a) **25 MB** cache size per node (b) **50 MB** cache size per node (c) **100 MB** cache size per node

Fig. 7: Average achieved cache hit ratio per node with an **exponentially distributed start time** using a **CEE caching** policy.



(a) **25 MB** cache size per node (b) **50 MB** cache size per node (c) **100 MB** cache size per node

Fig. 8: Average achieved cache hit ratio per node with **exponentially distributed start times** using **PC caching** ($p = 0.6$).

a zero bitrate segment in lieu thereof.

Having the results of the baseline DASH evaluation in mind, it is evident that any combination of cache size, forwarding strategy and adaptation logic (even no adaptation logic) is able to obtain a higher average video bitrate in NDN. Based on the figures, we make the following observations. BestRoute, which strongly focuses on the single *best* delivery path, benefits from caching in contrast to the standard TCP/IP scenario with DASH. The other forwarding strategies (particularly SAF), which make extensive use of multi-path forwarding, obtain an extra performance boost, especially when the cache size increases. So, increasing the cache size has a positive impact on the average obtained video bitrate by the clients. Assuming an exponential distribution of the starting times of the clients has a negative impact on the average video bitrate obtained for both caching policies CEE and PC. The performance of PC is worse than that of CEE, particularly with small cache sizes. Please note that we only present results for PC with parameter $p = 0.6$ because lower values provide even lower cache hit ratios. This is due to the selected topology (cf. Figure 2) [27]. We further observe that the buffer-based adaptation logic obtains a higher average video bitrate compared to results of the rate-based adaptation logic. When we distribute the starting times of clients exponentially, we observe that SAF achieves the highest average video bitrate even without any adaptation logic when having bigger cache sizes (e.g., 50 MB or 100 MB). This is caused by the fact that SAF tries to maximize the throughput and in this case the retrieved representation of the multimedia content is not restricted by the adaptation logic.

These findings are affirmed by Figure 6, 7, and 8 which depict the 95% CI of the overall cache hit ratio with respect to the combination of forwarding strategies, caching strategies

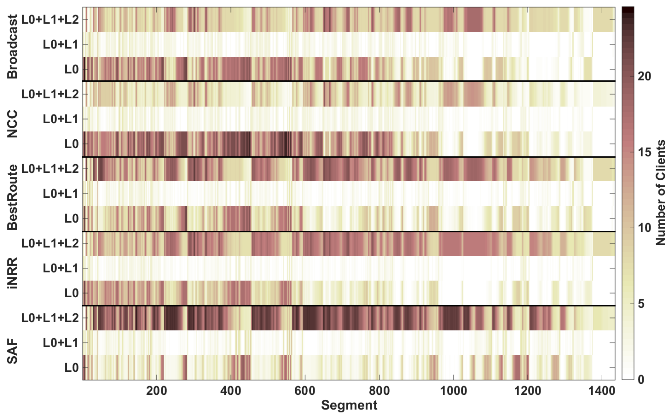


Fig. 9: Number of clients that retrieve a given segment with a certain quality (layers, e.g., $L0 + L1$) for playback under different forwarding strategies using **no adaptation**.

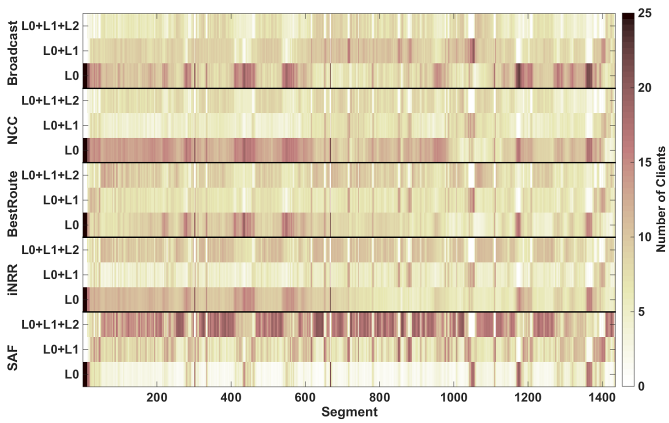


Fig. 10: Number of clients that retrieve a given segment with a certain quality (layers, e.g., $L0 + L1$) for playback under different forwarding strategies using **rate-based adaptation**.

and starting times of the clients. It is evident from the figures that the larger the cache size the higher the overall cache hit ratio. However, also the selected adaptation method and forwarding strategy have significant influence on the cache hit ratio. The buffer-based adaptation is able to obtain a higher cache hit ratio than the rate-based adaptation logic. Having no adaptation mechanism affects the cache hit ratio negatively leading to the worst results. Having a closer look at the influence of the forwarding strategies, it shows that particularly the strategies BestRoute and iNRR maintain the highest cache hit ratios if cache sizes are low (e.g., 25 MB). However, SAF is able to achieve the highest cache hit ratio among all forwarding strategies when the cache size increases (e.g., 100 MB).

Still, the average video bitrate and cache hits do not tell the whole story. To further assess the performance of the adaptation logics, we take a look at the clients' switching frequencies among the available representations and their playback stability with respect to the representations. Therefore, we study their behavior in the case of letting the clients start streaming simultaneously, having CEE as the caching strategy and a cache size of 50 MB. The figures are very similar for the other parameter settings (and are omitted due to space constraints). Figures 9, 10, and 11 depict the number of clients that are able to retrieve a certain quality of a segment under

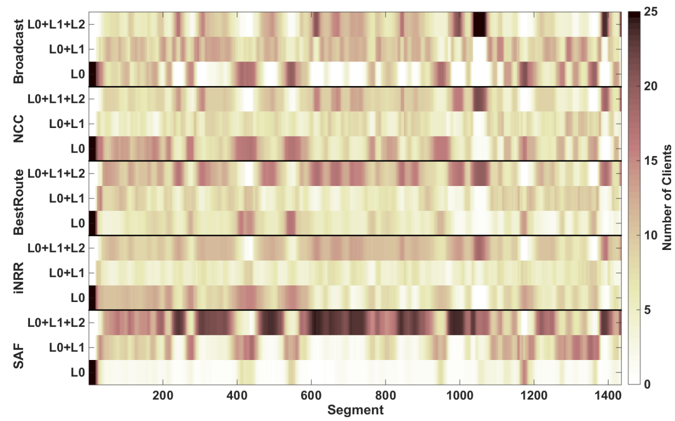


Fig. 11: Number of clients obtaining a given segment with a certain quality (layers, e.g., $L0 + L1$) for playback under different forwarding strategies using **buffer-based adaptation**.

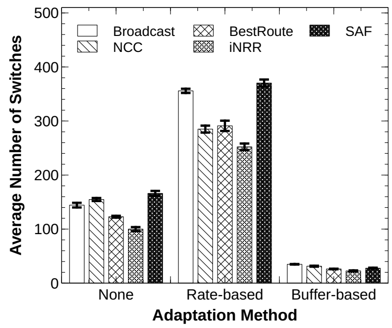
different forwarding strategies and adaptation mechanisms for the mentioned settings, respectively. The x-axis denotes the segment numbers (which have a duration of two seconds) and the y-axis denotes the representations (layers) for every forwarding strategy. The figure depicts the number of clients receiving the different representations (layers) over time. The optimal case would occur if the row of $L0+L1+L2$ would be black, and all others white. This would indicate that all clients have got the highest available representation for all segments.

Figure 9 depicts the case where no adaptation strategy is used. The figure clearly shows that the clients suffer from stalls if no adaptation algorithm is employed regardless of the forwarding strategy, indicated by the bright areas for the last segments (many of the clients are not able to retrieve these segments during the simulation time due to previous playback stalls). The forwarding strategy SAF clearly outperforms the other strategies as more clients are receiving a high quality layer (e.g., $L0 + L1 + L2$), followed by BestRoute and iNNR, which lie close together.

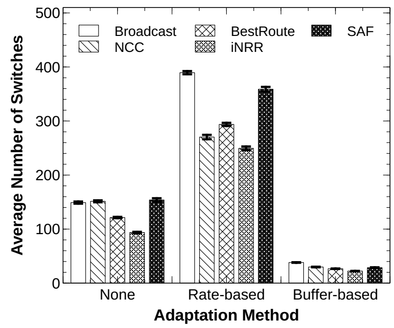
Figure 10 depicts the same case using a rate-based adaptation algorithm. The rate-based adaptation mechanism enables the clients to receive more segments during the simulation time for playback, so fewer playback stalls are encountered by the clients (darker tail). This is due to the fact that fewer clients receive the best quality ($L0 + L1 + L2$). The available bandwidth is distributed more equally among the clients. Comparing Figure 9 and Figure 10 one observes that the latter shows more fine-grained variation patterns. This indicates that the clients in Figure 10 suffer from higher representation switching frequencies.

Figure 11 depicts the received quality when the clients use a buffer-based adaptation mechanism. The first thing that attracts the attention is that the buffer-based adaptation provides a more stable quality to the clients, indicated by the homogeneous colored areas (compared to no adaptation and rate-based adaptation). Furthermore, all forwarding strategies are able to provide a better quality to the clients compared to the rate-based and no adaptation approaches.

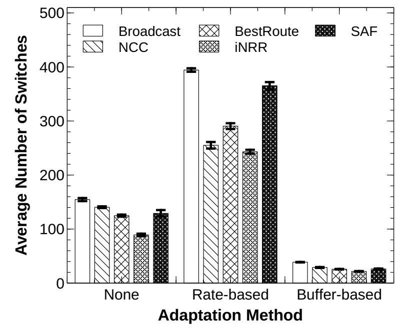
Figures 12, 13, and 14 depict the 95% CI of the average number of representation switches per client for the given parameter settings. Comparing Figures 12, 13, and 14 to Fig-



(a) **25 MB** cache size per node

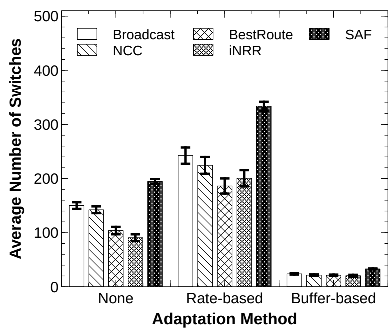


(b) **50 MB** cache size per node

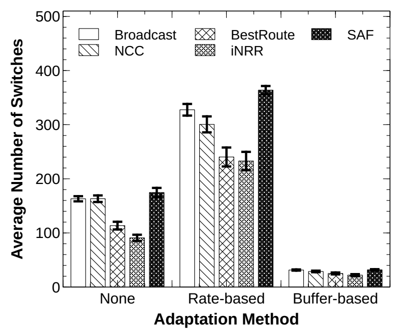


(c) **100 MB** cache size per node

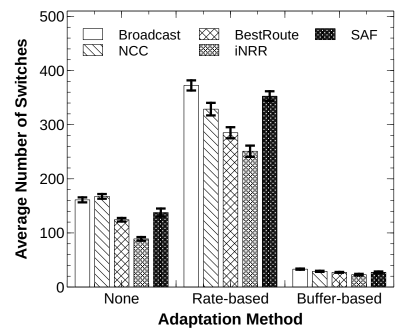
Fig. 12: Average number of switches per client with a **simultaneous start time** using a **CEE caching** policy.



(a) **25 MB** cache size per node

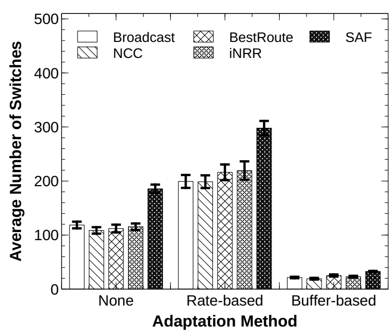


(b) **50 MB** cache size per node

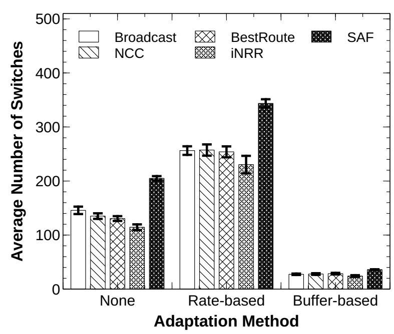


(c) **100 MB** cache size per node

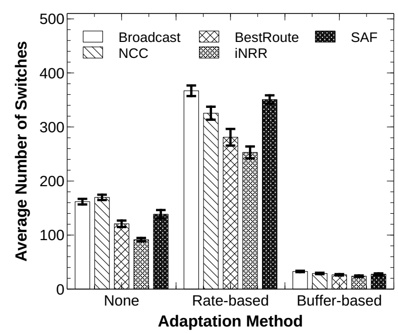
Fig. 13: Average number of switches per client with an **exponentially distributed start time** using a **CEE caching** policy.



(a) **25 MB** cache size per node



(b) **50 MB** cache size per node



(c) **100 MB** cache size per node

Fig. 14: Average number of switches per client with **exponentially distributed start times** using **PC caching** ($p = 0.6$).

ures 9, 10 and 11, it follows that rate-based adaptation causes the clients to oscillate between different representations. This has two reasons. First, NDN's multi-path transmission does not allow an accurate estimate of the available bandwidth. Second, if Interests cause a *hit* in a cache near to the client, the rate-based adaptation mechanism reacts and over-estimates the bandwidth when requesting the next segment (most likely from an higher representation) which then may lead to a cache *miss* because only the previously requested representation is cached. Taking a look at the cache hit ratios (cf. Figures 6, 7, and 8) we see that with higher cache hit ratios the number of switches increases in the case of the rate-based adaptation logic. Thus, we can conclude that the oscillation effect caused by a rate-based adaption logic is amplified if the cache size is increased. The results show that oscillation can be easily avoided by using a buffer-based adaptation logic instead of a rate-based one.

V. Discussion and Conclusion

The goal of this paper was to investigate the multimedia streaming performance in NDN relying on the principles of DAS. We have presented an MCFP that provides the theoretical upper bounds for multi-path multimedia transmissions without and with caching. These bounds *do not consider protocol overhead* introduced by NDN; thus, they are *purely theoretical*. Nevertheless, the bound obtained when solving the MCFP given by Equations 1a to 1d provides the upper bound for traditional IP-based networks using a multi-path enabled transmission protocol (e.g., MPTCP) without considering proxies acting as caches. In Section IV we showed that today's most prominent streaming technology MPEG-DASH over TCP/IP is far away from the optimum without caching, which is definitely due to a lack of multi-path support in IP networks. Considering NDN's inherent multi-path and caching capabilities we assumed that it will easily exceed the first theoretical bound that does not consider caching. The results clearly show that this is possible if an appropriate forwarding strategy and sufficiently large caches are employed. There is still a gap between the second theoretical bound that considers *idealized* caching and the results that can be reached in NDN.

It is evident from the results that buffer-based adaptation mechanisms should be preferred in NDN. Since multi-path transmissions and network-inherent caching do not allow for an accurate estimation of the available bandwidth, rate-based adaption logics should not be used. Clients in NDN will suffer from the same oscillation behavior as in traditional IP-based networks using HTTP proxies as caches, if rate-based adaptation is employed [30]. This oscillation behavior is amplified by increasing cache sizes. Grandl et al. provide a detailed discussion on this topic [31].

In this paper we have investigated client-based adaptation only. We briefly discuss in-network adaptation of multimedia content as a second possibility. It could be achieved by allowing each node to decide whether to adapt the multimedia content by discarding ("not forwarding") Interests that belong to a specific version/representation of the content [17]. But a purely distributed approach may not be feasible due to NDN's inherent multi-path transmission principle. Thus, it is likely that decisions have to be coordinated among all nodes that are forwarding Interests for a specific multimedia content. This implies a communication protocol such that the nodes can communicate and exchange adaptation information. This would introduce even more overhead and complexity.

Acknowledgment. This work was supported in part by the Austrian Science Fund (FWF) under the CHIST-ERA project CONCERT, project number *I1402*. We are grateful for the feedback from the anonymous JSAC reviewers, which greatly improved the article.

References

[1] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, 2011.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A Survey of Information-centric Networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.

[3] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, "A Survey of Information-Centric Networking Research," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[4] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of Information (NetInf) - An Information-centric Networking Architecture," *Computer Communications*, vol. 36, no. 7, pp. 721 – 735, 2013.

[5] N. Fotiou, D. Trossen, and G. Polyzos, "Illustrating a Publish-Subscribe Internet Architecture," *Telecomm. Sys.*, vol. 51, no. 4, pp. 233–245, 2012.

[6] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *Proc. of the 5th Int. Conf. on Emerging Networking Experiments and Technologies*, 2009.

[7] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A Data-oriented (and Beyond) Network Architecture," in *Proc. of the Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm.* ACM, 2007, pp. 181–192.

[8] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *ACM SIGCOMM Computer Comm. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.

[9] L. Fleischer, "Approximating Fractional Multicommodity Flow Independent of the Number of Commodities," in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 1999, pp. 24–31.

[10] S. Lederer, C. Müller, B. Rainer, C. Timmerer, and H. Hellwagner, "Adaptive Streaming over Content-Centric Networks in Mobile Networks Using Multiple Links," in *Proceedings of the IEEE International Conference on Communications Workshops (ICC)*, 2013, pp. 677–681.

[11] ——, "An Experimental Analysis of Dynamic Adaptive Streaming over HTTP in Content Centric Networks," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*, July 2013, pp. 1–6.

[12] Y. Liu, J. Geurts, J.-C. Point, S. Lederer, B. Rainer, C. Müller, C. Timmerer, and H. Hellwagner, "Dynamic Adaptive Streaming over CCN: A Caching and Overhead Analysis," in *Proceedings of the IEEE International Conference on Communications*, 2013, pp. 3629–3633.

[13] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F. Plass, P. Stewart, and J. D. Thornton, "VoCCN: Voice-over Content-centric Networks," in *Proc. of the Workshop on Re-architecting the Internet*. ACM, 2009.

[14] A. Detti, M. Pomposini, N. Blefari-Melazzi, S. Salsano, and A. Bragagnini, "Offloading Cellular Networks with Information-Centric Networking: The Case of Video Streaming," in *Proc. of the IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks*, 2012.

[15] S. Lederer, D. Posch, C. Timmerer, C. Westphal, A. Azgin, S. Liu, C. Müller, A. Detti, and D. Corujo, *Adaptive Video Streaming over ICN*, Internet Draft, ICNRG/IETF, 2015.

[16] 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, 2007.

[17] 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 Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, 2014, pp. 25–30.

[18] B. Rainer, S. Lederer, C. Müller, and C. Timmerer, "A Seamless Web Integration of Adaptive HTTP Streaming," in *Proceedings of the 20th European Signal Processing Conference*, Aug 2012, pp. 1519–1523.

[19] C. Sieber, T. Hoßfeld, T. Zinner, P. Tran-Gia, and C. Timmerer, "Implementation and User-centric Comparison of a Novel Adaptation Logic for DASH with SVC," in *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*, 2013.

[20] C. Yi, J. Abraham, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "On the Role of Routing in Named Data Networking," in *Proc. of the 1st Int. Conference on Information-centric Networking*, 2014, pp. 27–36.

[21] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM 2.0: A new version of the NDN simulator for NS-3," Tech. Rep., 2015.

[22] D. Posch, B. Rainer, and H. Hellwagner, "SAF: Stochastic Adaptive Forwarding in Named Data Networking," in *arXiv.org*, 2015, pp. 1–10.

[23] G. Rossini and D. Rossi, "Coupling Caching and Forwarding: Benefits, Analysis, and Implementation," in *Proc. of the 1st International Conference on Information-centric Networking*. ACM, 2014, pp. 127–136.

[24] S. Barr, C. Paasch, and O. Bonaventure, "MultiPath TCP: From Theory to Practice," in *NETWORKING 2011*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6640, pp. 444–457.

[25] ITEC, "Open Source Implementations of Multimedia Streaming Scenarios: https://github.com/danposch/itec-ndn," 2015.

[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 Conference on Multimedia Systems*, 2015.

[27] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache Less for More in Information-Centric Networks," in *NETWORKING 2012*, ser. Lecture Notes in Computer Science. Springer, 2012, vol. 7289, pp. 27–40.

[28] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A Case for Stateful Forwarding Plane," *Computer Communications*, vol. 36, no. 7, pp. 779 – 791, 2013.

[29] S. Lederer, C. Müller, and C. Timmerer, "Dynamic Adaptive Streaming over HTTP Dataset," in *Proceedings of the 3rd Multimedia Systems Conference*, ser. MMSys '12. ACM, 2012, pp. 89–94.

[30] C. Müller, S. Lederer, and C. Timmerer, "A Proxy Effect Analyis and Fair Adaptation Algorithm for Multiple Competing DASH Clients," in *IEEE Visual Communications and Image Processing*, Nov 2012.

[31] R. Grandl, K. Su, and C. Westphal, "On the Interaction of Adaptive Video Streaming with Content-Centric Networking," in *20th International Packet Video Workshop*, 2013.

Benjamin Rainer received the B.Sc., M.Sc. (Dipl.-Ing.), and Ph.D. (Dr. techn.) with distinction all from the Alpen-Adria-Universität Klagenfurt. He is a PostDoc researcher at the Department of Information Technology (ITEC) in the Multimedia Communication (MMC) research group, Alpen-Adria-Universität Klagenfurt, Austria.

Daniel Posch received the B.Sc., and M.Sc. (Dipl.-Ing.) with distinction all from the Alpen-Adria-Universität Klagenfurt. He is pursiung a Ph.D. (Dr. techn.) in computer science at the Department of Information Technology (ITEC) in the Multimedia Communication (MMC) research group, Alpen-Adria-Universität Klagenfurt, Austria. His research interests are multimedia technologies in ICN/NDN. Currently, he works on Interest forwarding strategies for NDN enhancing multimedia content delivery.

Hermann Hellwagner is a full professor of computer science in the Institute of Information Technology (ITEC), Alpen-Adria-Universität Klagenfurt, Austria, leading the Multimedia Communications group. His current research areas are distributed multimedia systems, multimedia communications, ICN/NDN and quality of service. He has received many research grants from national (Austria, Germany) and European funding agencies as well as from industry, is the editor of several books, and has published more than 200 scientific papers on parallel computer architecture, parallel programming, and multimedia communications and adaptation. He is a member of the IEEE, ACM, GI (German Informatics Society) and OCG (Austrian Computer Society), and vice president of the Austrian Science Fund (FWF).